การออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน

MOBILE APPLICATION DESIGN FOR HEARING TEST AND HEARING AID FOR HEARING IMPAIRED

โดย

นางสาวชนกนันท์ สุรมูล

นายธนิสร ภิญโญพรพาณิชย์

นางสาวธวัลรัตน์ เติมเวชศยานนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

การออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน

MOBILE APPLICATION DESIGN FOR HEARING TEST AND HEARING AID FOR HEARING IMPAIRED

โดย

นางสาวชนกนันท์	สุรมูล	61010188
นายธนิสร	ภิญโญพรพาณิชย์	61010474
นางสาวธวัลรัตน์	เติมเวชศยานนท์	61010489

อาจารย์ที่ปรึกษา

ผศ. อัครพล ตรีรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2564

ปริญญานิพนธ์ปีการศึกษา 2564

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน

MOBILE APPLICATION DESIGN FOR HEARING TEST AND HEARING AID FOR HEARING IMPAIRED

ผู้จัดทำ

1. นางสาวชนกนันท์ สุรมูล 61010188
2. นายธนิสร ภิญโญพรพาณิชย์ 61010474
3. นางสาวธวัลรัตน์ เติมเวชศยานนท์ 61010489

อัครพล ตรีรัตน์

..... **อาจารย์ที่ปรึกษา**

(ผศ. อัครพล ตรีรัตน์)

กิตติกรรมประกาศ

การดำเนินปริญญานิพนธ์ “การออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน” จะไม่สามารถสำเร็จลุล่วงไปได้ หากไม่ได้รับความอนุเคราะห์และสนับสนุนจากผศ. อัครพล ตรีรัตน์ อาจารย์ที่ปรึกษา และผศ.ดร. ศรวัฒน์ ชิวปรีชา ที่ให้คำแนะนำและแนวทางการแก้ไขปัญหา รวมถึงสนับสนุนเครื่องมือและอุปกรณ์ต่าง ๆ ที่ใช้ในปริญญานิพนธ์นี้

ขอขอบคุณท่านอาจารย์ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอนและประสิทธิ์ประสาทวิชาความรู้ให้แก่คณะผู้จัดทำ

ขอขอบคุณผู้มีส่วนเกี่ยวข้องทุกท่านเป็นอย่างสูงที่คอยช่วยเหลือให้ความรู้เพิ่มเติมและขอขอบคุณครอบครัวและเพื่อนที่เป็นกำลังใจให้แก่ผู้จัดทำ ให้สามารถลุล่วงไปได้ด้วยดี

นางสาวชนกนันท์ สุรมูล
นายธนิสร ภิญโญพรพาณิชย์
นางสาวธวัลรัตน์ เติมเวชศยานนท์
ผู้จัดทำ

การออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน

MOBILE APPLICATION DESIGN FOR HEARING TEST AND HEARING AID FOR HEARING IMPAIRED

โดย นางสาวชนกนันท์ สุรมูล 61010188
นายธนิสร ภิญโญพรพาณิชย์ 61010474
นางสาวธวัลรัตน์ เติมเวชศยานนท์ 61010489

อาจารย์ที่ปรึกษา ผศ. อัครพล ตรีรัตน์

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอเกี่ยวกับการออกแบบและสร้างแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับอุปกรณ์มือถือบนระบบปฏิบัติการณ์แอนดรอยด์ แอปพลิเคชันนี้ถูกสร้างขึ้นตามมาตรฐานที่กำหนดไว้ และให้บริการแก่ผู้ใช้งานที่มีความต้องการใช้งานแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังภายในประเทศ ซึ่งการจัดเตรียมปริญญานิพนธ์นี้ได้ทำการออกแบบระบบการทำงานของแอปพลิเคชันสำหรับการทดสอบการได้ยินและแอปพลิเคชันสำหรับเครื่องช่วยฟังผ่านระบบปฏิบัติการณ์แอนดรอยด์โดยการใช้ภาษาโปรแกรมฟลัทเทอร์ (Flutter Framework) สำหรับการจัดเก็บผลการทดสอบสมรรถภาพการได้ยินถูกเก็บบันทึกไว้ที่ฐานข้อมูล รวมไปถึงประวัติของผู้ใช้งาน นอกจากนี้ยังมีการใช้เซิร์ฟเวอร์ผ่านบริการดิจิทัลโอเชียน (DigitalOcean) สำหรับประมวลผลที่ได้จากการทดสอบสมรรถภาพการได้ยิน เพื่อให้ได้มาซึ่งสัมประสิทธิ์ที่ใช้สำหรับการออกแบบวงจรกรองความถี่แบบดิจิทัลชนิดผลตอบสนองของอิมพัลส์จำนวนไม่จำกัด (Infinite Impulse Response) ที่ถูกออกแบบอยู่ในช่วงความถี่ที่ เหมาะสม ในแต่ละช่วงความถี่จะถูกออกแบบให้มีขนาดของอัตราการขยายที่เหมาะสมในแต่ละช่วงความถี่ที่มีค่าแตกต่างกัน เมื่อนำมารวมกันจะให้ผลตอบสนองทางความถี่ที่ใกล้เคียง และเป็นไปตามกราฟได้ยิน (Audiogram) เพื่อถูกนำมาใช้ใน แอปพลิเคชันเครื่องช่วยฟัง เป็นตัวเชื่อมต่อความสัมพันธ์ระหว่างแอปพลิเคชันทั้งสอง การชดเชยการสูญเสียสมรรถการได้ยินจะถูกชดเชยตามช่วงเวลาที่เกิดขึ้นในขณะนั้น ซึ่งจะทำให้ผู้ใช้งานสามารถชดเชยการสูญเสียสมรรถภาพการได้ยินตลอดการใช้งานบนโมบายแอปพลิเคชันเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน

ABSTRACT

This thesis presents the design, creation and results of a hearing test and hearing aid application for Android mobile devices. The application is created in accordance with defined standard and is to serve the possible group of people is a domestic environment. In this thesis, the mobile application was implemented through Android operating system by using Flutter programming language. Results from hearing test application from users were collected in database altogether with user profiles. Furthermore, by utilizing a server through DigitalOcean to process hearing test results in order to obtains the coefficients for the set of infinite impulse response (IIR) digital filters with selected bandwidths are designed. Each of these bands is provided with sufficient magnitude gain such that the different bands will combine to give a frequency response that closely matches the audiogram and then implement in hearing aid applications. As a result, it is the connection between two applications. Hearing loss is compensated by real-time processing on a hearing aid mobile app for the hearing impaired and allowing users to correct for their hearing loss over time.

สารบัญ

สารบัญ (ต่อ)

สารบัญรูป

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญรูป (ต่อ)

หน้า

รูปที่

สารบัญตาราง

หน้า

ตารางที่

สารบัญตาราง (ต่อ)

หน้า

ตารางที่

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันจำนวนของผู้ที่บกพร่องทางการได้ยินมีจำนวนเพิ่มขึ้นจากหลายสาเหตุอาทิเช่น การเพิ่มขึ้นของประชากรผู้สูงอายุภายในประเทศ การอยู่ในสภาวะแวดล้อมที่ก่อให้เกิดมลพิษทางเสียง เป็นต้น ปริญญานิพนธ์นี้จึงเล็งปัญหาที่เกิดขึ้นจึงได้พัฒนาโมบายแอปพลิเคชันที่ใช้สำหรับการทดสอบการได้ยินและการเครื่องช่วยฟังเพื่อให้ทุกคนสามารถเข้าถึงได้ง่ายและใช้งานได้สะดวกมากยิ่งขึ้น

การออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยินจะแบ่งการทำงานออกเป็นสองส่วนได้แก่ ส่วนที่หนึ่งคือ การออกแบบและสร้าง โมบายแอปพลิเคชันสำหรับทดสอบการได้ยิน และส่วนที่สองคือ การออกแบบและสร้างโมบายแอปพลิเคชันสำหรับเครื่องช่วยฟัง โดยมีตัวกลางที่ใช้ในการเก็บผลการทดสอบ ตลอดจนประวัติของผู้ใช้งานคือ ฐานข้อมูล

1.2 วัตถุประสงค์

1) เพื่อศึกษาและประยุกต์ใช้การประมวลผลสัญญาณดิจิทัลบนแอปพลิเคชัน
2) เพื่อศึกษาการเขียนแอปพลิเคชันผ่านโปรแกรม Flutter และการใช้ Database
3) เพื่อศึกษาและพัฒนาเกี่ยวกับแอปพลิเคชันสำหรับผู้ที่มีปัญหาเกี่ยวกับการได้ยิน
4) เพื่อนำการออกแบบวงจรกรองความถี่แบบดิจิทัลให้เข้าถึงผู้ใช้งานได้สะดวกมากยิ่งขึ้น

1.3 ขอบเขตของปริญญานิพนธ์

ปริญญานิพนธ์นี้ได้ออกแบบและพัฒนาการทดสอบสมรรถภาพการได้ยินโดยประยุกต์ในรูปแบบแอปพลิเคชันโดยเป็นการทดสอบสมรรถภาพการได้ยินแบบใช้สัญญาณความถี่เดียว (Puretone) โดยใช้เสียงสัญญาณไซน์ตั้งแต่ความถี่ 250 เฮิรตซ์ 500 เฮิรตซ์ 1000 เฮิรตซ์ 2000 เฮิรตซ์ 4000 เฮิรตซ์ และ 8000 เฮิรตซ์ ให้สามารถทดสอบสมรรถภาพการได้ยินได้ใกล้เคียงกับการทดสอบที่ศูนย์การทดสอบสมรรถภาพการได้ยินมากที่สุด โดนจะนำค่าที่ได้จากการทดสอบไปออกแบบวงจรกรองความถี่แบบดิจิทัลเพื่อใช้ในส่วนของเซิร์ฟเวอร์ ปริญญานิพนธ์นี้ได้ออกแบบ และพัฒนาเครื่องช่วยฟังโดยประยุกต์ให้อยู่ในรูปแบบแอปพลิเคชัน โดยนำวงจรกรองความถี่แบบดิจิทัลที่ได้ทำการออกแบบไปใช้ในส่วนของแอปพลิเคชันเครื่องช่วยฟัง ให้สามารถทดแทนการผลิตและความยุ่งยากของการใช้งานเครื่องช่วยฟังในปัจจุบัน

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญานิพนธ์ “การออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน” ได้จัดทำขึ้นเพื่อออกแบบและพัฒนาระบบการทำงานของการทดสอบสมรรถภาพทางการได้ยินและเครื่องช่วยฟังให้สามารถใช้งานได้บนโมบายแอปพลิเคชัน ซึ่งมีหลักการเกี่ยวกับการทดสอบการได้ยิน การใช้เครื่องช่วยฟังสำหรับชดเชยการสูญเสียการได้ยิน และการเก็บข้อมูลและประวัติการใช้งานในฐานข้อมูลที่สามารถตรวจสอบประวัติย้อนหลังได้ โดยมีทฤษฎีและหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 ศึกษาวิธีการตรวจสมรรถภาพการได้ยิน

การตรวจสมรรถภาพการได้ยิน (Audiometry) เป็นการตรวจวัดความสามารถในการได้ยินของผู้เข้ารับการตรวจจากแพทย์หรือผู้เชี่ยวชาญเกี่ยวกับโสตวิทยา โดยวิธีที่เป็นที่นิยมมากที่สุดคือการตรวจสมรรถภาพการได้ยินโดยใช้เครื่องตรวจการได้ยิน (Audiometer) เนื่องจากเป็นการตรวจที่ค่อนข้างง่าย สะดวก ปลอดภัยแก่ผู้เข้ารับบริการ วิธีการตรวจสมรรถภาพทางหูนั้นสามารถแบ่งออกได้เป็น 3 วิธีได้แก่ [1]

2.1.1 Routine Audiometry

เป็นการตรวจสมรรถภาพการได้ยินที่พบได้ทั่วไปในการตรวจชั้นคลินิก เพื่อใช้วินิจฉัยโรคและติดตามผลการรักษา สามารถแยกออกเป็น 3 วิธี

2.1.1.1 Pure Tone Air Conduction

คือ การตรวจการได้ยินโดยการนำเสียงผ่านอากาศโดยการใช้สัญญาณความถี่เดียว (Puretone) ปล่อยเข้าไปที่หูฟัง (หูฟังสีแดงครอบหูขวา และหูฟังสีน้ำเงินครอบหูซ้าย) เพื่อดูการตอบสนองของผู้เข้ารับการทดสอบว่าได้ยินเสียงที่ปล่อยออกมาหรือไม่ ตามความถี่ออคเตฟ (Octave Frequency) ได้แก่ ช่วงความถี่ 250, 500, 1000, 2000, 4000 และ 8000 เฮิร์ตซ์ (Hertz, Hz) ค่อยๆ ลดระดับความดังจนกว่าผู้เข้ารับการตรวจจะได้ยินระดับความดังที่ต่ำที่สุดของตน โดยวิธีนี้จะเป็นวิธีที่นิยมที่สุดสำหรับการทดสอบสมรรถภาพการได้ยินเนื่องจากง่าย และใช้ระยะเวลาที่สั้น

รูปที่ 2.1 การทดสอบสมรรถภาพการได้ยินแบบ Pure Tone Air Conduction [2]

การทดสอบนั้นจะเริ่มจากหูขวา ที่ความถี่ 1000 เฮิรตซ์ ที่ระดับความดัง 40 เดซิเบล (Decibel, dB) จากนั้นจะตรวจที่ความถี่ 2000, 3000, 4000, 6000, 8000 เฮิรตซ์ กลับมาตรวจที่ความถี่ 500 เฮิรตซ์ และทดสอบที่ความถี่ 1000 เฮิรตซ์อีกครั้ง เพื่อทดสอบความแปรปรวน ซึ่งค่าที่ได้จะต้องแตกต่างกันไม่เกิน 5 เดซิเบล เจ้าหน้าที่จะต้องทำการทดสอบใหม่อีกครั้ง โดยที่การทดสอบความแปรปรวนนี้ไม่ต้องทำกับหูทั้งสองข้าง [2]

ในส่วนของการหาระดับเสียงที่ต่ำที่สุดที่ผู้เข้ารับการทดสอบสามารถได้ยินได้ เจ้าหน้าที่จะทำการลดระดับความดังของสัญญาณเสียงลงทีละ 10 เดซิเบล จนผู้เข้ารับการทดสอบไม่ได้ยิน และจากนั้นจะเพิ่มระดับความดังของสัญญาณเสียงขึ้นทีละ 5 เดซิเบล ให้ได้ยินซ้ำประมาณ 2-4 ครั้ง หลังจากนั้นจะทดสอบในความถี่ถัดๆ ไป โดยเริ่มระดับที่ความดังมากกว่าระดับของสัญญาณเสียงที่ต่ำที่สุดที่ผู้เข้ารับการทดสอบสามารถได้ยินที่ความถี่ก่อนหน้า 30 เดซิเบล จากนั้นจึงใช้วิธีการเดิมที่กล่าวไว้ข้างต้น ทำเช่นเดียวกันกับหูอีกข้างจนครบทุกความถี่

2.1.1.2 Pure Tone Bone Conduction

คือ การตรวจการได้ยินโดยการนำเสียงผ่านกระดูกโดยการใช้แบบใช้สัญญาณความถี่เดียว (Puretone) ทดสอบโดยวางเครื่องสั่นกระดูก (Bone Vibrator) ไว้ที่บริเวณกระดูกมาสตอย์ของหูข้างที่กำลังทดสอบให้สัญญาณเสียงเดินทางผ่านตามรูปที่ 3 ซึ่งการทดสอบแบบ Bone Conduction นั้นจะเป็นการทดสอบของหูชั้นใน [3]

รูปที่ 2.2 การทดสอบสมรรถภาพการได้ยินแบบ Puretone Air Conduction [4]

2.1.1.3 Speech Audiometry

คือ การตรวจการได้ยินโดยใช้คำพูด ซึ่งมีความจำเป็นสำหรับผู้ที่มีปัญหาในการเข้าใจคำพูดในชีวิตประจำวัน ผลที่ได้จากการตรวจในลักษณะนี้จะใช้ร่วมกับการตรวจด้วย Pure Tone เป็นวิธีการประเมินผลขั้นพื้นฐานสำหรับการทดสอบสมรรถภาพการได้ยิน [5]

2.1.2 Masking Audiometry

เป็นการทดสอบการได้ยินที่ต้องใช้ความระมัดระวังเป็นพิเศษ เพื่อไม่ให้ในขณะตรวจสอบเสียงข้างนึงไปมีผลกระทบข้ามมายังกะโหลกของอีกข้างนึง โดยการปล่อยเสียงรบกวนเข้าไป

2.1.3 Special Audiometer

เป็นการทดสอบที่พิเศษนอกเหนือไปจากการทดสอบประจำคลินิกที่ใช้เพื่อการรักษาหรือหาความผิดปกติของโรค

2.2 ศึกษามาตรฐานที่ใช้สำหรับการตรวจสมรรถภาพการได้ยิน

2.2.1 มาตรฐาน ISO 8253

มาตรฐาน ISO 8253 เป็นมาตรฐานที่ใช้สำหรับการทดสอบสมรรถภาพการได้ยินด้วยสัญญาณ Pure Tone (Acoustics — Audiometric test methods) คิดค้นโดย ISO (International Standard Organization) ตีพิมพ์ครั้งแรกในปี ค.ศ. 1983 โดย American National Standards Institute (ANSI) ซึ่งจะแบ่งรายละเอียดที่เกี่ยวข้องกับมาตรฐานออกเป็น 3 ส่วน ได้แก่ [6]

2.2.1.1 ISO 8253-1 : Basic pure-tone air and bone conduction threshold audiometry

เป็นมาตราฐานที่กำหนดการหาจุด Threshold สำหรับการทดสอบสมรรถภาพการได้ยินโดยใช้สัญญาณเสียง Pure Tones ทั้งในรูปแบบของหูฟังและเครื่องสั่นกระดูก ระดับความดันเสียง (Sound Pressure Level) ที่อนุญาตสำหรับการทดสอบการได้ยินโดยการนำเสียงผ่านอากาศและกระดูก รวมไปถึงการคำนวณระดับความดันเสียงรอบข้างสูงสุด (Maximum Permissible Ambient) ซึ่งควรมีค่าความไม่แน่นอนอยู่ที่ 2 หรือ 5 เดซิเบล [7]

2.2.1.2 ISO 8253-2: Sound field audiometry with pure-tone and narrow-band test signals

เป็นมาตรฐานที่กำหนดคุณลักษณะของสัญญาณที่ใช้ทดสอบ การกำหนดเกณฑ์การได้ยินในช่วงความถี่ 125 – 8000 เฮิรตซ์ ขั้นตอนการวัดเสียงโดยใช้ Pure Tone ซึ่งจะไม่ได้รวมถึงข้อกำหนดสำหรับลำโพงมือถือ และการทดสอบการได้ยินโดยใช้คำพูด

- คุณสมบัติของสัญญาณที่ใช้ทดสอบ

สัญญาณที่ใช้ในการทดสอบเป็นสัญญาณ Pure Tone ซึ่งมีคุณสมบัติของแหล่งกำเนิดเสียงอิสระ (Free Sound Field) ดังต่อไปนี้

1) ลำโพงจะถูกจัดวางที่ความสูงเดียวกับระดับศีรษะของผู้รับการทดสอบ โดยแกนอ้างอิงคือ ระยะห่างระหว่างจุดอ้างอิงและลำโพงอย่างน้อย 1 เมตร

2) ระดับความดันเสียงที่เกิดขึ้นที่เกิดจากลำโพงที่ตำแหน่ง 0.15 เมตร จากจุดอ้างอิงจะเบี่ยงเบนไม่เกิน ±1 เดซิเบล จากระดับความดันเสียงที่จุดอ้างอิง สำหรับการทดสอบที่ความถี่สูงสุดและความถี่ 400 เฮิรตซ์ ไม่เกิน ±2 เดซิเบล

3) ความแตกต่างของระดับความดันเสียงที่เกิดจากลำโพงที่จุดบนแกนอ้างอิง 0.15 เมตร จะเบี่ยงเบนจากค่าทางทฤษฎีที่กำหนดโดยกฎระยะความดันเสียงผกผันไม่เกิน ± 1 dB สำหรับสัญญาณทดสอบใด ๆ

- ระดับเสียงรบกวนในห้องที่ใช้ทดสอบ

ระดับเสียงรบกวนในห้องที่ใช้ทดสอบจะต้องเป็นไปตามข้อกำหนดในรูปที่ 2.3 ถ้าค่าต่ำสุดของ Hearing Threshold เป็นค่าอื่นที่ไม่ใช่ 0 เดซิเบล จะต้องวัดในห้องทดสอบตามความเหมาะสม ซึ่งค่า Maximum Permissible Ambient ได้มาจากการเพิ่มค่าของรูปที่ 2.3 ซึ่งเป็นค่าต่ำสุดของระดับเกณฑ์การได้ยินที่วัด

Mid-frequency of one-third-octave band[a,b] Hz	Maximum permissible ambient sound pressure levels (ref. 20 μPa)[c] L_{max} dB	
	Lowest test tone frequency	
	125 Hz	250 Hz
31,5	52	60
40	44	53
50	38	46
63	32	41
80	27	36
100	22	32
125	17	25
160	14	18
200	12	12
250	10	10
315	8	8
400	6	6
500	5	5
630	5	5
800	4	4
1 000	4	4
1 250	4	4
1 600	5	5
2 000	5	5
2 500	3	3
3 150	1	1
4 000	−1	−1
5 000	1	1
6 300	6	6
8 000	12	12
10 000	14	14
12 500	15	15

[a] Using the values given in this table, the lowest hearing threshold level to be measured is 0 dB, with a maximum uncertainty of + 2 dB due to ambient noise. If a maximum uncertainty of + 5 dB due to ambient noise is permitted, the values in this table may be increased by 8 dB. The values are derived from ISO 8253-1, assuming binaural listening conditions.

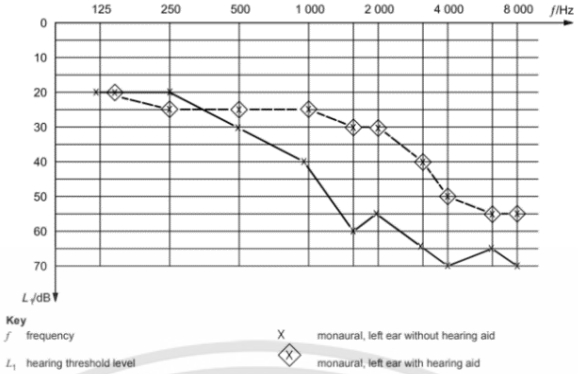
[b] When narrow-band noise is used as a test signal, maximum permitted ambient sound pressure levels should be lower than those specified in this table.

[c] With most of the current sound level meters, it is difficult to measure sound pressure levels below 5 dB.

รูปที่ 2.3 ค่า Maximum Permissible Ambient Sound Pressure Levels [6]

- การปรับเทียบอุปกรณ์ตามระดับการได้ยิน

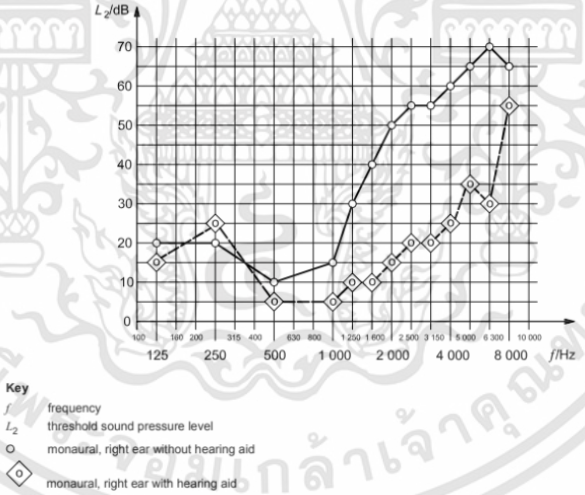
หากอุปกรณ์ที่ใช้สำหรับการทดสอบได้รับการปรับเทียบ (calibrated) ตามระดับการได้ยิน ให้แสดงผลลัพธ์ที่ได้ออกมาอยู่ในรูปของผลการทดสอบสมรรถภาพการได้ยิน (Audiogram) ตาม ISO8253-1 ตามรูปที่ 2.4



รูปที่ 2.4 ผลการทดสอบสมรรถภาพการได้ยินที่ได้รับการปรับเทียบอุปกรณ์ตามระดับการได้ยิน [6]

- การปรับเทียบอุปกรณ์ตามระดับความดันเสียง

หากอุปกรณ์ที่ใช้สำหรับการทดสอบได้รับการปรับเทียบตามระดับความดันเสียงให้แสดงผลลัพธ์ที่ได้ออกมาอยู่ในรูปของผลการทดสอบสมรรถภาพการได้ยิน (Audiogram) ตามรูปที่ 2.5



รูปที่ 2.5 กราฟการได้ยินที่ได้รับการปรับเทียบอุปกรณ์ตามระดับความดันเสียง [6]

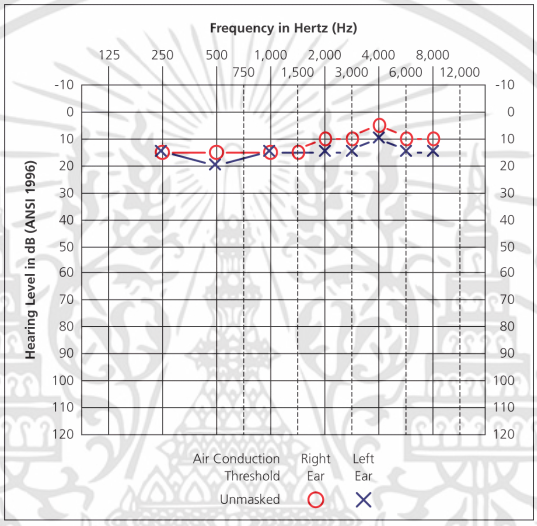
2.2.1.3 ISO 8253-3: Speech Audiometry

เป็นมาตรฐานที่กำหนดสำหรับการทดสอบการได้ยินโดยใช้คำพูด

2.3 ศึกษาผลการทดสอบสมรรถภาพการได้ยิน

จากผลการทดสอบการได้ยินจะแสดงออกมาเป็นผลการทดสอบสมรรถภาพการได้ยิน (Audiogram) เป็นการบอกการสูญเสียการได้ยิน ซึ่งแกน X จะเป็นแกนที่บอกความถี่มีหน่วยเป็น Hz และในแกน Y จะเป็นตัวบอกระดับความดังของเสียงมีหน่วยเป็น dB

หลักการใช้เครื่องหมายในการบันทึกผลตรวจการทดสอบการได้ยินผ่านอากาศ (Pure tone average Air conduction threshold) จะแสดงให้เห็นดังรูปที่ 2.6



รูปที่ 2.6 ตัวอย่างผลการทดสอบสมรรถภาพการได้ยินแบบทดสอบการได้ยินผ่านอากาศ [8]

จะเห็นว่าการทดสอบการได้ยินหูข้างขวาจะใช้เครื่องหมาย O ส่วนการทดสอบการได้ยินหูข้างซ้ายจะใช้เครื่องหมาย X และจากผลการทดสอบสมรรถภาพการได้ยินจะเห็นว่าทุกความถี่ในการทดสอบการได้ยินนั้นมีความดังไม่เกิน 25 dB จึงสรุปว่าผลการทดสอบสมรรถภาพการได้ยินอันนี้มีสมรรถภาพการได้ยินที่ปกติ

2.3.1 ความผิดปกติของระดับการได้ยิน

ในปกติระดับการได้ยินของคนปกติจะอยู่ระหว่าง -10 ถึง 25 dBซึ่งถ้าระดับการได้ยินเสียงสูงกว่านี้จะถือว่าเริ่มผิดปกติโดยสามารถแบ่งระดับความผิดปกติในการได้ยินได้ แสดงตามตารางที่ 2.1

ตารางที่ 2.1 ตารางความผิดปกติของระดับการได้ยิน [2]

ระดับการได้ยิน (dB)	ความผิดปกติ
-10 ถึง 25	การได้ยินอยู่ในระดับปกติ
26 ถึง 40	หูตึงเล็กน้อย
41 ถึง 55	หูตึงปานกลาง
56 ถึง 70	หูตึงมาก
71 ถึง 90	หูตึกอย่างรุนแรง
มากกว่า 90	หูหนวก

2.4 การสร้างสัญญาเสียงเพื่อใช้ในการทดสอบการได้ยิน

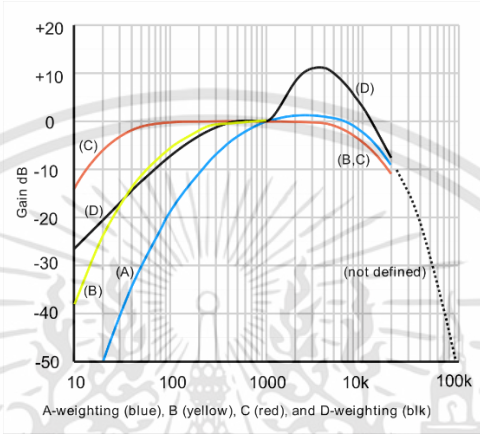
เนื่องจากการทดสอบการได้ยินของแอปพลิเคชันนี้จะเป็นการทดสอบแบบ Air-conduction หรือการทดสอบแบบทางอากาศ โดยการจะปล่อยเสียง Sinewave แต่ละความถี่ทั้งหมด 6 ความถี่ซึ่งจะมี 250, 500, 1000, 2000, 4000 และ 8000 เฮิร์ตซ์ ดังนั้นการทดสอบการได้ยินในแอปพลิเคชันจะต้องมีเสียง Sinewave จึงจะต้องสร้างเสียงจากภาษา python เพื่อนำมาใช้ในการทดสอบ โดยมีวิธีดังนี้

2.4.1 ทฤษฎีความดังของเสียง

การวัดระดับ “ความดังของเสียง” ที่มนุษย์แต่ละคนรับรู้อย่างแท้จริงนั้น ปัจจุบันยังไม่สามารถทำได้เนื่องจากกลไกการได้ยินของหูในมนุษย์แต่ละคนมีความแตกต่างกัน ระดับความเข้มเสียงในแต่ละความถี่ ที่มนุษย์แต่ละคนได้ยินก็แตกต่างกันออกไป รวมถึงการประมวลผลที่สมอง ทำให้การรับรู้ความดังของเสียงในมนุษย์แต่ละคนแตกต่างกันออกไปด้วย แต่เพื่อให้สามารถทำการวัดความดังของเสียงที่มนุษย์ได้รับโดยประมาณได้จึงมีการพยายามวัดความดังของเสียงเป็นหน่วยที่เรียกว่าเดซิเบลเอ (Decibel A หรือ dB(A) หรือ dBA) ขึ้น

หลักของการวัดความดังของเสียงเป็นหน่วยเดซิเบลเอคือ การวัดระดับความเข้มเสียงด้วยเครื่องวัดเสียง (Sound level meter) จะมีการปรับระดับการวัดความเข้มเสียงในแต่ละความถี่ให้มีค่าไม่เท่ากัน โดยการปรับที่นิยมมากที่สุดคือปรับแบบ A-weighting ซึ่งเป็นการปรับความเข้มเสียงที่วัดได้ในแต่ละความถี่ให้มีลักษณะคล้ายคลึงกับความสามารถในการรับเสียงของหูมนุษย์ ซึ่งรับเสียงได้ดีในช่วง 1,000 – 4,000 เฮิรตซ์ การปรับนี้จะทำในลักษณะของการถ่วงน้ำหนัก โดยนำค่าความเข้มเสียงที่วัดได้มาคำนวณแบบลอการิทึมกับค่าถ่วงน้ำหนักที่กำหนดไว้ค่าความดังของเสียงที่ได้จากการปรับแบบ

A-weighting นี้จะมีหน่วยเป็นเดซิเบลเอซึ่งเป็นหน่วยที่นิยมนำมาใช้ในการบอกความดังของเสียงในสิ่งแวดล้อม นอกจากนี้ยังมีการปรับค่าระดับความเข้มเสียงด้วยระบบอื่น ๆ เช่น B-weighting และ D-weighting แต่ในปัจจุบันสองระบบนี้ไม่ได้ใช้แล้ว อีกระบบหนึ่งคือ C-weighting จะทำให้ได้ระดับเสียงเป็นหน่วยเดซิเบลซี

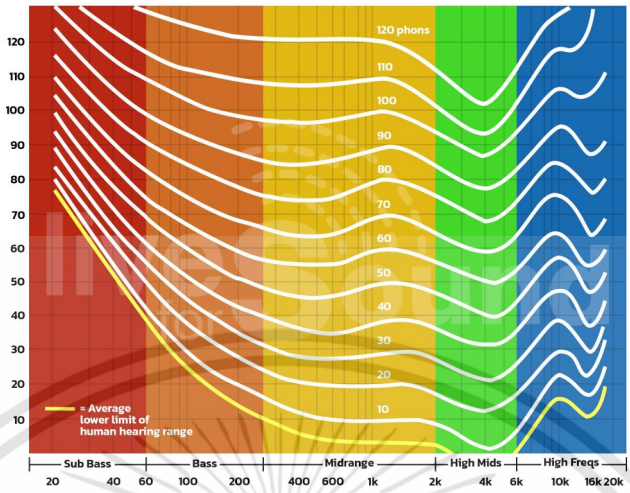


รูปที่ 2.7 กราฟ A-weighting, B-weighting, C-weighting, D-weighting [9]

2.4.2 Equal Loudness Contour

Equal Loudness Contour คือกราฟแสดงระดับการรับรู้ความถี่ของหูมนุษย์ที่ความดังต่าง ๆ หูมนุษย์นั้นจะรับรู้ความดังของแต่ละความถี่ไม่เท่ากัน ต่อให้เปิดความดังของความถี่ทั้งหมดมาเท่ากัน แต่หูจะรู้สึกว่าแต่ละความถี่มันดังไม่เท่ากัน โดยเป็นการทดลองการฟังของหูของมนุษย์ด้วยการใช้โทนเสียงความถี่เดียว ที่ความถี่ต่าง ๆ ที่มีความความเข้มเสียงเพิ่มขึ้นมากกว่า 10 dB ของเสียงที่ไปกระตุ้น โดยที่ผู้ทำการทดสอบยังคงฟังความถี่ 1000 Hz ควบคู่ไปกับความถี่อื่น ๆ ที่ป้อนเข้าไป เพื่อให้ระดับความดังของเสียงความถี่เดียว ที่ความถี่ต่าง ๆ มีค่าเท่ากับระดับความดังของความถี่เดียวที่ความถี่ 1000 Hz เพื่อสร้างเส้นกราฟที่แสดงความสัมพันธ์ระหว่างความดังกับความถี่ขึ้น

โดยปกติแล้วระบบการได้ยินของหูมนุษย์นั้น ได้ยินความถี่ตั้งแต่ 20 Hz ถึง 20000 Hz แม้ว่าขีดความสามารถการได้ยินย่านความถี่เสียงแหลมจะลดลงตามอายุที่มากขึ้น แต่ในส่วนของช่วงความถี่ 2 kHz ถึง 5 kHz จะยังคงรับรู้การได้ยินดีอยู่ ซึ่งความถี่ช่วง 2 kHz ถึง 5 kHz นี้ เป็นช่วงความถี่ที่หูมนุษย์จะมีความไวในการรับรู้มากที่สุด ส่วนใหญ่เกิดจากโครงสร้างของรูหูและกระดูกของหูชั้นกลางที่ให้ผลการสะท้อนของเสียงในช่องหูที่ความถี่ 2 kHz ถึง 5 kHz ได้มากกว่าความถี่อื่น



รูปที่ 2.8 กราฟ Equal Loudness Contour [10]

โดยกราฟ Equal Loudness Contour เส้นที่ 40 phon หรือ เส้นที่มีระดับความดัง 40 dB ที่ความถี่ 1000 Hz จะมีค่าใกล้เคียงกับกราฟ A-weighting ซึ่งจะแสดงว่าค่าระดับความดังบนกราฟ Equal Loudness Contour เส้นที่ 40 phon จะเป็นค่าที่เริ่มต้นการได้ยินของมนุษย์ ดังนั้นจึงจะนำมาเป็นค่าเริ่มต้นของการทดสอบการได้ยิน

2.4.3 dB SPL & dB HL

dB SPL หรือ Decibels in sound pressure level คือระดับความดันเสียงสามารถตรวจวัดได้ โดยมีค่าเท่ากันกับระดับความเข้มเสียง ซึ่งค่า SPL [4] นี้จะวัดจากค่าความเปลี่ยนแปลงความกดดันของตัวกลางโดยตรงเลย หน่วยเป็น Pascal แล้วเอามาเทียบกับค่าการเปลี่ยนแปลงความกดดันที่ใช้เป็นตัวอ้างอิงที่ 0 dB SPL (20 μPa หรือ 0.000020 Pascal) ภายใต้ข้อตกลงที่ว่า 0 dB SPL คือระดับ Threshold of Hearing หรือระดับความดังต่ำที่สุดที่คนส่วนใหญ่จะเริ่มได้ยินเสียง ค่า dB SPL จะถูกใช้กันมากในการวัดระดับความดังเสียง เพราะเป็นการวัดความกดดันเสียงที่เกิดขึ้นจริง ๆ ในอากาศ

dB HL หรือ Decibels in hearing level คือระดับความดังของเสียงที่หูมนุษย์ได้ยิน ซึ่งหน่วยนี้ส่วนใหญ่จะถูกใช้เป็นหน่วยของระดับความสูญเสียในกราฟ Audiogram

โดย dB SPL และ dB HL มีความสัมพันธ์กับกราฟ Equal Loudness Contour เนื่องจากในกราฟ Equal Loudness Contour จะแสดงค่าระดับความดังของเสียงเป็นหน่วย dB SPL และเมื่อสนใจที่เส้น 40 phon จะเห็นว่าที่ความถี่ 1000 Hz จะมีระดับความดังของเสียงที่ 40 dB SPL ซึ่งจะ

เป็นค่าระดับความดังของเสียงที่เริ่มได้ยิน จึงจะมีค่า 0 dB HL ด้วย ดังนั้นทุก ๆ ค่าบนเส้น 40 phon จะมีค่า 0 dB HL ที่ความถี่นั้น ๆ

2.4.4 การแปลงหน่วย Volt เป็นหน่วย dB SPL

เสียงที่จะนำมาทดสอบจะเป็นสัญญาณ Sinewave ที่ถูกกำหนดความดังไว้ เพื่อปรับความดังให้เหมาะสมกับการทดสอบการได้ยิน โดยจะถูกสร้างขึ้นด้วยสมการดังนี้

$$y(t) = A\sin(\omega t + \theta) \tag{2.1}$$

โดยการกำหนดระดับความดังของเสียง Sinewave จะถูกควบคุมด้วยแอมพลิจูดหรือ A ในสมการซึ่งจะมีหน่วยเป็น Volt แต่ในการทดสอบจะใช้หน่วย dB SPL เป็นหน่วยของระดับความดังของเสียง ซึ่งสามารถแปลงหน่วยด้วยสมการที่ 2.2 โดยในที่นี้ค่า x ของเครื่องโทรศัพท์ Samsung Galaxy J7 มีค่าเท่ากับ 93.9

$$\text{dB SPL} = 20\log\left(\frac{\text{Volt}}{0.05}\right) + x \tag{2.2}$$

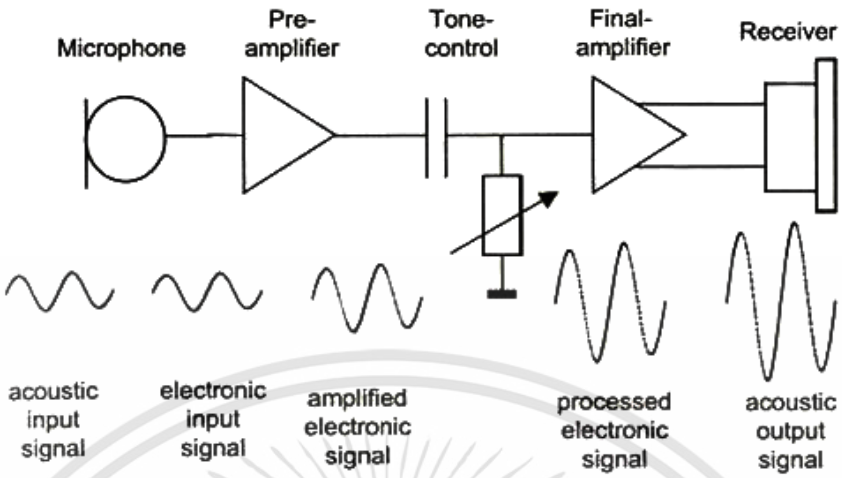
ดังนั้นในความถี่ 1000 Hz จะมีระดับความดังที่ 40 dB SPL หรือ 0 dB HL จะมีค่า Amplitude เท่ากับ 0.00010091831818407818 V ซึ่งจะนำไปใช้เป็นระกับเสียงที่ 0 dB HL

2.5 ศึกษาการทำงานของเครื่องช่วยฟัง

เครื่องช่วยฟัง คืออุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก ประกอบด้วยไมโครโฟนรับเสียง ชิปคอมพิวเตอร์ (Computer chip) ซึ่งทำหน้าที่จัดกระบวนการขยายเสียงด้วยระบบดิจิตอล และผลิตเป็นสัญญาณเสียงคุณภาพ ส่งไปที่ลำโพงเพื่อเข้าช่องหูให้ผู้ใช้ได้ยิน กระบวนการควบคุมการทำงานและผลิตคุณภาพเสียงที่ออกมาถูกควบคุมโดย ชิปคอมพิวเตอร์ (Computer chip) ที่มีตั้งแต่ระบบที่จัดการเสียงไม่ซับซ้อนมาก จนถึงระบบที่มีความซับซ้อนสูง เครื่องช่วยฟัง ดิจิตอล เป็นอุปกรณ์ที่ทำหน้าที่ขยายเสียงจากภายนอก ส่งเข้าไปในช่องหู ให้ผู้ที่มีการสูญเสียการได้ยินในระดับและรูปแบบที่แตกต่างกันสามารถได้ยินเสียงพูดและเสียงรอบข้างได้มากขึ้น อุปกรณ์เครื่องช่วยฟัง ทำงานด้วยแบตเตอรี่ใช้แล้วทิ้งแบบซิงค์แอร์ ซึ่งในปัจจุบันเป็นชนิดไร้สารปรอท (Mercury-Free Zinc Air) หรือแบบชาร์จไฟได้ [39]

2.5.1 เครื่องช่วยฟังแอนะล็อก (Analog Hearing Aid)

เป็นเครื่องช่วยฟังที่ใช้การขยายเสียงที่ใช้มาตรฐานทั่วไป ใช้เทคนิคที่ไม่ซับซ้อน ทำในการขยายเสียงที่เกิดขึ้นเป็นการขยายเสียงทุกเสียงที่ผ่านเข้ามาซึ่งอาจจะก่อให้เกิดความรำคาญแก่ผู้ใช้งานได้



รูปที่ 2.9 บล็อกไดอะแกรมแสดงโครงสร้างการทำงานของเครื่องช่วยฟังแอนะล็อก

2.5.2 เครื่องช่วยฟังดิจิทัล (Digital Hearing Aid)

เป็นการพัฒนาเครื่องช่วยฟังให้สามารถขยายเสียงให้มีคุณภาพเสียงที่ดีขึ้น และสามารถเลือกปรับขยายเสียงให้เป็นไปตามช่วงที่ต้องการได้ ในการชดเชยเสียงมีการคำนึงถึงการสูญเสียการได้ยินที่ต่างออกไปของผู้ใช้งาน โดยเครื่องช่วยฟังมีแอมพลิฟายเออร์เป็นไมโครชิพเหมือนชิพในคอมพิวเตอร์ สามารถที่จะโปรแกรมด้วยซอฟต์แวร์คอมพิวเตอร์ จึงสามารถทำงานได้เร็วมาก เพียงหนึ่งในแสนวินาที และทำงานได้หลายรูปแบบตามที่ออกแบบ

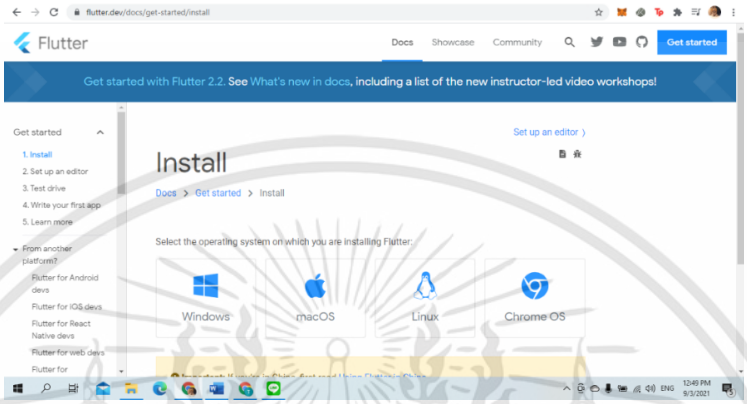
2.6 ภาษาโปรแกรม Flutter

2.6.1 โปรแกรม Flutter

Flutter เป็นชุดพัฒนาซอฟต์แวร์ (SDK: Software Development Kit) ในการพัฒนาแอปพลิเคชันสำหรับแอนดรอยด์ (Android) และ ไอโอเอส (IOS) บนโทรศัพท์มือถือ ภาษาที่ใช้ในการเขียน Flutter คือ ภาษา Dart

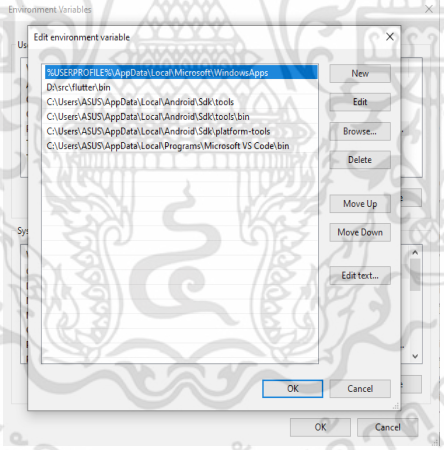
2.6.2 การติดตั้ง Flutter

2.6.2.1 เข้าไปที่ www.flutter.dev และทำการดาวน์โหลดไฟล์ติดตั้งโปรแกรม



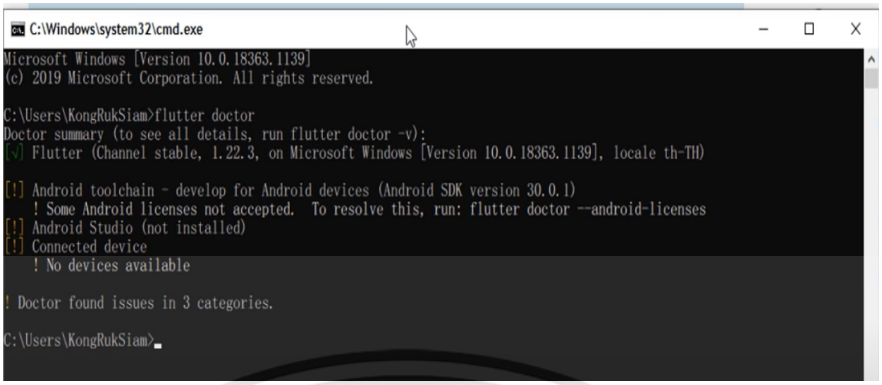
รูปที่ 2.10 หน้าเว็บ Flutter

2.6.2.2 เข้าไปที่ Environment และเข้าไปติดตั้ง Path C:\src\flutter\bin



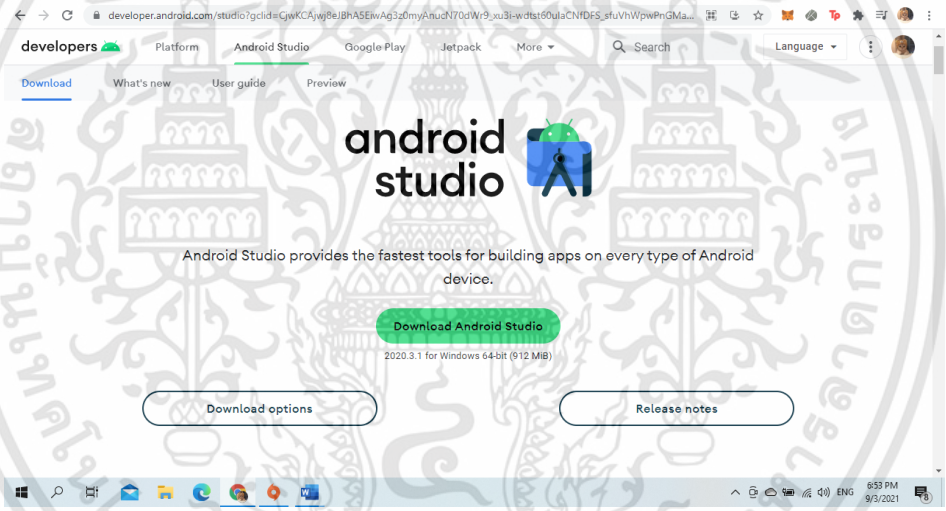
รูปที่ 2.11 หน้าต่างการติดตั้ง Path

2.6.2.3 เข้าไปที่ Command Windows แล้วพิมพ์ Flutter doctor เพื่อตรวจสอบโปรแกรมที่ยังไม่ได้ทำการติดตั้ง



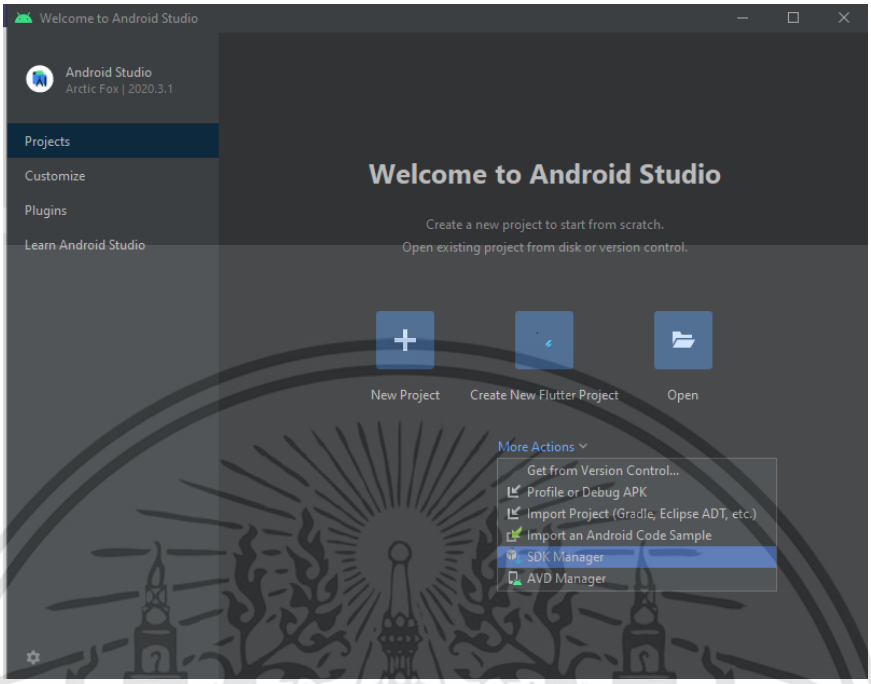
รูปที่ 2.12 Command Windows ที่ใช้ตรวจสอบการติดตั้ง

2.6.2.4 ทำการดาวน์โหลดไฟล์ติดตั้งตัวโปรแกรม Android Studio โดยไปเข้าไปที่เว็บ developer.android.com/ และทำการติดตั้งโปรแกรม



รูปที่ 2.13 หน้าเว็บ Android Studio

2.6.2.5 เข้าไปที่ตัวโปรแกรม Android Studio และเข้าไปที่ SDK Manager

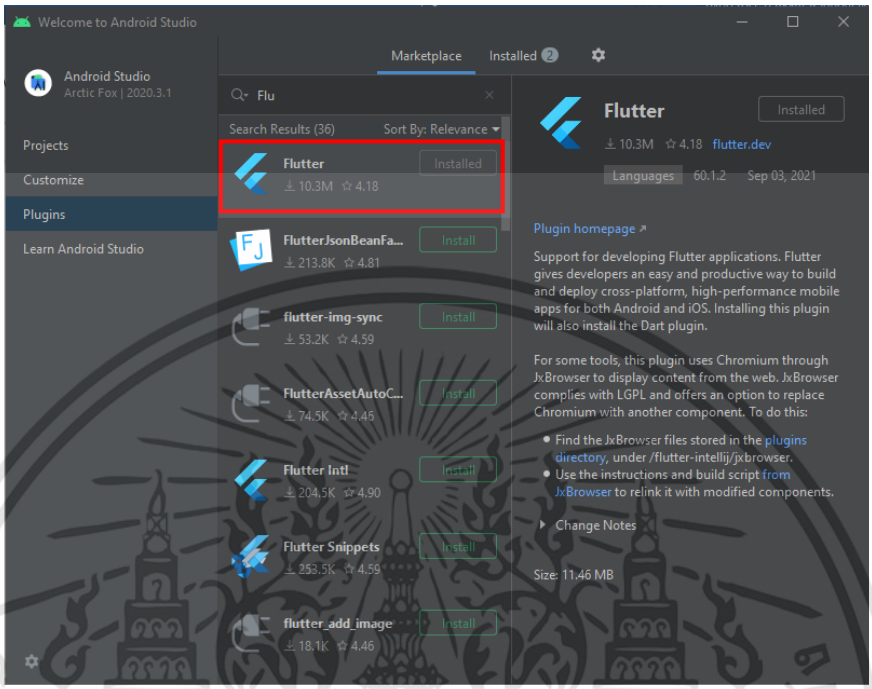


รูปที่ 2.14 หน้าแอพพลิเคชั่น Android Studio

2.6.2.6 ทำการติดตั้ง Android SDK รุ่นล่าสุด

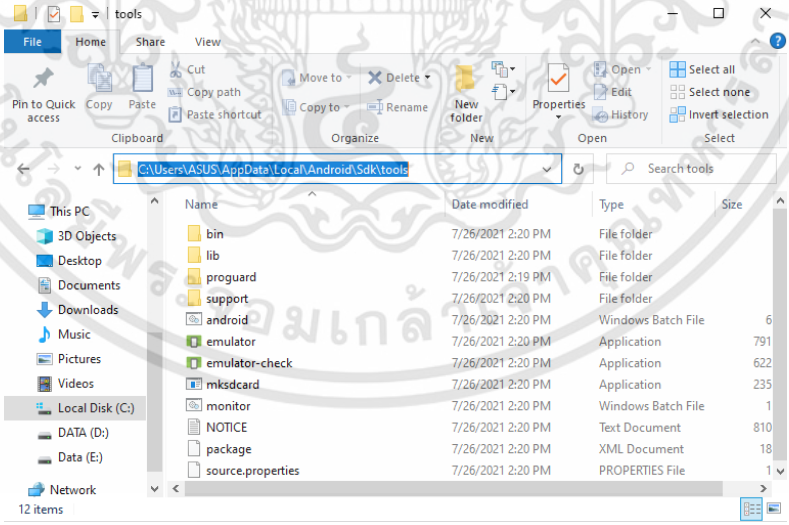
รูปที่ 2.15 หน้า Android SDK

2.6.2.7 เข้าไปที่ Plugins และทำการติดตั้ง Flutter

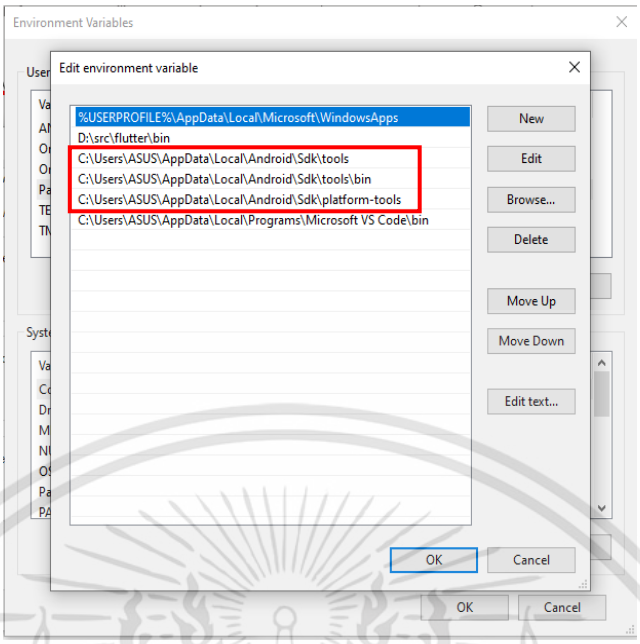


รูปที่ 2.16 หน้า Plugins

2.6.2.8 ทำการคัดลอกและติดตั้ง Path ที่ Environment ดังรูปที่ 2.17

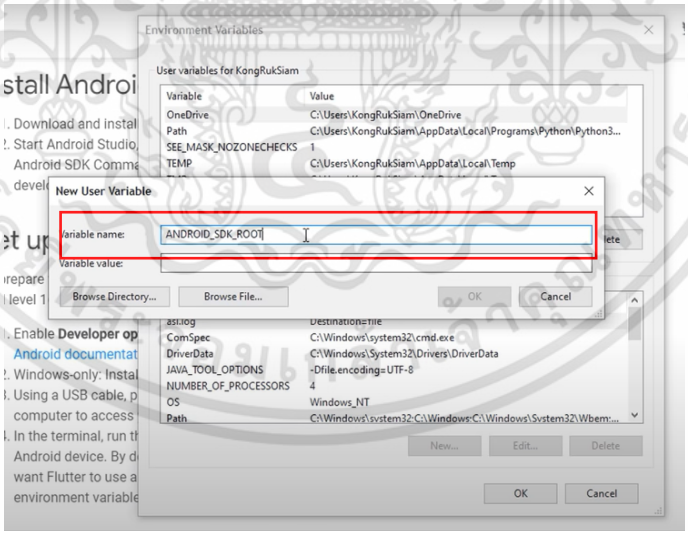


รูปที่ 2.17 การคัดลอก Path ที่ใช้จัดเก็บ

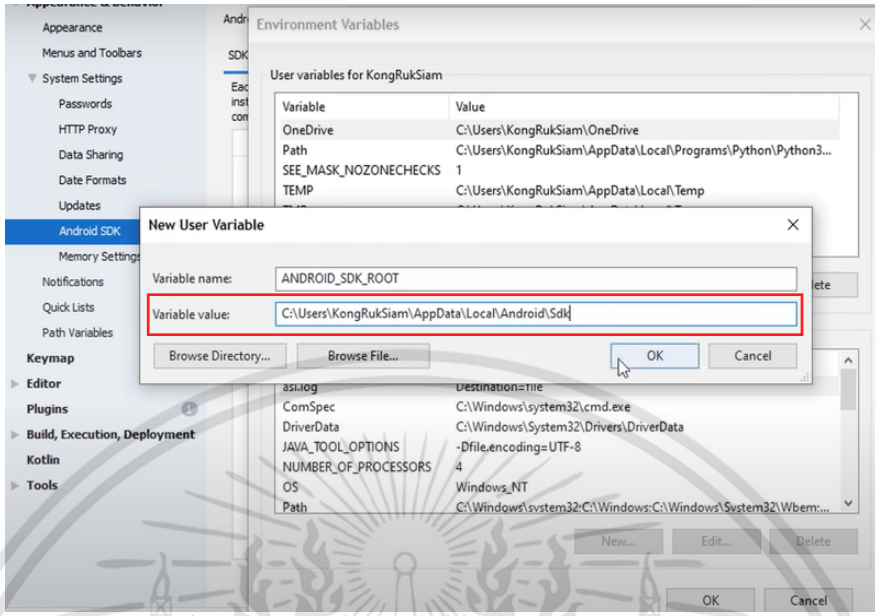


รูปที่ 2.18 การติดตั้ง Path ที่ Environment

2.6.2.9 ไปที่ Environment กดที่ปุ่ม ‘new’ ในหัวข้อ User variables ทำการตั้งชื่อตัวแปร ‘ANDROID_SDK_ROOT’ ดังรูปที่ 2.19 และทำการวาง path ของไฟล์ที่ติดตั้งคือ ‘C:\Users\ASUS\ AppData\Local\Android\Sdk’ ดังรูปที่ 2.20

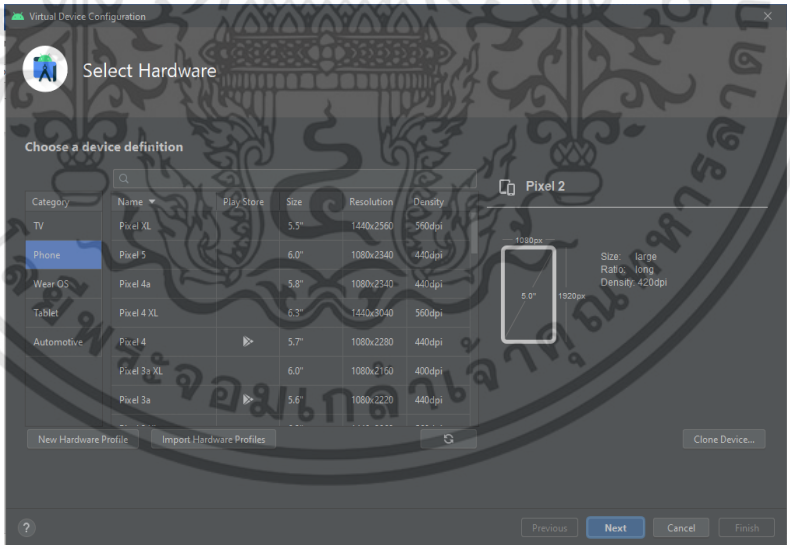


รูปที่ 2.19 ตัวอย่างการตั้งชื่อตัวแปร

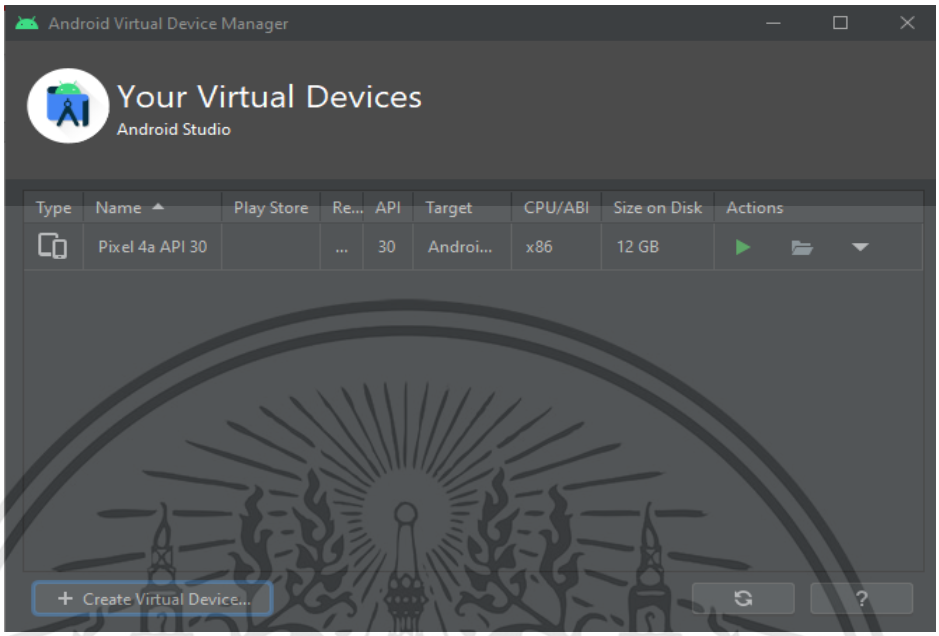


รูปที่ 2.20 ตัวอย่าง Path ที่ใส่ลงไปใน Environment

2.6.2.10 เข้าโปรแกรม Android Studio จากนั้นไปที่ AVD Manager เพื่อทำการสร้างอุปกรณ์จำลอง (Emulator) ขึ้นมา โดยเลือกที่ Phone แล้วเลือกรุ่นโทรศัพท์ ดังรูปที่ 2.21 จากนั้นจึงทำการดาวน์โหลด และสร้างอุปกรณ์ จะได้รายชื่ออุปกรณ์ที่จำลองขึ้นมาดังรูปที่ 2.22

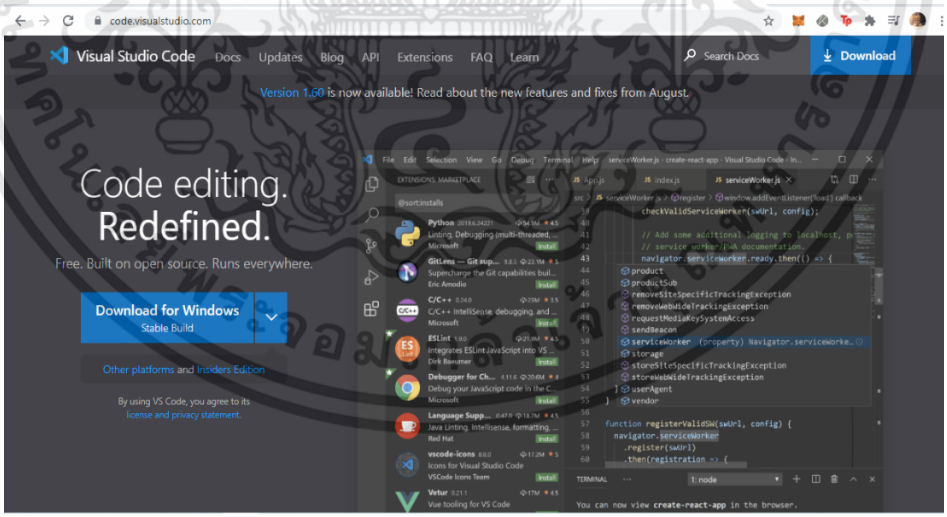


รูปที่ 2.21 ตัวอย่างอุปกรณ์จำลองที่ใช้

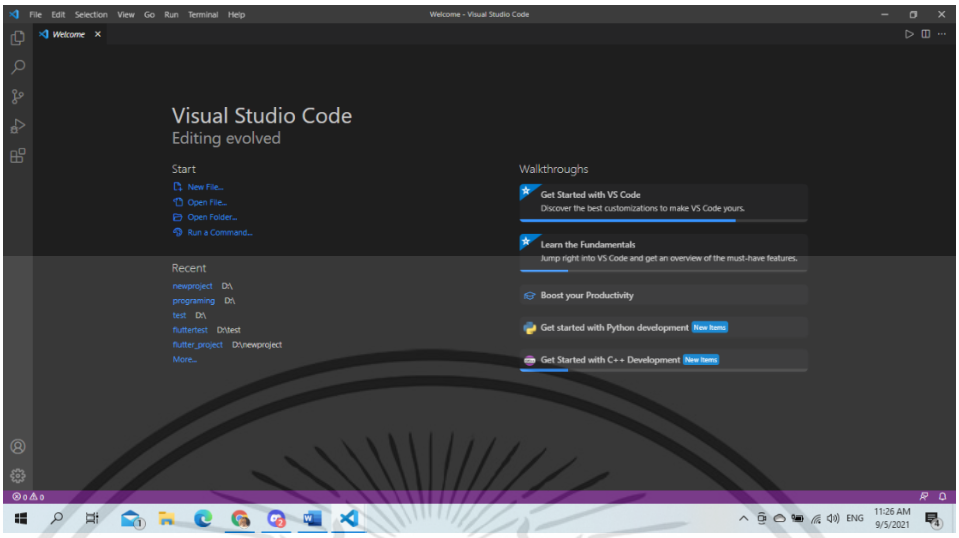


รูปที่ 2.22 รายชื่ออุปกรณ์จำลองที่สร้างขึ้น

2.6.2.11 เข้าไปที่เว็บ https://code.visualstudio.com/ ทำการดาวน์โหลดโปรแกรม Visual Studio Code แล้วทำการติดตั้งให้เรียบร้อยจะได้หน้าต่างโปรแกรมดังรูปที่ 2.24

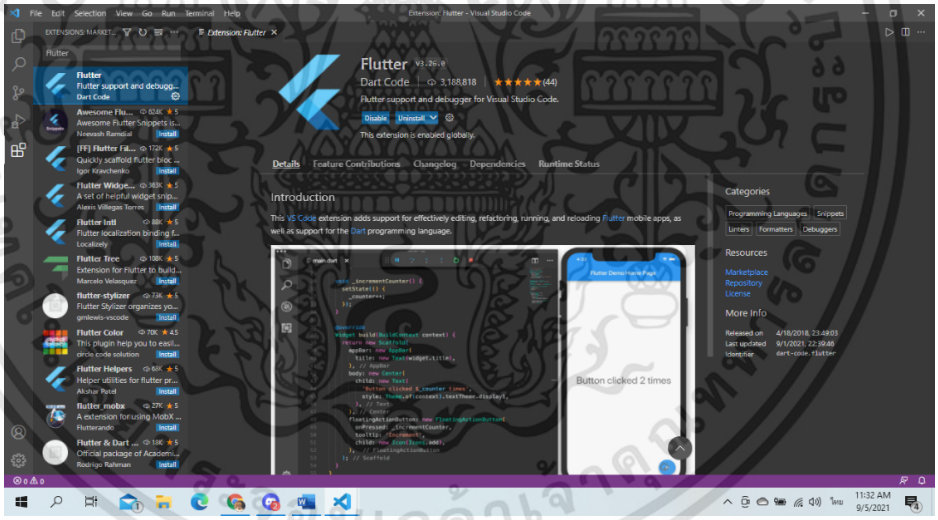


รูปที่ 2.23 เว็บไซต์ Visual Studio Code



รูปที่ 2.24 หน้าต่างโปรแกรม Visual Studio Code

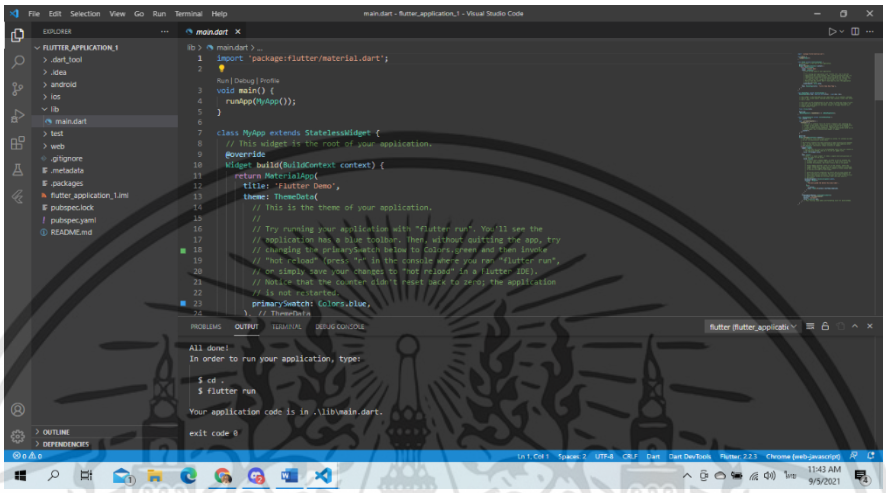
2.6.2.12 ทำการติดตั้งเครื่องมือ (Extension) ใน Visual Studio Code โดยไปที่ Extensions และค้นหา ‘Flutter ‘ทำการติดตั้งให้เรียบร้อย ดังรูป 2.25



รูปที่ 2.25 การติดตั้ง Flutter ใน Visual Studio Code

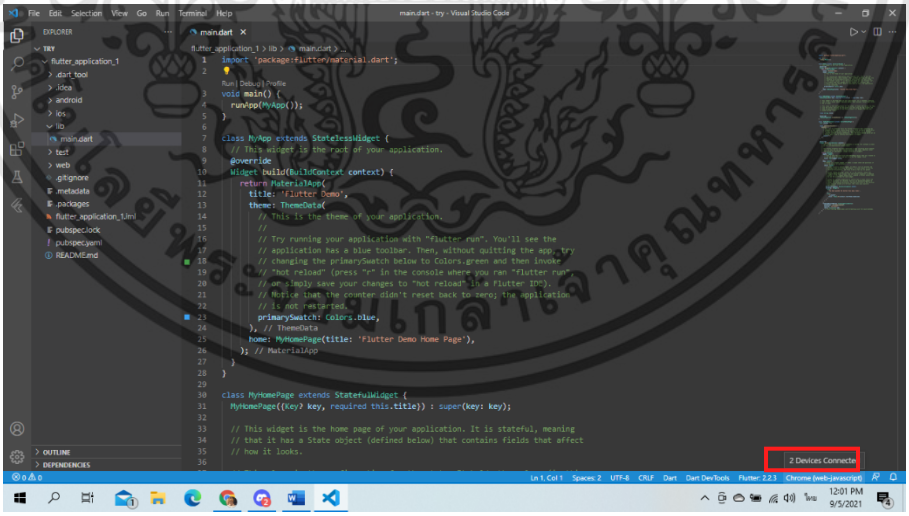
2.6.3 วิธีการสร้างโครงงาน Flutter

2.6.3.1 ไปที่ Visual Studio Code กด Ctrl + Shift + P และค้นหา ‘New Application Project’ เลือกแฟ้มข้อมูลที่จะใช้เก็บโครงงานแล้วทำการตั้งชื่อโครงงานจะได้ดังรูปที่ 2.26

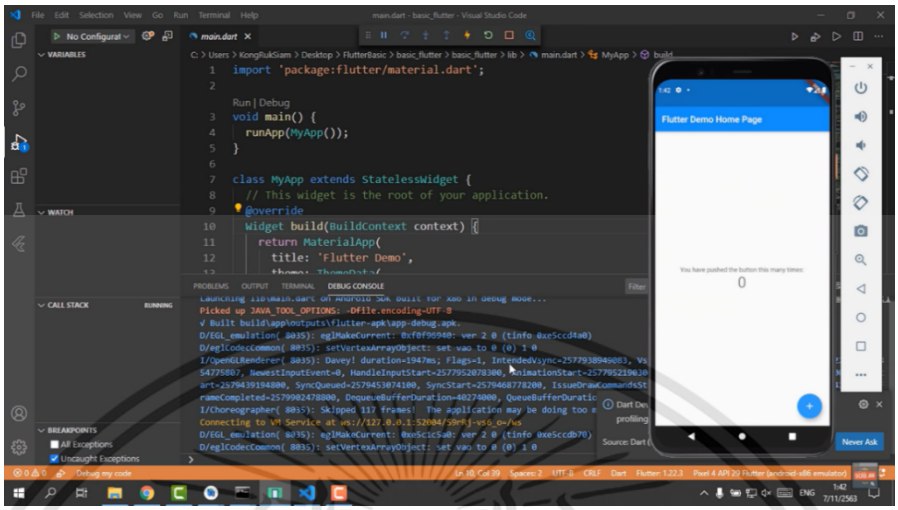


รูปที่ 2.26 หน้าโครงงาน Flutter

2.6.3.2 ทำการทดสอบโครงงานในอุปกรณ์จำลองโดยไปที่กรอบสีแดงดังรูปที่ 2.27 เพื่อเชื่อมต่อกับอุปกรณ์จำลอง จากนั้นทำการสั่งประมวลผลโปรเจคแสดงดังรูปที่ 2.28



รูปที่ 2.27 การเชื่อมต่ออุปกรณ์จำลองกับตัวโครงงาน



รูปที่ 2.28 ตัวอย่างโครงงานบนอุปกรณ์จำลอง

2.7 ศึกษาฐานข้อมูลโดยใช้บริการ Firebase

Firebase คือ แพลตฟอร์มที่รวบรวมเครื่องมือต่าง ๆ ไว้เพื่อจัดการกับระบบฐานข้อมูลประเภท NoSQL ทำให้ผู้พัฒนาแอปพลิเคชันบนโทรศัพท์หรือเว็บไซต์ก็ตามมีความสะดวกมากขึ้น โดย Firebase จะแบ่งได้ 3 หมวดหมู่ดังนี้

2.7.1 Build better apps

2.7.1.1 Realtime Database

เป็นบริการของ Firebase ที่สร้างบริการฐานข้อมูลแบบ NoSQL โดยมีวิธีการเก็บข้อมูลแบบ JSON และยังสามารถซิงค์ข้อมูลระหว่างผู้ใช้บนแอปพลิเคชันต่าง ๆ ได้อย่างเรียลไทม์ และยังสามารถทำงานในแบบออฟไลน์ได้แอปพลิเคชัน

2.7.1.2 Firebase ML (Beta)

เป็น Machine learning SDK เป็นบริการที่นำไปประยุกต์ใช้ช่วยให้แอปพลิเคชันทำงานสะดวก เช่น Text Recognition คือการช่วยแกะข้อความจากรูปภาพสลากกินแบ่งรัฐบาลแล้วนำไปตรวจผล

2.7.1.3 Cloud Function

เป็นบริการที่ทำงานในส่วนเซิร์ฟเวอร์เมื่อเกิดเหตุการณ์ที่เกิดขึ้นบน Firebase โดยฟังก์ชันที่ได้ทำการเขียนขึ้นจะถูกนำไปเก็บที่ Google Cloud ซึ่งมีความเสถียรและปลอดภัย

2.7.1.4 Authentication

เป็นบริการที่ไว้ตตรวจสอบผู้ใช้ โดยสามารถตรวจสอบได้ในหลายวิธี เช่น Email password, Facebook ,Twitter ,Google , Phone Number และอื่น ๆ ซึ่งจะมีฐานข้อมูลของตัวเองโดยไม่ต้องออกแบบสร้างฐานข้อมูลเพื่อจัดเก็บ

2.7.1.5 Hosting

เป็นบริการในการฝากไฟล์ เช่น JPG, CSS, HTML และอื่น ๆ เพื่อให้ผู้ใช้เข้าใช้งานในเว็บไซต์ที่สร้างขึ้นได้ ซึ่งบริการนี้จะนิยมใช้ในการฝากไฟล์ที่ได้จากสร้างของ JavaScript Framework ต่าง ๆ เช่น React, Angular จะเห็นว่าไฟล์ข้างต้นไม่จำเป็นที่จะต้องใช้ Framework และยังมี Content Delivery Network, Secure Socket Layer ซึ่งจะทำให้ผู้ใช้ปลอดภัยในการใช้งานและมีความรวดเร็วในการใช้

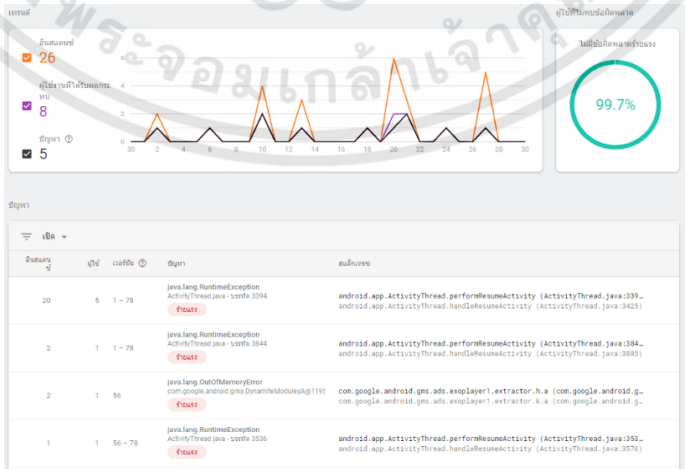
2.7.1.6 Cloud Storage

เป็นบริการในการเก็บไฟล์รูปภาพ, ไฟล์วิดีโอ, ไฟล์เสียง เพื่อนำมาใช้งานบนแอปพลิเคชัน เช่น วิดีโอที่ต้องการใช้บนแอปพลิเคชัน, รูปประจำตัวของสมาชิกและอื่น ๆ

2.7.2 Release & Monitor

2.7.2.1 Crashlytics

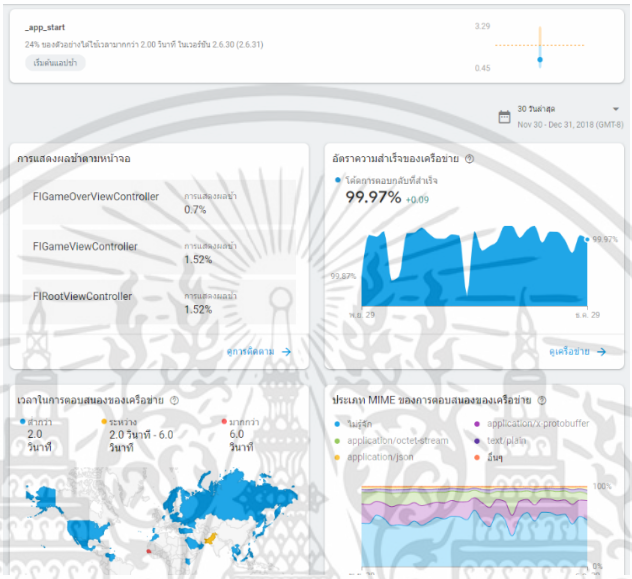
เป็นบริการใรการตรวจสอบและแจ้งเตือนหากแอปพลิเคชันเกิดมีอาการข้อผิดพลาดเพื่อให้แอปพลิเคชันมีความเสถียรจะมีการแจ้งเตือนให้ทราบถึงผลกระทบและข้อผิดพลาดผ่านทาง Firebase Console และ Email เพื่อมีการแก้ปัญหาที่รวดเร็ว



รูปที่ 2.29 หน้าการใช้งานบริการ Crashlytics [11]

2.7.2.2 Performance Monitoring

เป็นบริการในการตรวจสอบคุณภาพของแอปพลิเคชันเพื่อทำให้แอปพลิเคชันมีการตอบสนองได้เร็วอยู่เสมอ โดยจะมีการตรวจสอบในเวลาและรายละเอียดการทำงาน เช่น เวลาในการเปลี่ยนหน้า UI, ขนาดข้อมูลในการดาวน์โหลดหรืออัปโหลดและอื่น ๆ



รูปที่ 2.30 หน้าการใช้งานบริการ Performance Monitoring [11]

2.7.2.3 Test Lab

เป็นบริการในการทดสอบแอปพลิเคชันบนฮาร์ดแวร์ เพื่อทำให้ทราบว่าระบบแอปพลิเคชันสามารถรองรับฮาร์ดแวร์ที่ต้องการได้จริง ๆ สามารถระบุรุ่นที่ต้องการและระบุรุ่นในการทดสอบ

2.7.3 Engage

2.7.3.1 Remote Config

เป็นบริการที่ทำให้ควบคุมการเปลี่ยนแปลงลักษณะการทำงานหรือลักษณะการแสดงผลของแอปพลิชันของผู้ใช้แต่ละบุคคลได้ทันที เช่น เปลี่ยนภาษาตามผู้ใช้งาน

2.7.3.2 Google Analytics

เป็นบริการในการแสดงข้อมูลสถิติของแอปพลิเคชัน เช่น มีผู้ใช้งานกี่คน, มีการใช้งานที่ส่วนไหน เพื่อนำมาวิเคราะห์ในกลุ่มเป้าหมาย

2.7.3.3 Predictions

เป็นบริการในการวิเคราะห์ข้อมูลในการใช้งานแอปพลิเคชัน ทำให้ช่วยในการรู้ว่าผู้ใช้งานใช้งานในส่วนไหนบ้าง และส่วนไหนมีการตอบสนองได้ดีและส่วนไหนควรมีการปรับปรุง

2.7.3.4 A/B Testing

เป็นความสามารถมรกรแสดงผลแอปพลิเคชันหลายรูปแบบเพื่อทดสอบในการแสดงผลหรือการทำงาน เพื่อที่จะได้รู้ว่ามีการใช้งานให้ผู้ใช้งานได้ดีไหม

2.7.3.5 Cloud Messaging

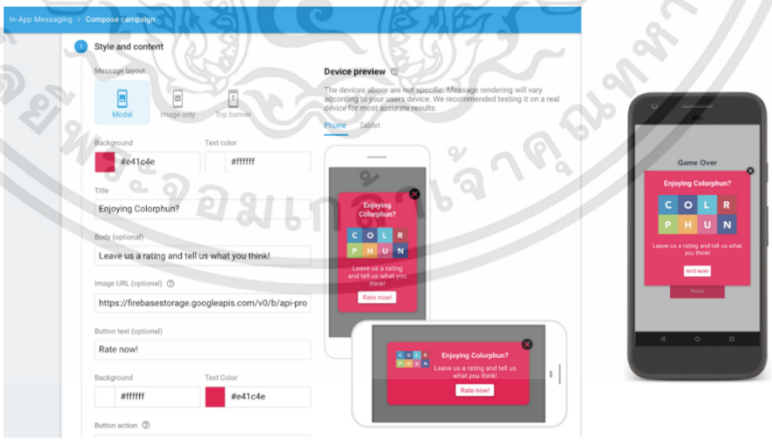
เป็นบริการส่งการแจ้งเตือนไปที่โทรศัพท์มือถือหรือบนเว็บไซต์เพื่อแจ้งข้อความไปยังผู้ใช้ เช่นการแจ้งเตือนจาก Email, การแจ้งเตือนจาก Facebook และอื่น ๆ

2.7.3.6 Dynamic Links

เป็นลิงก์ในการเชื่อมโยงไปยังแอปพลิเคชัน ใช้ในการแสดงบนหน้าเว็บเพื่อให้ผู้ใช้งานติดตั้งแอปพลิเคชันนี้ผ่านลิงก์

2.7.3.7 In-App Messaging

เป็นบริการในการแสดงข้อความขึ้นมาภายในแอปพลิเคชัน เช่น โฆษณา, ข่าวสาร, การแจ้งเตือน เป็นต้น



รูปที่ 2.31 หน้าการใช้งานบริการ In-App Messaging [11]

2.8 ภาษาโปรแกรม Python

2.8.1 ภาษาโปรแกรม Python

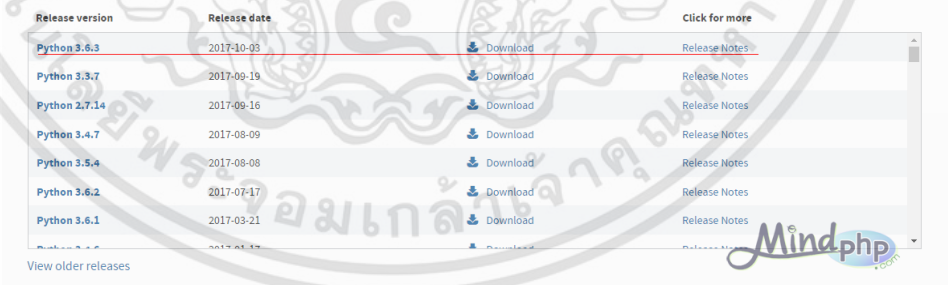
ภาษาโปรแกรม Python คือ ภาษาโปรแกรมคอมพิวเตอร์ระดับสูง โดยถูกออกแบบมาให้เป็นภาษาสคริปต์ที่อ่านง่าย โดยตัดความซับซ้อนของโครงสร้างและไวยกรณ์ของภาษาออกไป ในส่วนของการแปลงชุดคำสั่งที่เขียนให้เป็นภาษาเครื่อง Python มีการทำงานแบบ Interpreted คือเป็นการแปลชุดคำสั่งทีละบรรทัด เพื่อป้อนเข้าสู่หน่วยประมวลผลให้คอมพิวเตอร์ทำงานตามที่ต้องการ นอกจากนั้นภาษาโปรแกรม Python ยังสามารถนำไปใช้ในการเขียนโปรแกรมได้หลากหลายประเภท โดยไม่ได้จำกัดอยู่ที่งานเฉพาะทางใดทางหนึ่ง (General-purpose language) จึงทำให้มีการนำไปใช้กันแพร่หลายในหลายองค์กรใหญ่ระดับโลก เช่น Google, YouTube, Instagram, Dropbox และ NASA เป็นต้น [12]

2.8.2 การติดตั้งภาษาโปรแกรม Python

วิธีการติดตั้งภาษา Python ซึ่งเป็นแพ็กเกจของภาษาเพื่อให้โค้ดของ Python สามารถทำงานได้ และการติดตั้งเครื่องมือในการพัฒนาโปรแกรมซึ่งซอฟต์แวร์เหล่านี้นั้นฟรีจาก Python ได้

2.8.2.1 ดาวน์โหลดไฟล์สำหรับการติดตั้งโปรแกรม Python จาก www.python.org

2.8.2.2 เลือก "Python 3.6.3" จากนั้นกด "Download"



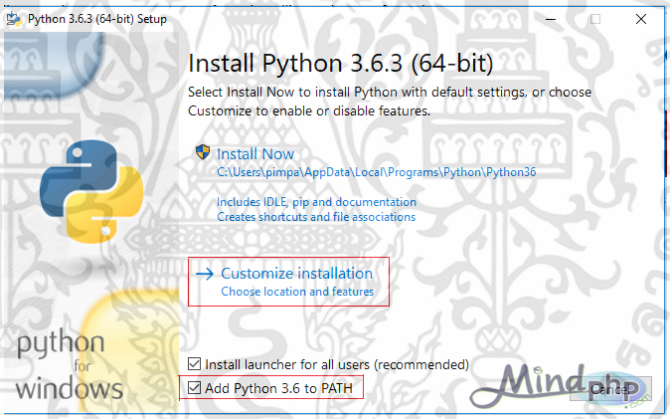
รูปที่ 2.32 ไฟล์การติดตั้งของ Python รุ่น 3.6.3 [13]

2.8.2.3 เมื่อกด "Download" ให้เลือก "Windows x86-64 executable Installer"

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		e9180c69ed9a878a4a8a3ab221e32fa9	22673115	SIG
XZ compressed source tarball	Source release		b9c2c36c33fb89bda1fefd37ad5af9be	16974296	SIG
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	ce31f17c952c657244a5cd0cccae34ad	27696231	SIG
Windows help file	Windows		a82270d7193f9fb8554687e7ca342df1	8020197	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64, not Itanium processors	b1daa2a41589d7504117991104b96fe5	7145844	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64, not Itanium processors	89044fb577636803bf49f36371dca09c	31619840	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64, not Itanium processors	b6d61642327f25a5ebd1a7f11a6d3707	1312480	SIG
Windows x86 embeddable zip file	Windows		cf1c75ad7ccf9dec57ba7269198fd56b	6388018	SIG
Windows x86 executable installer	Windows		3811c6d3203358e0c0c6b6677ae		SIG
Windows x86 web-based installer	Windows		39c2879cecf252d4c935e4		SIG

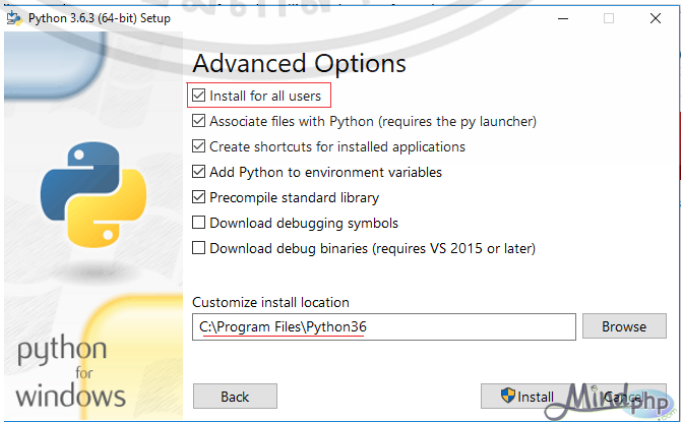
รูปที่ 2.33 เวอร์ชั่นที่ใช้ของไฟล์การติดตั้ง Python รุ่น 3.6.3 [13]

2.8.2.4 เมื่อโปรแกรมดาวน์โหลดเสร็จ จะขึ้นหน้าต่างการติดตั้ง เลือก "Add Python 3.6 PATH" และกดตรง "Customize installation"



รูปที่ 2.34 หน้าต่างการตั้งค่าหลังดาวน์โหลดโปรแกรม [13]

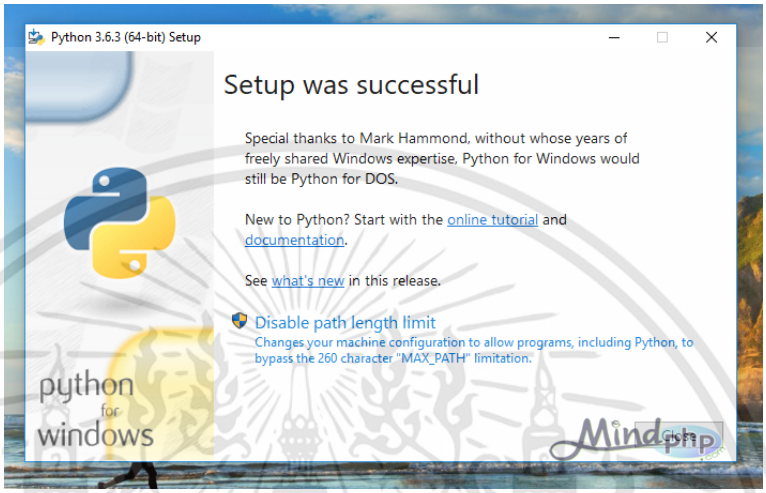
2.8.2.5 หลังจากนั้นคลิกที่ว่า "Install for all users"



รูปที่ 2.35 หน้าต่างการตั้งค่าสำหรับที่เก็บไฟล์ [13]

2.8.2.6 หลังจากนั้นรอให้โปรแกรมทำการติดตั้ง

2.8.2.7 เมื่อโปรแกรมติดตั้งเสร็จ จะแสดงหน้าต่าง "Setup was successful" แสดงถึงการติดตั้งโปรแกรมอย่างสมบูรณ์ จากนั้นให้กด "Close"



รูปที่ 2.36 หน้าโปรแกรม Python ที่ติดตั้งสำเร็จ [13]

2.8.2.8 หลังจากนั้น ทดลองหาโปรแกรม Python ถ้าขึ้นดังนี้รูปที่ 2.37 การติดตั้งโปรแกรมเป็นอันเสร็จสมบูรณ์

รูปที่ 2.37 โปรแกรม Python หลังจากติดตั้งเสร็จ [13]

2.9 Node.js

2.9.1 ภาษาโปรแกรม Node.js

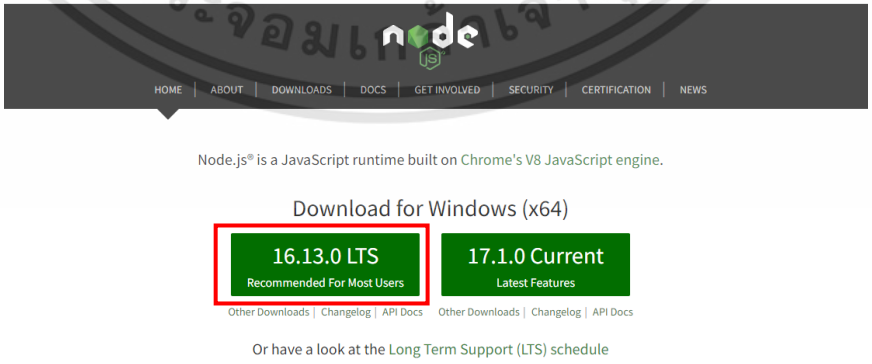
ภาษาโปรแกรม Node.js เป็นภาษาที่ใช้โครงสร้างภาษา JavaScript ถูกพัฒนาและทำงานโดยใช้ Chrome V8 engine ซึ่งเป็นการเพิ่มประสิทธิภาพสำหรับประมวลภาษา JavaScript ให้เป็นภาษาเครื่อง (Machine Language) ด้วยการประมวลแบบ Just-in-time (JIT) เพื่อเพิ่มประสิทธิภาพในการทำงานของภาษา JavaScript ซึ่งแต่เดิมเป็นภาษาที่มีการทำงานแบบ Interpreted ภาษาโปรแกรม Node.js สามารถใช้งานได้ทั้งบน Windows, Linux และ macOS X

รูปที่ 2.38 สัญลักษณ์ของภาษาโปรแกรม Node.js [14]

2.9.2 วิธีการติดตั้ง Node.js

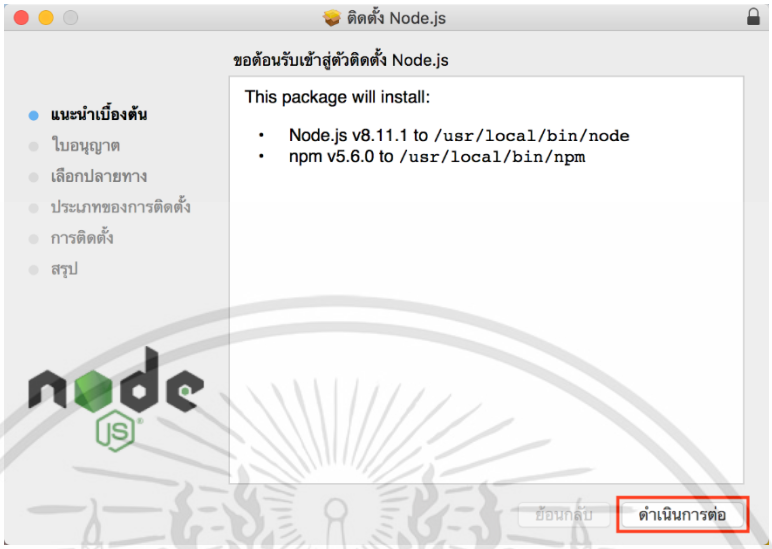
วิธีการติดตั้งภาษาโปรแกรม Node.js สามารถดำเนินการได้ตามขั้นตอนต่อไปนี้

2.9.2.1 เข้าเว็บไซต์ https://nodejs.org/en/ เพื่อทำการดาวน์โหลดโปรแกรม จากรูปที่ 2.39 ให้เลือก ‘Recommended for Most Users’



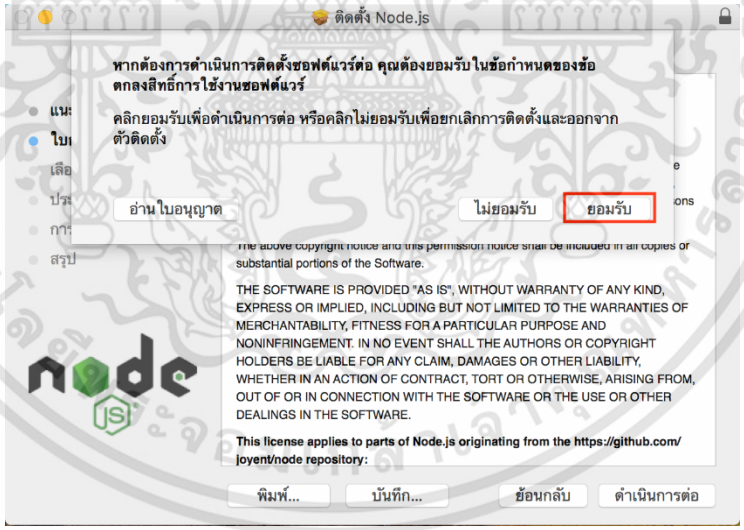
รูปที่ 2.39 เวอร์ชั่นที่ใช้ของภาษาโปรแกรม Node.js [14]

2.9.2.2 กด “ดำเนินการต่อ” เพื่อทำการติดตั้ง



รูปที่ 2.40 ขั้นตอนในการดำเนินการติดตั้งโปรแกรม Node.js [14]

2.9.2.3 กด “ดำเนินการต่อ” เพื่อทำการติดตั้ง เมื่อกดดำเนินการต่อเสร็จระบบจะแจ้งเตือนให้กด “ยอมรับ” เพื่อดำเนินการในขั้นตอนถัดไป

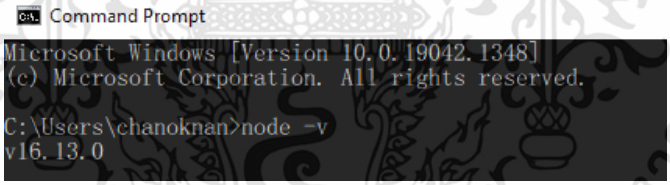


รูปที่ 2.41 ขั้นตอนในการดำเนินการติดตั้ง (ต่อ) [14]

2.8.2.4 จากนั้นกด “ติดตั้ง”

รูปที่ 2.42 ขั้นตอนในการดำเนินการติดตั้ง (ต่อ) [14]

2.9.2.5 เมื่อติดตั้งเสร็จแล้วให้ตรวจสอบว่าสามารถติดตั้งได้สำเร็จ โดยใช้คำสั่ง ‘node -v’ บน Command prompt



รูปที่ 2.43 ผลลัพธ์ที่ได้หลังจากการติดตั้งโปรแกรม Node.js

2.10 Message Queue Telemetry Transport (MQTT)

เป็นโปรโตคอลในการส่งข้อมูลระหว่าง Machine to Machine เป็นส่วนหนึ่งของเทคโนโลยี อินเทอร์เน็ตของสรรพสิ่ง (Internet of Thing: IOT) MQTT ได้พัฒนามาจาก สถาปัตยกรรมเครือข่ายคอมพิวเตอร์ชุดโปรโตคอล TCP/IP โดย MQTT จะมีโครงสร้างคือ Broker และ Client เป็นหลัก

รูปที่ 2.44 MQTT Client และ MQTT Broker [15]

2.10.1 MQTT Client

เป็นส่วนส่งข้อมูลที่ต้องการเผยแพร่ขึ้นไปยัง MQTT Broker และสามารถรับข้อมูลจาก MQTT Broker ผ่านทาง TCP/IP Protocol

2.10.2 MQTT Broker

เป็นซอฟต์แวร์สำหรับรับส่งข้อมูลจาก MQTT Client ที่ได้ส่งเข้ามาและสามารถส่งข้อมูลจาก MQTT Broker ไปยัง MQTT Client อื่น ๆ ที่ได้ลงชื่อติมตามกับ Client ที่เผยแพร่ได้

2.11 Google Cloud Platform (GCP)

Cloud Computing หรือ การประมวลผลแบบกลุ่มเมฆ เป็นลักษณะของการทำงานของผู้ใช้งานคอมพิวเตอร์ผ่านอินเทอร์เน็ต ที่ให้บริการใดบริการหนึ่งกับผู้ใช้ โดยผู้ให้บริการจะแบ่งปันทรัพยากรให้กับผู้ต้องการใช้งานนั้น

โดยสถาบันมาตรฐานและเทคโนโลยีแห่งชาติของสหรัฐอเมริกา (NIST) ได้ให้จำกัดความไว้ว่า เป็นรูปแบบที่ให้บริการเพื่อใช้ทรัพยากรคอมพิวเตอร์ร่วมกับผู้อื่นผ่านทางเครือข่ายเน็ตเวิร์ค เช่น โครงข่ายเน็ตเวิร์ค เซิร์ฟเวอร์ หน่วยเก็บสำรองข้อมูล แอพพลิเคชั่นและเซอร์วิสต่าง ๆ เป็นต้น โดยขึ้นอยู่กับความต้องการของผู้ใช้

2.11.1 ประเภทของ Cloud Computing

2.11.1.1 Public Cloud (คลาวด์แบบสาธารณะ) คือ คลาวด์ที่สร้างขึ้นเพื่อให้ทุกคนสามารถใช้งานได้ เช่น บริการคลาวด์ของอเมซอน Amazon Elastic Compute Cloud (EC2) หรือของ Google อย่าง Google App Engine

2.11.1.2 Private Cloud (คลาวด์ส่วนตัว) คือ คลาวด์ที่สร้างขึ้นเพื่อใช้ภายในองค์กรเองเท่านั้น ผู้ให้บริการและผู้ใช้สามารถควบคุมและปรับปรุงระบบความปลอดภัยได้ด้วยตนเอง

2.11.1.3 Hybrid Cloud (คลาวด์แบบผสม) คือ ระบบที่ผสมผสานระหว่าง Private Cloud และ Public Cloud ซึ่งเป็นการรวมเอาข้อดีของการทำงานทั้งสองแบบมาผสมผสานเพื่อตอบโจทย์การใช้งานเชิงธุรกิจ

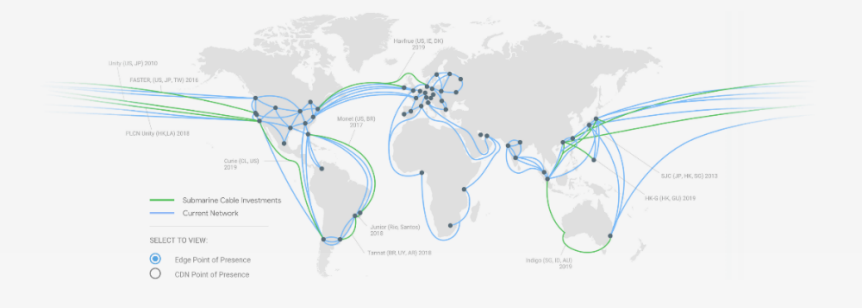
2.11.2 Google Cloud Platform

Google Cloud Platform หรือ GCP เป็นกลุ่มทรัพยากรในการประมวลผล (Computing Resource) ของ Google ที่เปิดให้ทุกคนสามารถเข้าใช้งานได้ในรูปแบบของ Public Cloud

รูปที่ 2.45 ตำแหน่งของเซิร์ฟเวอร์ของ GCP [27]

Google Cloud Platform สามารถใช้งานได้ทั่วโลก โดยสามารถเลือกใช้งานได้ตามพื้นที่และบริเวณ ทั้งนี้ขึ้นอยู่กับ ความหน่วง ความพร้อมใช้งาน และ ความทนทาน

พื้นที่จะขึ้นอยู่กับลักษณะทางภูมิศาสตร์ เช่น US East, US West, US Central, Europe, East Asia, Southeast Asia, South Asia หรือ Australia เป็นต้น และในแต่ละพื้นที่จะประกอบด้วยบริเวณ เช่นใน พื้นที่ของ Southeast Asia ที่ตั้งอยู่ในสิงค์โปร์ก็จะมี บริเวณ A และ บริเวณ B เป็นต้น



รูปที่ 2.46 เครือข่ายของเซิร์ฟเวอร์ของ GCP [27]

2.11.3 Services ของ Google Cloud Platform

Google Cloud Platform ที่ใช้ปริญญานิพนธ์จะใช้บริการในส่วน Compute และ Storage and Database ซึ่งสามารถแบ่งตามลักษณะการใช้งานได้ดังนี้

2.11.3.1 Compute

1) Compute Engine คือ Virtual Machine (VM) ที่ทำงานอยู่บนโครงสร้างพื้นฐาน (infrastructure) ของ Google Data Center

2) App Engine คือ บริการคลาวด์แบบ Platform as a Service (PaaS) ที่ให้นักพัฒนาสามารถ deploy เว็บแอพพลิเคชั่น โดยที่ไม่จำเป็นต้องกังวลกับการการจัดการโครงสร้างพื้นฐาน (infrastructure)

3) Kubernetes Engine คือ ระบบจัดการ Docker container ด้วย Kubernetes

4) Container registry คือ บริการที่ให้นักพัฒนาสามารถนำ Docker Image ที่สร้างขึ้นมาเก็บไว้บน Google Cloud

5) Cloud Functions คือ บริการที่ให้นักพัฒนาสามารถนำโปรแกรมมาใช้งานแบบ Serverless ได้ โดยไม่จำเป็นต้องจัดการกับเครื่องเซิร์ฟเวอร์

2.11.3.2 Storage & Databases

1) Cloud Storage คือ บริการสำหรับจัดเก็บไฟล์เอกสารหรือข้อมูลต่าง ๆ

2) Cloud SQL คือ ระบบจัดการฐานข้อมูลแบบ Relational Database ที่เข้าถึงข้อมูลและจัดการข้อมูลด้วยภาษา SQL โดยรองรับการใช้งานทั้ง PostgreSQL และ MySQL

3) Cloud Bigtable คือ ระบบจัดการฐานข้อมูลแบบ Non-Relational Database หรือ NoSQL โดยรองรับ workload ขนาดใหญ่ (หลายร้อย Petabytes) ซึ่งให้ Latency ที่ต่ำและ Throughput ที่สูง

4) Cloud Spanner คือ Global Distributed Relational Database Service ที่ทำงานอยู่บน Google Cloud Platform โดยดึงเอาจุดเด่นของ Relational Database และ NoSQL เข้ามารวมกัน ทำให้ได้ฐานข้อมูลที่มีความสามารถสูง รองรับ ACID Transaction, Relational Schemas, SQL Queries, High Availability และยังรองรับการ Scale ระบบให้มีขนาดใหญ่จนถึงหลักพันเซิร์ฟเวอร์ได้

5) Cloud Datastore คือ บริการ NoSQL Database เช่นเดียวกับ Cloud Bigtable เพียงแต่มีการออกแบบให้รองรับ Datasets ที่มีขนาดเล็กกว่า เพื่อใช้งานกับ Web หรือ Mobile Applications

6) Persistent Disk คือ Persistent Storage ที่มีทั้งแบบ SSD และ HDD

7) Cloud Memory store คือ บริการสำหรับ in-memory data store ที่ใช้ Redis ในการจัดการ

2.12 DigitalOcean

DigitalOcean คือ บริการที่ให้ใช้เช่าพื้นที่ของคลาวด์เซิร์ฟเวอร์ โดยที่สามารถสร้างเครื่องเสมือน (Virtual Machine) ขึ้นมาใช้งานได้ นอกจากนี้ยังสามารถสร้างเครื่องเพิ่มเติมได้ตามความต้องการของผู้ใช้งาน (Droplet) สามารถจัดการกับเครื่องได้ทุกอย่างและสามารถเพิ่มเครื่องตามที่ต้องการได้ ซึ่งบริการ DigitalOcean เป็นบริการที่ใช้งานได้ง่าย สะดวก และ รวดเร็ว ราคาไม่สูงเมื่อเทียบกับผู้ให้บริการคลาวด์เซิร์ฟเวอร์รายอื่น ๆ

รูปที่ 2.47 บริการ DigitalOcean [29]

โดยในปริญญานิพนธ์นี้เลือกใช้บริการประเภท Droplet เพื่อใช้ในการสร้างคลาวด์เซิร์ฟเวอร์ Droplet คือ ระบบปฏิบัติการที่ทำงานบนฮาร์ดแวร์เสมือนจริง ซึ่งเมื่อทำการสร้าง Droplet เสร็จระบบของ DigitalOcean จะทำการกำหนดไอพีแอดเดรสสาธารณะ (Public IP Address) ให้คลาวด์เซิร์ฟเวอร์นั้น ๆ ต่างจากบริการของ AWS ที่ต้องมีการผูกไอพีแอดเดรสขึ้นมาเอง จึงถือได้ว่าเป็นข้อดีหลักของบริการ DigitalOcean [29]

2.13 การสร้างเครื่องช่วยฟังแบบดิจิทัลโดยวงจรกรองถี่แบบดิจิทัล

การสูญเสียการได้ยินสามารถชดเชยได้โดยการปรับคลื่นความถี่ย่อยของตัวกรองที่สม่ำเสมอหรือไม่สม่ำเสมอแต่อุปกรณ์ช่วยฟังที่มีความแม่นยำสูงนั้นต้องการแถบความถี่ที่เพิ่มขึ้นตามไปด้วย เนื่องจากรูปแบบการสูญเสียการได้ยิน (audiograms) มีความต่างกันในแต่ละบุคคล และวงจรกรองความถี่แบบดิจิทัลต้องสามารถปรับความยืดหยุ่นได้เพื่อทำให้เหมาะสมกับการสูญเสียการได้ยินรูปแบบต่าง ๆ โดยในการออกแบบเครื่องช่วยฟังดิจิทัลจะคำนึงถึงการใช้พลังงานต่ำและการปรับความยืดหยุ่นของแถบความถี่ได้สูง แต่จะทำให้มีโครงสร้างของวงจรกรองความถี่แบบดิจิทัลมีความซับซ้อนและทำให้พลังงานเพิ่มมากขึ้น ดังนั้นจึงต้องมีการออกแบบวงจรกรองความแบบดิจิทัลที่มีโครงสร้างที่ไม่ซับซ้อนแต่มีความแม่นยำสูงสำหรับการสูญเสียการได้ยินในรูปแบบต่าง ๆ [28]

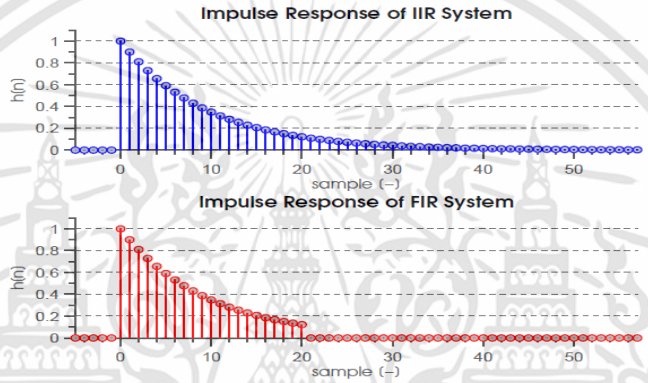
ในการออกแบบวงจรกรองความถี่แบบดิจิทัล ประกอบไปด้วยประเภทของวงจรกรองความถี่อยู่ 2 ประเภทได้แก่

2.13.1 Finite Impulse Response (FIR)

คือ วงจรกรองความถี่ที่มี Impulse response เกิดขึ้นด้วยจำนวนจำกัด โดย Transfer function จะมีค่าสัมประสิทธิ์ของ zero เท่านั้น ซึ่งมีข้อดีคือ สามารถออกแบบให้ผลตอบสนองความถี่เฟสเชิงเส้นได้ง่าย มีความเสถียรภาพเสมอ แต่มีข้อเสียคือ ผลตอบสนองความถี่ (Frequency response) ไม่ดีเท่า IIR ที่อันดับเท่า ๆกัน และไม่สามารถใช้ตัวกรองแอนะล็อก (Analog filter) ในการออกแบบได้

2.13.2 Infinite Impulse Response (IIR)

คือ วงจรกรองความถี่ที่มี Impulse response จำนวนที่ไม่สิ้นสุด โดย Transfer function จะมีสัมประสิทธิ์ทั้งค่า pole และ zero อยู่ด้วย ซึ่งจะทำให้วงจรมีความยาวที่ไม่สิ้นสุด โดยมีข้อดีคือ วงจรกรองความถี่แบบ IIR สามารถที่จะทำให้ได้คุณสมบัติของวงจรกรองความถี่ ที่เป็นไปตามที่ต้องการด้วยอันดับของวงจรกรองความถี่ที่ต่ำกว่าการออกแบบ โดยใช้ตัวกรองความถี่แบบ FIR ซึ่งจะมีผลทำให้โครงสร้างที่จะใช้ในการดำเนินงานมีความซับซ้อนน้อยกว่า และข้อเสียคือ ประสิทธิภาพของการคำนวณในรูปแบบที่เป็น finite-length arithmetic ซึ่งมีผลทำให้เกิดสัญญาณรบกวนเกิดขึ้นจากการคำนวณ



รูปที่ 2.48 Impulse response ของ IIR และ FIR [16]

ซึ่งในการออกแบบวงจรกรองความถี่ที่นำมาใช้ในเครื่องช่วยฟัง จะใช้วงจรกรองความถี่แบบดิจิทัลประเภท IIR เนื่องจากมีการคำนวณที่ซับซ้อนน้อยกว่า และมีประสิทธิภาพมากกว่า FIR ที่อันดับเท่า ๆกัน

2.14 สมการไบควอตเดรติก (Biquadratic Equation)

สมการไบควอตเดรติกเป็นสมการของทางคณิตศาสตร์ ซึ่งเป็นฟังก์ชันกำลังสี่ที่จะไม่มีกำลังคี่ในสมการโดยจะนำมาจัดอยู่ในรูปแบบฟังก์ชันกำลังสอง ซึ่งในปริญญานิพนธ์นี้ใช้วงจรกรองสัญญาณแอนะล็อกต้นแบบที่ใช้สมการไบควอตเดรติก โดยจะใช้วงจรกรองสัญญาณแอนะล็อก 3 วงจร คือ วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter), วงจรกรองแถบความถี่ผ่าน (Band Pass Filter) และวงจรกรองความถี่สูงผ่าน (High Pass Filter) ซึ่งแต่ละวงจรจะได้ฟังก์ชันการถ่ายโอน H(s) ดังนี้

1) วงจรกรองสัญญาณความถี่ต่ำผ่าน

$$H(s) = \frac{\Omega_{LPF}^2}{\Omega_{LPF}^2 + \left(\frac{\Omega_{LPF}}{Q_{LPF}}\right)s + s^2} \tag{2.3}$$

2) วงจรกรองสัญญาณแถบความถี่ผ่าน

$$H(s) = \frac{\left(\frac{\Omega_{BPF}}{Q_{BPF}}\right)s}{\Omega_{BPF}^2 + \left(\frac{\Omega_{BPF}}{Q_{BPF}}\right)s + s^2} \tag{2.4}$$

3) วงจรกรองสัญญาณความถี่สูงผ่าน

$$H(s) = \frac{s^2}{\Omega_{HPF}^2 + \left(\frac{\Omega_{HPF}}{Q_{HPF}}\right)s + s^2} \tag{2.5}$$

โดย Ω_{LPF}, Ω_{BPF} และ Ω_{HPF}จะสัมพันธ์กับค่าความถี่ศูนย์กลาง (Center Frequency) โดย $\Omega = \frac{\pi f}{f_s}, f_s$ คือ ค่าความที่สุ่ม (Sampling frequency) และ Q คือ ตัวประกอบคุณภาพ (Quality factor) ของแต่ละวงจร

2.15 การแปลงไบลิเนียร์สำหรับวงจรกรองสัญญาณดิจิทัล

การออกแบบวงจรกรองสัญญาณแบบดิจิทัลจะได้ค่าสัมประสิทธิ์จากวงจรกรองสัญญาณแอนะล็อกต้นแบบซึ่งจะนำมาแปลงเป็นวงจรกรองสัญญาณดิจิทัล จะใช้หลักการแปลงไบลิเนียที่จะทำการแปลงสมการ S-Plane เป็น Z-Plane และใช้ปาสคาลเมตริกซ์ (Pascal Matrix) ในการหาสัมประสิทธิ์จากวงจรกรองสัญญาณดิจิทัล โดยสรุปได้ดังสมการที่ 2.6

$$H(s) \rightarrow H(z) \tag{2.6}$$

โดย H(s) คือฟังก์ชันการถ่ายโอนของวงจรกรองสัญญาณแอนะล็อกต้นแบบ

H(z) คือฟังก์ชันการถ่ายโอนของวงจรกรองสัญญาณดิจิทัลจากการแปลงไบลิเนียร์

หลังจากที่ทำการแปลงไบลิเนียร์ก็จะทำการใช้ปาสคาลเมตริกซ์ในการหาสัมประสิทธิ์ของเทอมเศษจากสมการ และเทอมส่วนจากสมการ

$$[a_i] = [P_{i,j}][A_i] \tag{2.7}$$

$$[b_i] = [P_{i,j}][B_i] \tag{2.8}$$

โดย a_i คือสัมประสิทธิ์เทอมเศษของวงจรกรองสัญญาณดิจิทัล

b_i คือสัมประสิทธิ์เทอมส่วนของวงจรกรองสัญญาณดิจิทัล

A_i คือสัมประสิทธิ์เทอมเศษของวงจรกรองสัญญาณแอนะล็อก

B_i คือสัมประสิทธิ์เทอมเศษของวงจรกรองสัญญาณแอนะล็อก

$P_{i,j}$ คือปาสคาลเมทริกซ์

โดย ปาสคาลเมทริกซ์มีองค์ประกอบเป็น

$$P_{i,j} = \sum_{n=0}^{i} \binom{N-j}{n} \binom{i}{i-n} (-1)^{i-n} \; ; i, j = 0,1 \dots, N \tag{2.9}$$

จากฟังก์ชันถ่ายโอนของวงจรแอนะล็อกต้นแบบมีอันดับวงจรเท่ากับ 2 ซึ่งจะใช้ปาสคาลเมตริกซ์ขนาด 3x3 สำหรับในการแปลงไบลิเนียร์

เมื่อแทนค่า N = 2 ในสมการจะได้ปาสคาลเมตริกซ์ที่ใช้ในการแปลงไบลิเนียร์ดังนี้

$$P = \begin{bmatrix} P_{0,0} & P_{0,1} & P_{0,2} \\ P_{1,0} & P_{1,1} & P_{1,2} \\ P_{2,0} & P_{2,1} & P_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix} \tag{2.10}$$

หลังจากทำการแปลงไบลิเนียและใช้ปาสคาลเมตริกซ์จะได้ฟังก์ชันการถ่ายโอนดิจิทัล

1) วงจรกรองสัญญาณดิจิทัลความถี่ต่ำผ่าน

$$H_{LPF}(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{b_0 + b_1 z^{-1} + b_2 z^{-2}} \tag{2.11}$$

2) วงจรกรองสัญญาณดิจิทัลแถบความถี่ผ่าน

$$H_{BPF}(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{b_0 + b_1 z^{-1} + b_2 z^{-2}} \quad (2.12)$$

3) วงจรกรองสัญญาณดิจิทัลความถี่สูงผ่าน

$$H_{HPF}(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{b_0 + b_1 z^{-1} + b_2 z^{-2}} \quad (2.13)$$

เมื่อนำทั้ง 3 วงจรมาต่อขนานกันจะมีการคำนวณที่ใช้วงจรคูณ 9 ตัวและวงจรบวกอีก 12 ตัว

2.16 การหาค่าที่เหมาะสมที่สุดและ Nelder-Mead algorithm

2.16.1 การหาค่าที่เหมาะสมที่สุด (Optimization)

เป็นวิธีการคณิตศาสตร์ ซึ่งเป็นการเรียนรู้เพื่อกำหนดวิธีการที่ดีที่สุดให้กับปัญหา (การหาค่าที่เหมาะสมที่สุด) ในการหาค่าสูงสุดหรือค่าต่ำสุด โดยเป็นรูปแบบของฟังก์ชันทางคณิตศาสตร์ จะแบ่งกระบวนการเป็น 3 ส่วน

2.16.1.1 ตัวแปรออกแบบ (Design Variable)

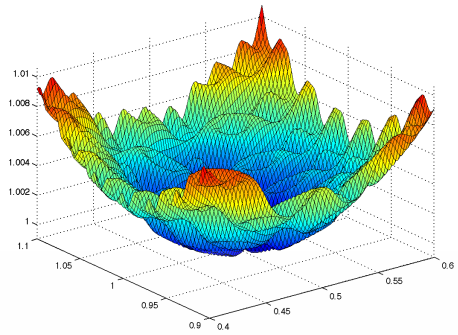
หมายถึง ตัวแปรที่เป็นคำตอบของการแก้ปัญหาที่เหมาะสมที่สุด ซึ่งตัวแปรออกแบบจะถูกกำหนดเพื่อใช้อธิบายลักษณะของระบบทางด้านวิศวกรรมอย่างชัดเจน

2.16.1.2 ฟังก์ชันวัตถุประสงค์ (Objective Function)

หมายถึง ฟังก์ชันที่ต้องการหาค่าต่ำสุดหรือสูงสุด โดยจะต้องทำการกำหนดฟังก์ชันวัตถุ ประสงค์ให้อยู่ในรูปของสมการทางคณิตศาสตร์ที่ติดอยู่ในรูปของตัวแปรออกแบบ เพื่อที่จะทำการหาค่าของตัวแปรที่เป็นจุด maximum หรือ minimum ของฟังก์ชันวัตถุประสงค์นั้น

2.16.1.3 ข้อจำกัด (Constraints)

หมายถึง เงื่อนไขหรือข้อจำกัดของ objective function เพื่อเป็นขอบเขตของการทำ optimization เมื่อต้องการจำกัดค่าของตัวแปรที่ต้องการหา

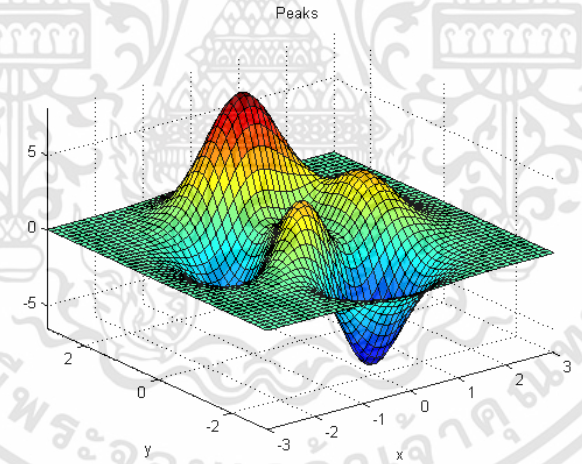


รูปที่ 2.49 ผลจากการ optimization ที่เป็นไปได้ [17]

จากรูปที่ 2.49 จะเห็นได้ค่าของการ optimization จะมีค่าต่ำสุด ซึ่งจุดนั้นจะเป็นจุดที่มีต่ำสุดในปัญหานี้

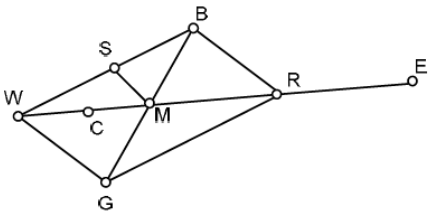
2.16.2 Nelder-Mead algorithm

ในการ Optimization จะเป็นไปได้ยากถ้าหากการหาค่าต่ำที่สุดมีลักษณะเป็นหลุม และเกิดหลุมมากกว่า 1 ที่ แสดงดังรูปที่ 2.50 จะทำให้การหา Optimization จะไม่ได้ค่าดีที่สุดที่ควรจะเป็น

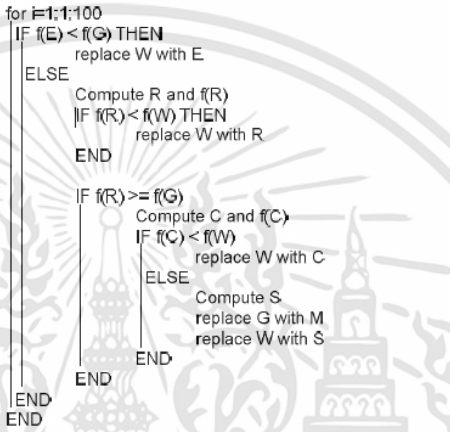


รูปที่ 2.50 การ Optimization ที่มีหลุมมากกว่า 1 ที่ [18]

จึงมีการค้นหาจุดที่น้อยที่สุดของ Objective Function โดยมีการดำเนินการด้วยการสุ่มค่ารอบ ๆ จุดเริ่ม 3 จุดประกอบกับการ Optimized เพื่อคำนวณหาค่าและทำการวนซ้ำไปเรื่อย ๆ โดยจะสุ่มค่าที่อยู่ในระยะที่สั้นลงของจุดเริ่มนั้น จนกว่าจะทำให้ค่าของที่นำไปใช้ใน function มีค่าต่ำที่สุด



รูปที่ 2.51 การหาจุดต่าง ๆ ด้วยวิธี Nelder-Mead algorithm [19]



รูปที่ 2.52 กระบวนการของ Nelder-Mead algorithm [19]

จากรูปที่ 2.52 อธิบายได้ว่า เมื่อทำการสุ่มค่าขึ้นมา 3 ค่า โดยค่าที่ทำการสุ่มจะถูกเก็บอยู่ในตัวแปร B,G,W โดย B ย่อมาจาก Best point, G ย่อมาจาก Good point, W ย่อมาจาก Worse point โดยค่า B เป็นค่าที่ทำให้ค่าในฟังก์ชันมีค่าน้อยที่สุด ส่วน G เป็นค่าที่ทำให้ฟังก์ชันมีค่ามากกว่า B แต่น้อยกว่า W และ W เป็นค่าที่ทำให้ฟังก์ชันมีค่ามากที่สุดในการสุ่มค่านี้ จากนั้นจะทำการหาค่า M (Mid-point), E (Expansion-point) ซึ่งค่าของ M และ E สามารถหาได้ตามสมการที่ 2.14

$$M = \frac{B+G}{2} \quad , E = 3M - 2W \tag{2.14}$$

โดยค่า M และ E ที่ได้จะนำไปหาค่าสัมประสิทธิ์ R, C และ S ตามสมการที่ 2.15

$$R = 2M - W \quad , C = \frac{W+M}{2} \quad , S = \frac{B+W}{2} \tag{2.15}$$

โดยค่าทั้งหมดนี้จะเป็นการกำหนดจุดเริ่มของกระบวนการของ Nelder-Mead algorithm และทำกระบวนการนี้ไปเรื่อย ๆ จนทำให้ได้ค่าที่ต่ำที่สุดหรือมากที่สุด

2.17 การประมาณค่าในช่วง (Interpolation)

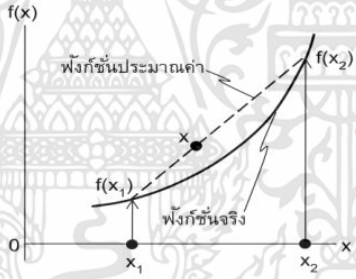
การประมาณค่าในช่วง (Interpolation) คือ กระบวนการในการหาฟังก์ชันที่ผ่านจุดที่กำหนดให้โดยเทคนิค Interpolation ที่ได้ใช้ในปริญญานิพนธ์ ได้แก่

2.17.1 Newton's Interpolating Polynomials

Newton's Interpolating Polynomials คือ การประมาณค่าในช่วงโดยใช้ผลต่างจากการแบ่งย่อยของนิวตัน ซึ่งเป็นที่นิยมใช้กันทั่วไปโดยจะศึกษา การประมาณค่าในช่วงแบบเชิงเส้นและแบบกำลังสอง

2.17.1.1 การประมาณค่าในช่วงเชิงเส้น (Linear Interpolation)

Linear Interpolation คือ การประมาณค่าในช่วงเชิงเส้น ซึ่งจะใช้ข้อมูลเพียง 2 ชุดเท่านั้น โดยจะอยู่ระหว่างค่าที่ต้องการประมาณค่ามาสร้างเป็นสมการเส้นตรง ประกอบด้วย ชุดที่ 1 คือ $f(x_1)$ คู่กับ x_1 และชุดที่ 2 คือ $f(x_2)$ คู่กับ x_2 ดังรูปที่ 2.53



รูปที่ 2.53 การประมาณค่าในช่วงเชิงเส้น [20]

โดย เส้นทึบ คือ ฟังก์ชันค่าจริง

เส้นประ คือ ฟังก์ชันค่าประมาณ

จากรูปที่ 2.53 ลากเส้นจากจุด 1 (x_1, $f(x_1)$) ไปยังจุด 2 (x_2, $f(x_2)$) คำตอบที่ต้องการประมาณค่าทั้งหมดจะหมดจะอยู่บนเส้นตรงประนี้ ซึ่งจะเห็นว่ามีความคลาดเคลื่อนจากคำตอบจริงจากเส้นโค้งทึบ โดยค่าฟังก์ชัน f(x) ได้ตามสมการที่ 2.16

$$f(x) = f(x_1) + \frac{f(x_1) - f(x_2)}{x_2 - x_1}(x - x_1) \tag{2.16}$$

2.18 สมการผลต่างสืบเนื่อง (Difference Equation)

สมการผลต่างสืบเนื่อง (Difference Equation) คือ สมการแสดงความสัมพันธ์ของผลลัพธ์กับเวลา โดยที่ค่าของผลลัพธ์จะมีการเปลี่ยนไปต่อเมื่อเวลามีการเปลี่ยนแปลง ซึ่งการเปลี่ยนแปลงของผลลัพธ์จะมีค่ามาจากผลต่างของผลลัพธ์ของเวลาล่วงหน้า กับผลลัพธ์ในเวลาปัจจุบัน ซึ่งสามารถเขียนให้อยู่ในรูปแบบของสมการได้ดังสมการที่ 2.17

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \frac{1}{a_0}\sum_{k=0}^{M} b_k x[n-k] \tag{2.17}$$

โดยที่ $y[n-k]$ คือ ค่าของ y ในคาบเวลาในอดีต

$x[n-k]$ คือ ค่าของ x ในคาบเวลาอดีตรวมกับคาบเวลาในปัจจุบัน

ซึ่งปริญญานิพนธ์นี้ได้อ้างอิงสมการที่ใช้ออกแบบวงจรกรองความถี่แบบดิจิทัลตามสมการไบควอตเดรติก (Biquadratic Equation) ซึ่งสมการดังกล่าวจะเป็นสมการอันดับที่สอง โดยจะมีฟังก์ชันถ่ายโอน (Transfer Function) ดังนี้

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{b_0 + b_1 z^{-1} + b_2 z^{-2}} = \frac{Y(z)}{X(z)} \tag{2.18}$$

เมื่อทำการจัดรูปสมการใหม่จะได้สมการดังนี้

$$X(z)a_0 + X(z)a_1 z^{-1} + X(z)a_2 z^{-2} = Y(z)b_0 + Y(z)b_1 z^{-1} + Y(z)b_2 z^{-2} \tag{2.19}$$

และทำการ Inversed Z-Transform กับสมการที่ 2.19 และจัดรูปสมการให้อยู่ในรูปอย่างง่ายได้ดังนี้

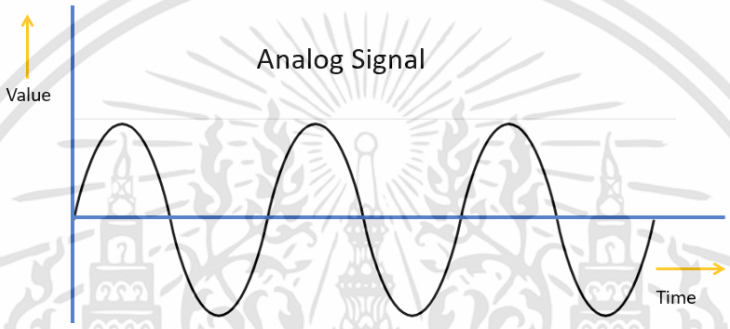
$$y(n) = a_0 x(n) + a_1 x(n-1) + a_2 x(n-2) - b_1 y(n-1) - b_2 y(n-2) \tag{2.20}$$

ซึ่งสมการที่ 2.20 จะเป็นสมการผลต่างสืบเนื่องที่จะถูกนำไปใช้ประยุกต์บนแอปพลิเคชันเครื่องช่วยฟังในส่วนของการชดเชยการสูญเสียการได้ยิน

2.19 การแปลงสัญญาณแอนะล็อกเป็นดิจิทัลและการแปลงสัญญาณดิจิทัลเป็นอนาล็อก

2.19.1 สัญญาณแอนะล็อก

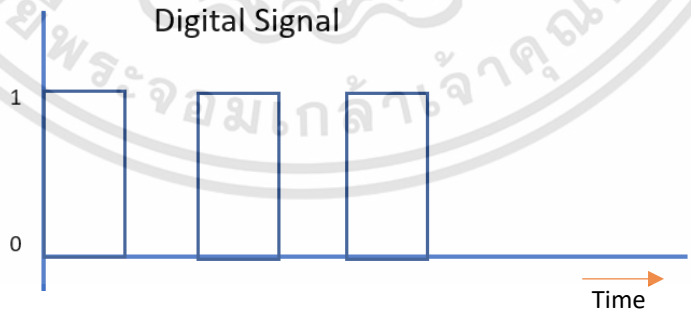
สัญญาณแอนะล็อก คือ สัญญาณข้อมูลแบบต่อเนื่อง มีขนาดสัญญาณไม่คงที่ การเปลี่ยนขนาดของสัญญาณจะแปลผันไปตามเวลา ทำให้จำนวนระดับหรือค่าของสัญญาณมาสามารถนับได้ สัญญาณอนาล็อกจะถูกรบกวนได้ง่ายเนื่องจากอาจถูกรบกวนให้มีความผิดพลาดได้ในขณะที่ทำการส่ง โดยสัญญาณอนาล็อกจะเป็นดังรูปที่



รูปที่ 2.54 สัญญาณแอนะล็อก [31]

2.19.2 สัญญาณดิจิทัล

สัญญาณดิจิทัล คือ สัญญาณข้อมูลแบบไม่ต่อเนื่อง มีจำนวนระดับหรือค่าของสัญญาณที่แน่นอน โดยจำนวนระดับของสัญญาณจะขึ้นอยู่กับจำนวนบิตของข้อมูล สัญญาณดิจิทัลจะเป็นสัญญาณที่ใช้ทำงานในคอมพิวเตอร์ โดยสัญญาณดิจิทัลจะเป็นดังรูปที่ 2.55

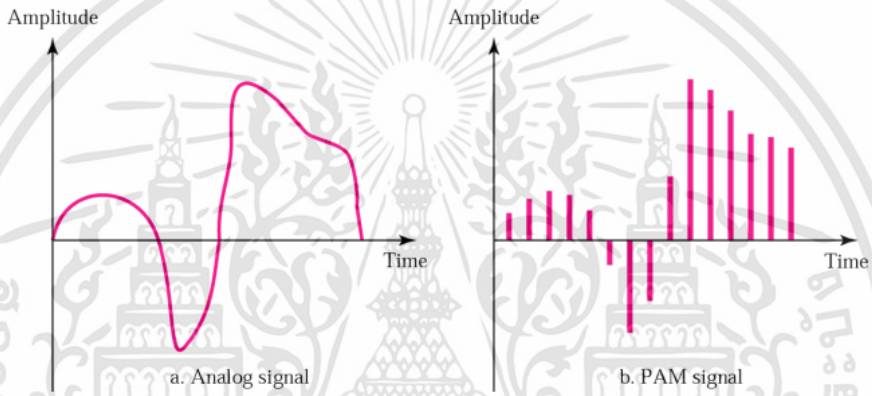


รูปที่ 2.55 สัญญาณดิจิทัล [31]

การแปลงสัญญาณอนาล็อกเป็นดิจิทัลคือการแปลงสัญญาณข้อมูลที่มนุษย์รับรู้ ให้เป็นสัญญาณทางไฟฟ้า เพื่อเข้าสู่การประมวลผล โดยการแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล ที่ศึกษามีดังนี้

2.19.2.1 การมอดูเลตแอมพลิจูดพัลส์ (PAM: Pulse Amplitude Modulation)

การมอดูเลตแอมพลิจูดพัลส์ เป็นการนำข้อมูลที่เป็นสัญญาณอนาล็อกมาทำการสุ่มตัวอย่างสัญญาณให้มีระยะช่วงเท่าๆ กัน ให้เป็นพัลส์ๆ โดยที่ขนาดของพัลส์แต่ละอันจะเท่ากับสัญญาณเดิมในช่วงเวลานั้น ๆ ในทางทฤษฎีการสุ่มตัวอย่างจะทำด้วยอัตราสองเท่าของแบนด์วิดธ์ของสัญญาณแอนะล็อก เป็นจำนวนครั้งต่อวินาที ไม่เหมาะสำหรับการสื่อสารเพราะจากระดับของสัญญาณยังมีจำนวนไม่จำกัด



รูปที่ 2.56 สัญญาณอนาล็อก (a) เมื่อผ่านกระบวนการมอดูเลตแอมพลิจูดพัลส์ จะได้สัญญาณแอมพลิจูดพัลส์ (b) [32]

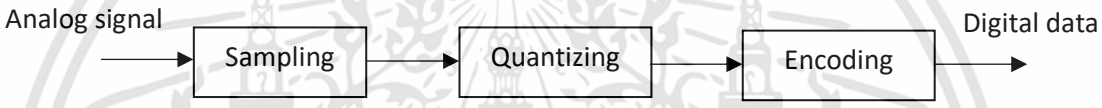
2.19.2.2 การมอดูเลตรหัสพัลส์ (PCM: Pulse Code Modulation)

การมอดูเลตรหัสพัลส์ สัญญาณอนาล็อกจะถูกสุ่มตัวอย่างสัญญาณ (Sampling) และถูกจัดระดับสัญญาณ (Quantizing) ซึ่งเป็นเลขฐานสองซึ่งมี 2 สถานะ คือ 1 กับ 0 การใช้การมอดูเลตรหัสพัลส์ ทำให้แปลงเป็นสัญญาณดิจิทัล เช่น รูปภาพ, เสียง และอื่น ๆ โดยการทำงานของการมอดูเลตรหัสพัลส์ ในกระเข้ารหัสสัญญาณจะประกอบด้วย 3 กระบวนการ คือ การสุ่มสัญญาณ (Sampling), จัดระดับสัญญาณ (Quantizing) และการเข้ารหัสสัญญาณ (Encoding)

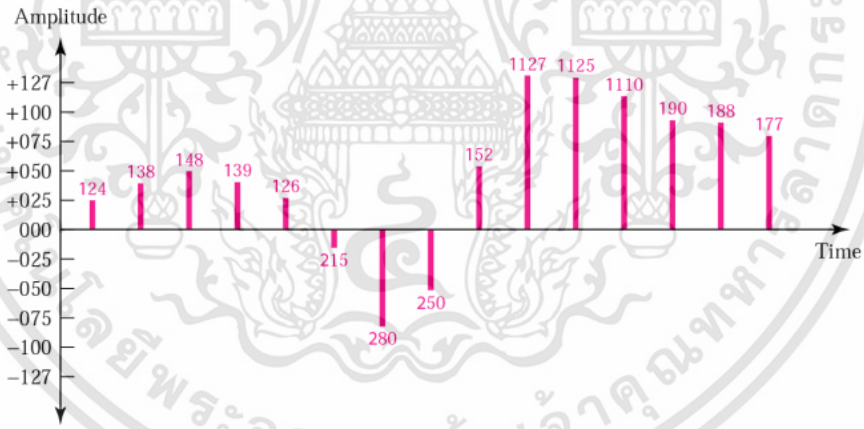
- การสุ่มสัญญาณ คือ การวัดขนาดของสัญญาณอนาล็อกเป็นจุด ๆ โดยที่ในแต่ละจุดจะมีระยะห่างเท่าๆ กัน ซึ่งแต่ละช่วงจะเรียกว่า ช่วงของการสุ่มสัญญาณ (Sampling interval) สัญญาณอนาล็อกจะกลายเป็นสัญญาณที่ไม่ต่อเนื่องทางเวลา (Discrete time signal)

- การจัดระดับสัญญาณ คือ การจัดระดับกลุ่มของตัวอย่างสัญญาณ (Sample) ที่ได้จากการสุ่มสัญญาณ ซึ่งอาจจะเท่ากันหรือไม่เท่ากันก็ได้ โดยค่าของกระบวนการการแอมพลิจูดพัลส์จะถูกจัดให้อยู่ในช่วงที่ใกล้เคียงกับค่าที่สุ่มได้มากที่สุด

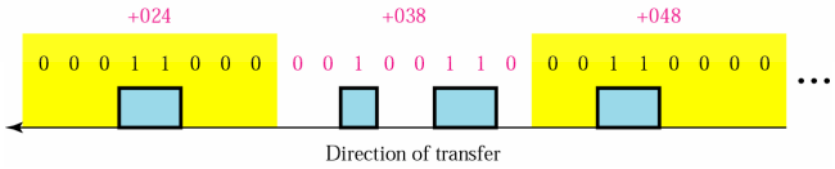
- การเข้ารหัสสัญญาณ คือ การแทนระดับของสัญญาณ ในการจัดระดับของสัญญาณด้วยกลุ่มของบิต โดยที่จำนวนบิตที่ใช้จะขึ้นอยู่กับความละเอียดของสัญญาณที่ต้องการ และนำมาแปลงเป็นเลขฐานสอง ซึ่งค่าของระดับสัญญาณจะเป็นบวกหรือลบก็ได้ โดยที่อาจจะใช้วิธี 2's Complement เพื่อใช้แทนจำนวนลบได้ เช่น ตัวอย่างต้องการความละเอียดที่ 16 บิต การจัดระดับสัญญาณจะเป็น 2^{16} = 65,536 ระดับ ซึ่งหากใช้ 2's Complement การแบ่งระดับจะเป็น -32,768 ถึง 32,767



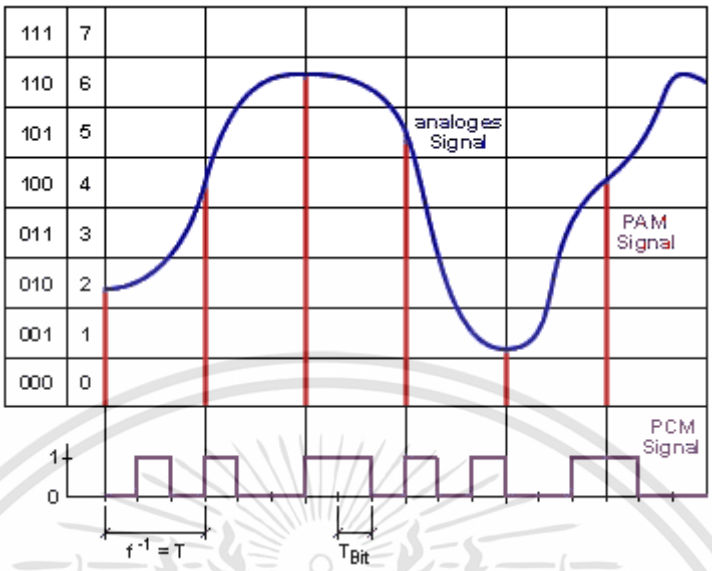
รูปที่ 2.57 บล็อกไดอะแกรมการเข้ารหัสสัญญาณ



รูปที่ 2.58 สัญญาณแอมพลิจูดพัลส์หลังผ่านการจัดระดับสัญญาณ [32]



รูปที่ 2.59 สัญญาณรหัสพัลส์บางส่วนที่ได้เป็นเลขฐานสองจากสัญญาณแอมพลิจูดพัลส์ [32]



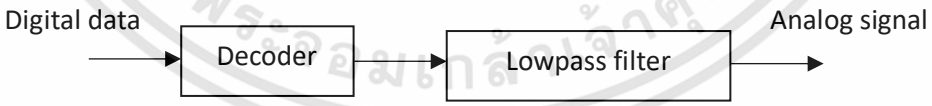
รูปที่ 2.60 กระบวนการการเข้ารหัสพัลส์ทั้งหมด [33]

2.19.3 การแปลงสัญญาณดิจิทัลเป็นแอนะล็อก

การแปลงสัญญาณดิจิทัลเป็นอนาล็อกคือการแปลงสัญญาณข้อมูลทางไฟฟ้า ให้เป็นสัญญาณที่ มนุษย์รับรู้ โดยการแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก จะมีกระบวนการ 2 กระบวนการ คือ การถอดรหัสสัญญาณ (Decoder) และ วงจรกรองความถี่ต่ำผ่าน (Lowpass filter)

การถอดรหัสสัญญาณ คือ การทำกระบวนการย้อนกลับในการแปลงข้อมูลที่ได้มาให้เป็นสัญญาณข้อมูลเดิม

วงจรกรองความถี่ต่ำผ่าน คือ วงจรที่ยอมให้สัญญาณความถี่ตั้งแต่ 0 เฮิรตซ์ จนถึงความถี่ที่กำหนดผ่านไปได้ โดยที่ความถี่ที่สูงกว่านั้นจะถูกลดทอนไปตามค่าลำดับ



รูปที่ 2.61 บล็อกไดอะแกรมการถอดรหัสสัญญาณ

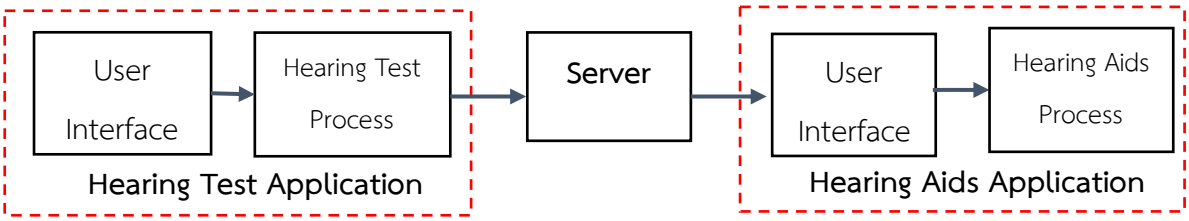
บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

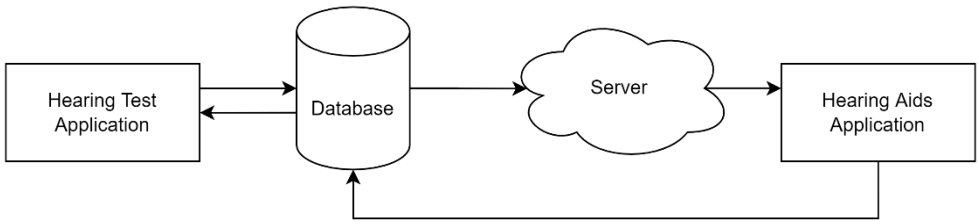
3.1 การออกแบบ

3.1.1 การออกแบบการทำงานของระบบ

ปริญญานิพนธ์นี้ได้ทำการออกแบบระบบการทำงานของโมบายแอปพลิเคชันสำหรับการทดสอบสมรรถภาพการได้ยินและเครื่องช่วยฟัง เพื่อให้ทุกคนโดยเฉพาะผู้สูงอายุสามารถเข้าถึงการใช้งานได้ง่ายและสะดวกมากยิ่งขึ้น โดยมี Firebase เป็นฐานข้อมูลที่ใช้สำหรับเก็บชื่อผู้ใช้งาน อายุ เพศ และประวัติการทดสอบการได้ยิน โดยในส่วนของประวัติการทดสอบการได้ยินจะถูกนำไปประมวลผลสำหรับออกแบบวงจรกรองความถี่แบบดิจิทัลที่เซิร์ฟเวอร์ (Server) โดยใช้การประมวลของเซิร์ฟเวอร์ในรูปแบบของคลาวด์คอมพิวติ้ง (Cloud Computing) ผ่านบริการของ DigitalOcean โดยบล็อกไดอะแกรมของระบบดังรูปที่ 3.1 เป็นการแสดงการทำงานของการออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน



รูปที่ 3.1 บล็อกไดอะแกรมแสดงการทำงานของการออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน

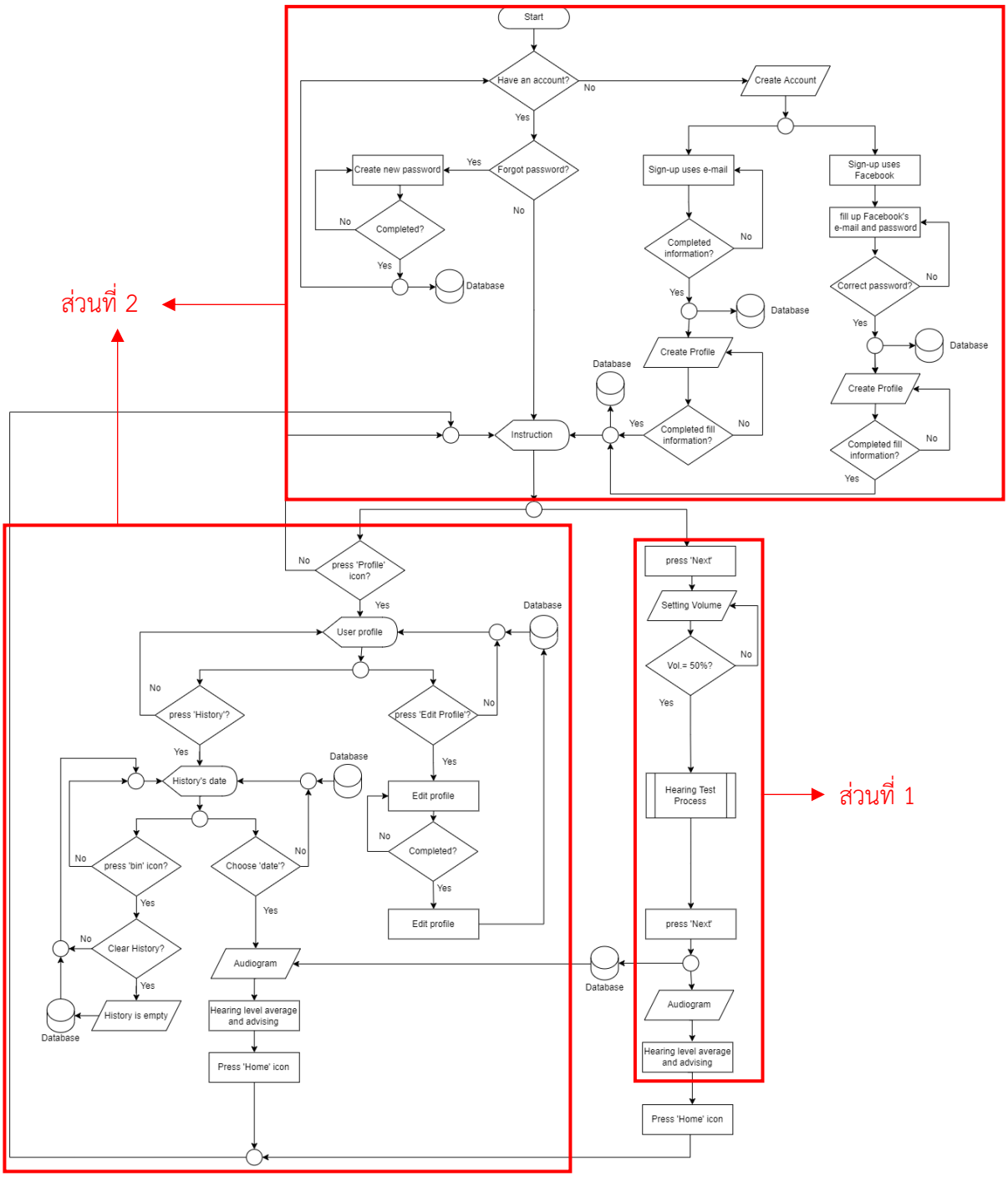


รูปที่ 3.2 บล็อกไดอะแกรมแสดงความสัมพันธ์ระหว่างแอปพลิเคชันสำหรับทดสอบการได้ยิน และเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน

จากรูปที่ 3.2 แสดงถึงความสัมพันธ์ระหว่างแอปพลิเคชันสำหรับทดสอบการได้ยินและเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน จะเห็นได้ว่าเมื่อแอปพลิเคชันสำหรับทดสอบการได้ยินได้ผลจากการทดสอบการได้ยินแล้วจะส่งค่าระดับการได้ยินไปยังฐานข้อมูล เพื่อเก็บประวัติการทดสอบและส่งไปยังส่วนของเซิร์ฟเวอร์เพื่อคำนวณค่าสัมประสิทธิ์สำหรับการออกแบบวงจรกรองความถี่แบบดิจิทัล ซึ่งจะนำไปใช้ต่อในแอปพลิเคชันเครื่องช่วยฟัง ในขณะเดียวกันสำหรับผู้ใช้งานที่ไม่เคยใช้งานแอปพลิเคชันการทดสอบสมรรถภาพการได้ยิน และต้องการนำผลการทดสอบจากภายนอกมาใช้งานสามารถกรอกค่าจากทางแอปพลิเคชันเครื่องช่วยฟังเพื่อนำข้อมูลไปเก็บไว้ที่ฐานข้อมูลแล้วประมวลผลเช่นเดียวกันสำหรับขั้นตอนถัดไป

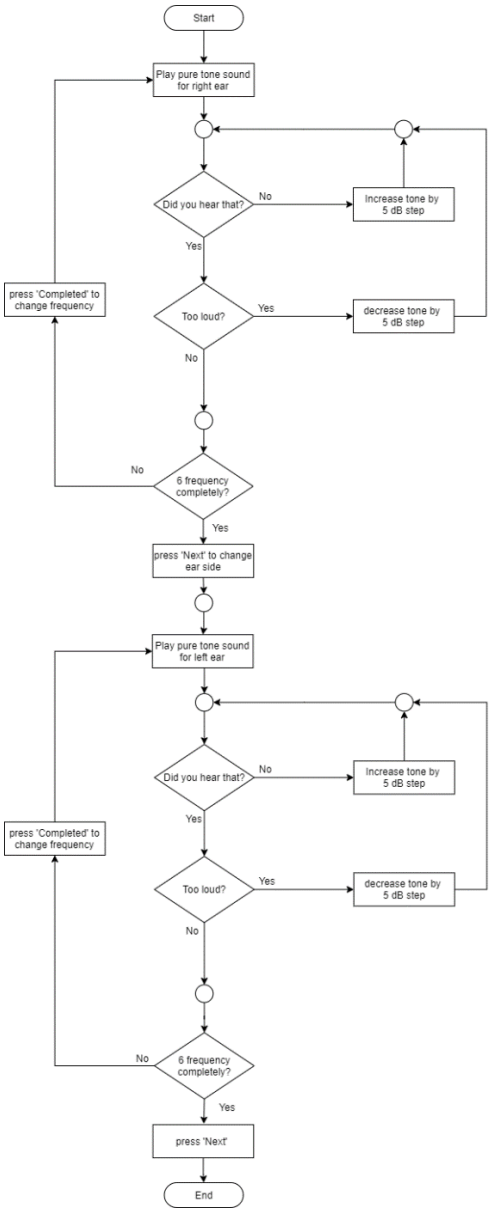
3.1.2 การออกแบบระบบการทำงานของแอปพลิเคชันสำหรับการทดสอบการได้ยิน

จากการศึกษาวิธีการตรวจสมรรถภาพการได้ยินและมาตรฐานที่ใช้สำหรับการทดสอบสมรรถภาพการได้ยิน ปริญญานิพนธ์นี้ได้ทำการออกแบบระบบและวิธีการทดสอบสมรรถภาพการได้ยินที่ใช้บนโมบายแอปพลิเคชันโดยใช้ภาษา Flutter ในการเขียนแอปพลิเคชันและใช้ Firebase เป็นฐานข้อมูลสำหรับเก็บประวัติของผู้ใช้งาน โดยระบบการทำงานโดยรวมของแอปพลิเคชันการทดสอบสมรรถภาพการได้ยินเป็นไปตามรูปที่ 3.3



รูปที่ 3.3 แผนภาพระบบการทำงานแอปพลิเคชันการทดสอบสมรรถภาพการได้ยิน

จากรูปที่ 3.3 แสดงแผนภาพระบบการทำงานของแอปพลิเคชันการทดสอบสรรถภาพการได้ยิน (Hearing Test Application) สามารถแบ่งระบบการทำงานออกเป็น 2 ส่วน ได้แก่ ส่วนที่ 1 ส่วนที่ใช้สำหรับการทดสอบสมรรถภาพการได้ยินด้วยวิธีการนำเสียงผ่านอากาศโดยการใช้สัญญาณความถี่เดียว (Pure Tone Air Conduction) โดยความถี่ออคเทฟ (Octave Frequency) ทั้ง 6 ความถี่ได้แก่ ความถี่ที่ 250, 500, 1000, 2000, 4000 และ 8000 เฮิร์ตซ์ ตามรูปที่ 3.4 และส่วนที่ 2 ที่เป็นส่วนของบัญชีผู้ใช้งานบนแอปพลิเคชัน



รูปที่ 3.4 แผนภาพระบบการทำงานการทดสอบสมรรถภาพการได้ยิน

3.1.3 การออกแบบเสียงที่ใช้ในการทดสอบ

ในการทดสอบสมรรถภาพการได้ยินจะให้ผู้ทดสอบฟังสัญญาณความถี่เดียวที่ความถี่ต่าง ๆ เพื่อให้สามารถเก็บผลการทดสอบทั้ง 6 ความถี่ และนำไปใช้ในการสร้างแอปพลิเคชันเครื่องช่วยฟังในขั้นตอนถัดไป แอปพลิเคชันจึงสร้างสัญญาณความถี่เดียวเพื่อใช้ในการทดสอบ โดยสัญญาณความถี่เดียวที่ใช้คือ สัญญาณที่เกิดจากสัญญาณ sine wave ตามสมการที่ 3.1 โดยจะถูกสร้างขึ้นด้วยการเขียนโปรแกรมภาษา python ซึ่งการทดสอบสมรรถภาพการได้ยินในปริญญานิพนธ์จะใช้การอ้างอิงจากกราฟ Equal Loudness Contour [10] โดยจะใช้เส้นที่ 40 phon หรือ เส้นที่มีระดับความดัง 40 dB ที่ความถี่ 1000 Hz เนื่องจากเป็นเส้นที่อยู่ในระดับเดียวกันกับกราฟ A-weighting [9] ซึ่งเป็นสัญญาณความถี่เดียวที่มนุษย์เริ่มได้ยิน โดยในความถี่ 250, 500, 1000, 2000, 4000 และ 8000 Hz จะมีระดับความดังเริ่มต้นที่อ้างอิงจากกราฟ A-weightingคือ 40, 34, 40, 37, 32 และ 43 dB HL ตามลำดับ จากนั้นจะนำค่าระดับความดังไปคำนวณแอมพลิจูด (ในหน่วยโวลต์) ดังสมการที่ 3.2

$$y(t) = A\sin(\omega t + \theta) \tag{3.1}$$

ในการกำหนดระดับความดังของเสียง Sinewave จะถูกควบคุมด้วยแอมพลิจูดหรือ A ในสมการซึ่งจะมีหน่วยเป็น Volt แต่ในการทดสอบจะใช้หน่วย dB SPL เป็นหน่วยของระดับความดังของเสียง ดังสมการที่ 3.2 โดยในที่นี้ค่า x ของเครื่องโทรศัพท์ Samsung Galaxy J7 มีค่าเท่ากับ 93.9

$$\text{dB SPL} = 20\log\left(\frac{\text{Volt}}{0.05}\right) + x \tag{3.2}$$

โดย x คือ ค่าการตอบสนองของไมโครโฟน (Microphone Sensitivity)

0.05 คือ ค่าความศักย์ขาเข้า (Vin)

เพื่อให้ได้ระดับความดังในหน่วยโวลต์ (Volt) เพื่อนำไปสร้างสัญญาณ sine wave ในความถี่ต่าง ๆ ซึ่งจะสามารถเขียนโปรแกรมด้วยภาษา python ได้ดังรูปที่ 3.5

```
T0 = 80
T = 4*T0
N = np.arange(0,T,1)
f = [250, 500, 1000, 2000, 4000, 8000]

SPL = [40,34,40,37,32,43]
dBv = []
Amp = []
for i in range(len(SPL)):
    v = 20*(math.log(0.05*(10**(((SPL[i]+100)-93.9)/20)),10))
    dBv.append(v)
    amp = 10**(v/20)
    Amp.append(amp)
```

รูปที่ 3.5 โปรแกรมที่ใช้คำนวณระดับความดัง

ระดับความดังของเสียงจะคำนวณออกมาให้มีค่า 100 dB HL เนื่องจากการนำเสียงไปใช้บนแอปพลิเคชันจะไม่สามารถเพิ่มความดังของเสียงได้ สามารถลดความดังของเสียงในรูปแบบของเปอเซ็นไทล์ จึงจำเป็นต้องคำนวณออกมาให้มีระดับความดังที่ 100 dB HLเมื่อได้ระดับความดังในแต่ละความถี่ ต่อไปจะเป็นส่วนของการสร้างสัญญาณ sine wave เพื่อนำมาใช้ทดสอบโดยใช้โปรแกรม python ได้ดังรูปที่ 3.6

```
a = []
c = []
Fs = []
j = 0
for i in range(6):
    b = []
    Fs.append(f[i]*40)
    x = 2*np.pi*f[i]*N/Fs[i]
    a = amp*np.sin(x)
    time = 300*(f[i]//250)
    for j in range(time):
        b.extend(a)
    c.append(b)
print('Audio play is done')
```

รูปที่ 3.6 โปรแกรมที่ใช้ในการสร้างสัญญาณ sine wave

เมื่อได้สัญญาณ sine wave ในแต่ละความถี่ จะสามารถสร้างเสียงและบันทึกเสียงเพื่อนำไปใช้ในการทดสอบสมรรถภาพการได้ยิน โดยการทดสอบจะต้องเป็นเสียงที่มีการแยกเสียงในแต่ละข้างของหู จึงจะต้องทำการแยกเสียงโดยใช้การ transpose ข้อมูล และใส่ 0 ที่มีความยาวเท่ากับข้อมูลที่ทำการ transpose ซึ่งจะสามารถแยกเสียงออกในแต่ละข้างได้ โดยความแตกต่างของลำดับข้อมูลหากข้อมูลเป็นการที่สร้างเสียงให้ออกทางด้านขวาจะต้องให้ข้อมูลที่เป็น 0 อยู่แถวที่ 1 ของข้อมูล แต่ถ้าหากต้องการให้เสียงออกทางด้านซ้ายจะต้องให้ข้อมูลที่เป็น 0 อยู่แถวที่ 2 ของข้อมูล ซึ่งจะสามารถเขียนโปรแกรมได้ดังรูปที่ 3.7 และ 3.8

```
for i in range(len(c)):
    x = np.float32(c[i])
    ct = np.transpose(c[i])
    ct0 = np.zeros(len(c[i]),dtype='d')
    ct1 = np.transpose(ct0)
    m = np.reshape(ct,1*len(ct))
    yleft = np.column_stack((m,ct1))
    x = np.float32(yleft)
    write("left{0}.mp3".format(i+1), Fs[i], x)
print('DONE')
```

รูปที่ 3.7 โปรแกรมสำหรับการบันทึกเสียงของเสียงที่ออกด้านซ้าย

```
for i in range(len(c)):
    x = np.float32(c[i])
    ct = np.transpose(c[i])
    ct0 = np.zeros(len(c[i]),dtype='d')
    ct1 = np.transpose(ct0)
    m = np.reshape(ct,1*len(ct))
    yleft = np.column_stack((ct1,m))
    x = np.float32(yleft)
    write("right{0}.mp3".format(i+1), Fs[i], x)
print('DONE')
```

รูปที่ 3.8 โปรแกรมสำหรับการบันทึกเสียงของเสียงที่ออกด้านขวา

เมื่อได้เสียงที่ใช้ในการทดสอบสมรรถภาพการได้ยินในความถี่ต่าง ๆ จะสามารถนำเสียงไปใช้ในแอปพลิเคชันได้ในขั้นตอนต่อไป

3.1.4 การออกแบบแอปพลิเคชันผ่านภาษาโปรแกรม Flutter ของแอปพลิเคชันสำหรับการทดสอบการได้ยิน

3.1.4.1 การออกแบบ User Interface และ User Experience

- User Interface คือ เป็นส่วนที่สื่อสารระหว่างผู้ใช้งานกับระบบซึ่งก็คือการออกแบบการแสดงผลหน้าเว็บไซต์หรือหน้าแอปพลิเคชันให้มีความสวยงามหน้าใช้งาน

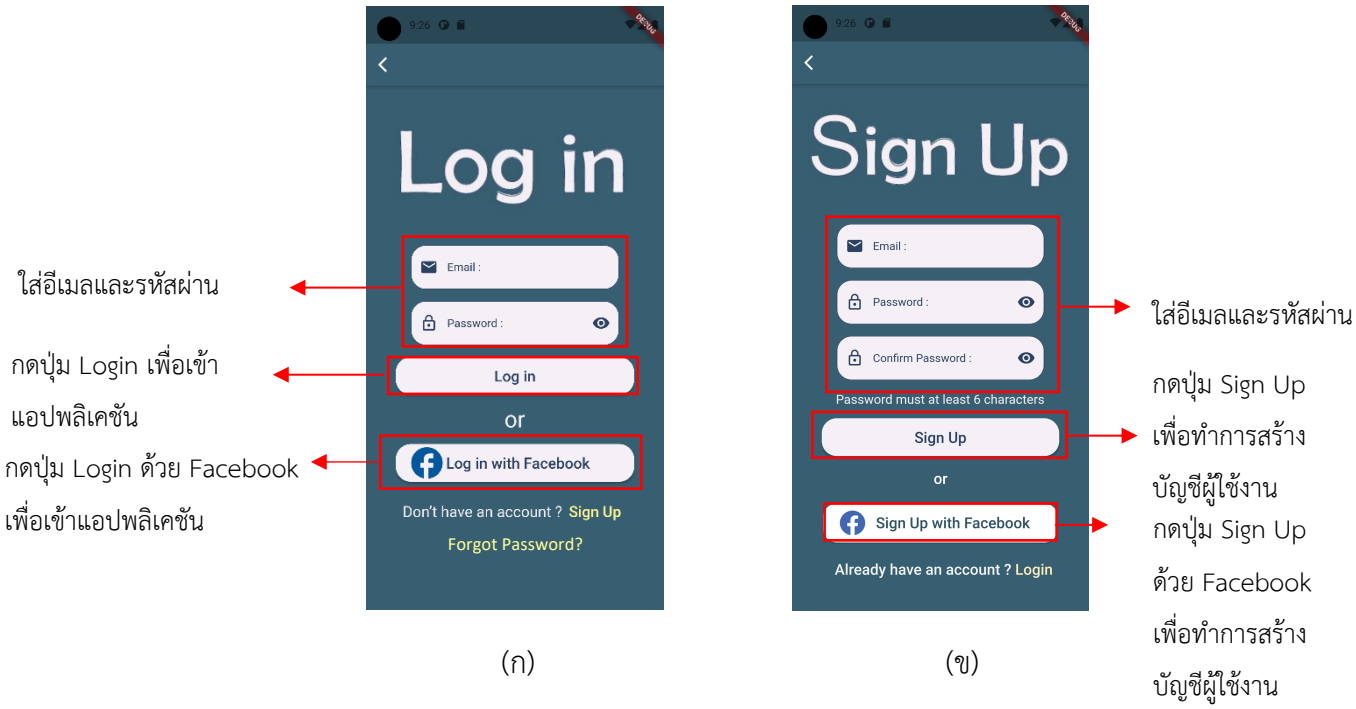
- User Experience คือ ประสบการณ์ของผู้ใช้งานซึ่งก็คือเป็นส่วนที่ทำให้ผู้ใช้งานนั้นมีความสะดวกในการใช้งาน เข้าใจได้ง่าย และมีลำดับขั้นตอนที่ชัดเจน

รูปที่ 3.9 User Interface และ User Experience ที่ออกแบบเพื่อใช้ในแอปพลิเคชัน

3.1.4.2 ส่วนบัญชีผู้ใช้งานผ่านแอปพลิเคชัน

รูปที่ 3.10 หน้าเข้าสู่แอปพลิเคชัน

จากรูปที่ 3.10 เมื่อทำการเปิดแอปพลิเคชัน Hearing Test ผู้ใช้งานที่มีบัญชีผู้ใช้แล้วให้ทำการกดปุ่ม Login ส่วนผู้ใช้งานที่ยังไม่เคยสร้างบัญชีผู้ใช้ให้กดปุ่ม Sign Up โดยหน้าจอที่จะแสดงผลจะเป็นไปตามรูปที่ 3.11



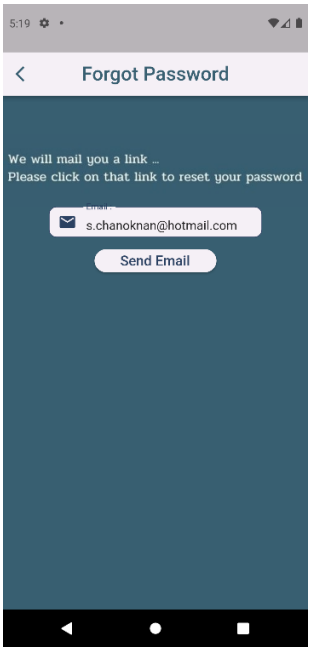
รูปที่ 3.11 หน้าจอแอปพลิเคชันเมื่อทำการกดปุ่มเข้าสู่ระบบ

(ก) หน้าสำหรับทำการเข้าสู่ระบบบัญชีผู้ใช้งาน (ข) หน้าสำหรับสร้างบัญชีผู้ใช้งาน

จากรูป 3.11 (ข) แสดงหน้าจอ Sign Up ซึ่งสามารถสร้างบัญชีผู้ใช้ด้วยอีเมลหรือ Facebook โดย ถ้าสร้างบัญชีผู้ใช้ด้วยอีเมลต้องกำหนดรหัสผ่านที่มีอักขระเกิน 6 ตัว และระบุแอดเดรสของอีเมลให้ครบถ้วน เมื่อกดปุ่ม Sign Up หรือกดปุ่ม Sign Up with Facebook จะเป็นขั้นตอนสร้างบัญชีผู้ใช้สำเร็จเรียบร้อยแล้ว ขั้นตอนต่อไปหลังจากที่ผู้ใช้งานกดปุ่ม Sign Up หรือ Sign Up with Facebook จะเข้าสู่หน้ากรอกข้อมูล รูปภาพ ชื่อ วันเกิด และเพศให้ครบตามที่ระบุไว้ และทำการกด Save เพื่อทำการส่งข้อมูลไปยังฐานเก็บข้อมูล ดังรูปที่ 3.12

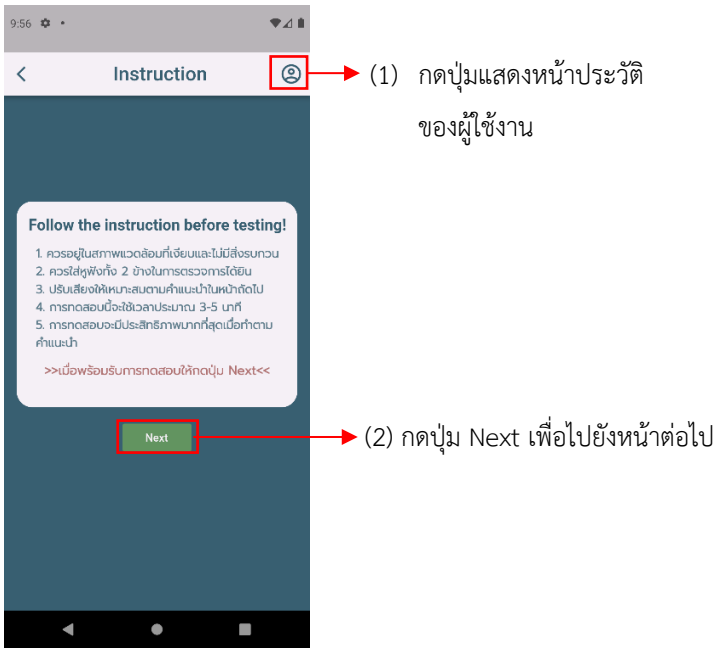
รูปที่ 3.12 หน้ากรอกข้อมูลผู้ใช้งาน

จากรูปที่ 3.11 (ก) แสดงหน้าจอ Login ในกรณีที่ผู้ใช้งานมีบัญชีผู้ใช้งานที่ลงทะเบียนแล้วสามารถเข้าสู่ระบบโดยเลือกการเข้าสู่ระบบด้วยอีเมลหรือ Facebook โดยกรณีที่ผู้ใช้งานลืมรหัสผ่านสามารถ Forgot Password? เพื่อทำการเปลี่ยนรหัสผ่านใหม่ ดังรูปที่ 3.13



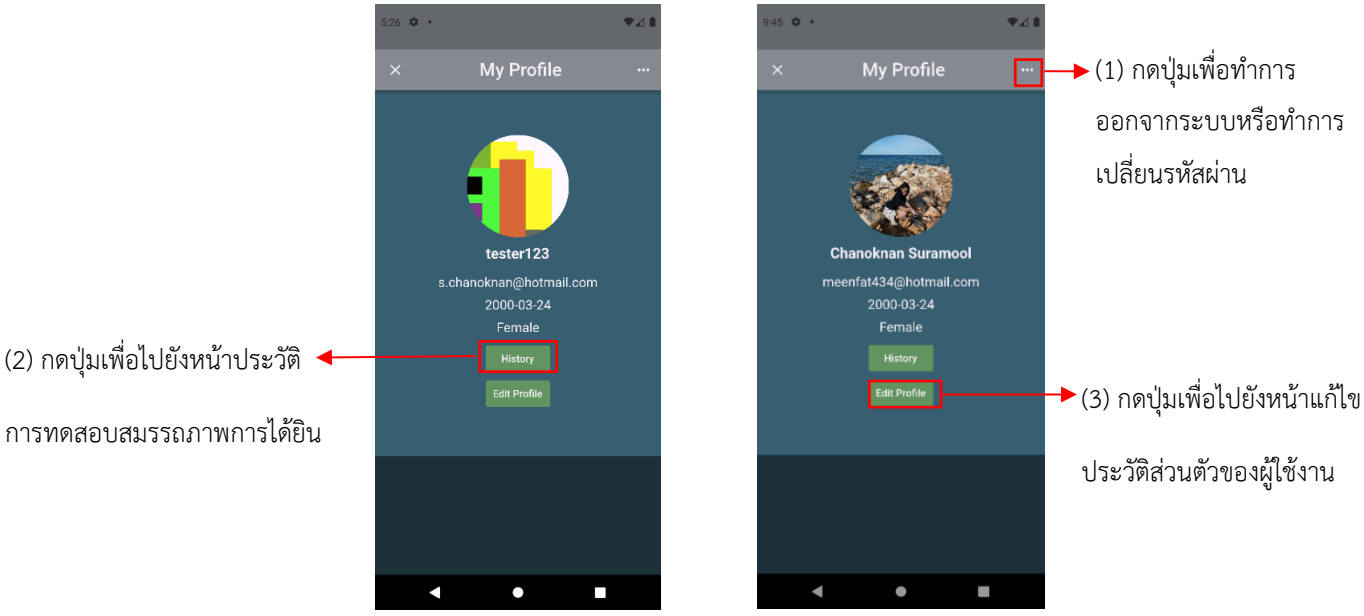
รูปที่ 3.13 หน้าตัวอย่างกรณีผู้ใช้งานลืมรหัสผ่าน

จากรูปที่ 3.13 กรณีผู้ใช้งานลืมรหัสผ่านให้ทำการกรอกอีเมลที่ลงทะเบียนไว้และทำการกดที่ปุ่มSend Email ระบบจะทำการส่งอีเมลไปยังอีเมลที่ผู้ใช้งานได้กรอกไว้เพื่อทำการเปลี่ยนรหัสผ่านใหม่



รูปที่ 3.14 หน้าคำแนะนำก่อนเข้าทำการทดสอบสมรรถภาพการได้ยิน

จากรูปที่ 3.11 (ก) เมื่อทำการเข้าสู่ระบบบัญชีผู้ใช้งานได้เรียบร้อย ระบบจะแสดงผลหน้าจอดังรูปที่ 3.14 ซึ่งเป็นการแสดงผลหน้าจอเกี่ยวกับคำแนะนำก่อนทำการทดสอบสมรรถภาพการได้ยิน เมื่อทำการกดสัญลักษณ์ตาม (1) จะแสดงผลหน้าประวัติผู้ใช้งานดังรูปที่ 3.15 เมื่อทำการกดปุ่ม Next (2) ระบบจะแสดงผลยังหน้าถัดไป ดังรูปที่ 3.16

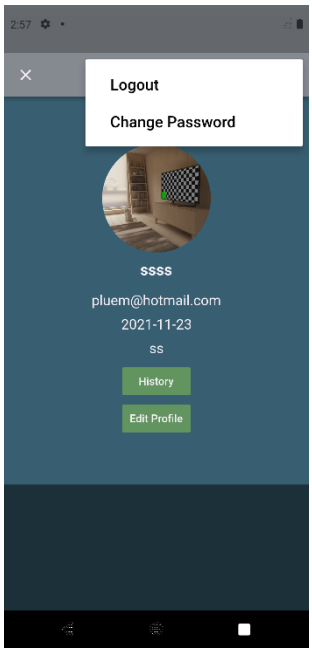


(ก) (ข)

รูปที่ 3.15 หน้าประวัติผู้ใช้งาน

(ก) ประวัติผู้ใช้งานที่เข้าสู่ระบบด้วยอีเมล (ข) ประวัติผู้ใช้งานที่เข้าสู่ระบบด้วย Facebook

จากรูปที่ 3.15 เมื่อทำการกดสัญลักษณ์ตาม (1) ระบบมีให้เลือกระหว่างการออกจากระบบบัญชีผู้ใช้งานและการเปลี่ยนรหัสผ่านของบัญชีผู้ใช้ ดังรูปที่ 3.16

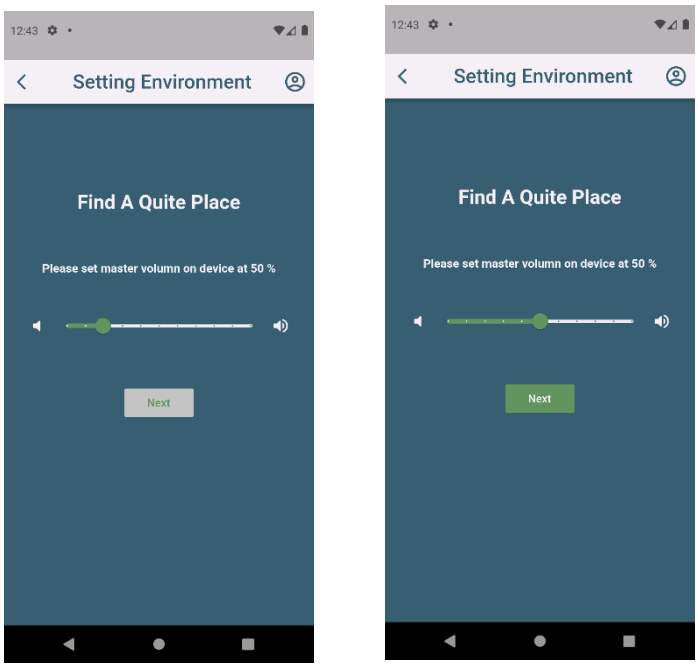


รูปที่ 3.16 กล่องข้อความที่แสดงเมื่อทำการกด (1) ดังรูปที่ 3.15

3.1.4.3 ส่วนที่ใช้สำหรับการทดสอบสมรรถภาพการได้ยิน

เมื่อทำการสร้างบัญชีผู้ใช้งานเสร็จเป็นที่เรียบร้อย ในส่วนถัดมาจะเป็นส่วนของระบบการทดสอบสมรรถภาพการได้ยิน โดยผู้ใช้งานควรหลีกเลี่ยงการอยู่ในสถานที่ที่มีเสียงดัง หรือเสียงรบกวนเพื่อให้ผลลัพธ์ที่ได้มีความผิดพลาดน้อยที่สุด ระบบจะเริ่มทดสอบที่หูขวาและหูซ้ายตามลำดับ เมื่อทำการทดสอบเสร็จเป็นที่เรียบร้อย ระบบจะแสดงผลของกราฟการได้ยินและข้อแนะนำระดับความสูญเสียการได้ยินโดยรวมซึ่งจะแบ่งเป็น 5 ระดับ ตั้งแต่ไม่มีการสูญเสียจนถึงการสูญเสียมากเกินไปจนต้องรักษา จึงจะถือเป็นการเสร็จสิ้นกระบวนการทดสอบ

จากรูปที่ 3.14 เมื่อทำการกดปุ่ม ‘Next’ จะเป็นในส่วนของหน้าสำหรับการปรับเสียงบนโทรศัพท์เพื่อเตรียมใช้สำหรับการทดสอบสรรถภาพทางการได้ยิน โดยที่เสียงที่ทำการปรับจะต้องอยู่ที่ 50% จึงจะสามารถเข้าสู่ถัดไปได้ดังรูปที่ 3.17



(ก) (ข)

รูปที่ 3.17 หน้าที่ใช้สำหรับการปรับเสียงโทรศัพท์ก่อนทำการทดสอบ

(ก) กรณีปรับเสียงไม่ถึง 50% (ข) กรณีปรับเสียงถึง 50%

```
I/flutter ( 6917): val:0.2
I/flutter ( 6917): val no multiply: 20.0
```

(ก)

```
I/flutter ( 6917): val:0.5
I/flutter ( 6917): val no multiply: 50.0
```

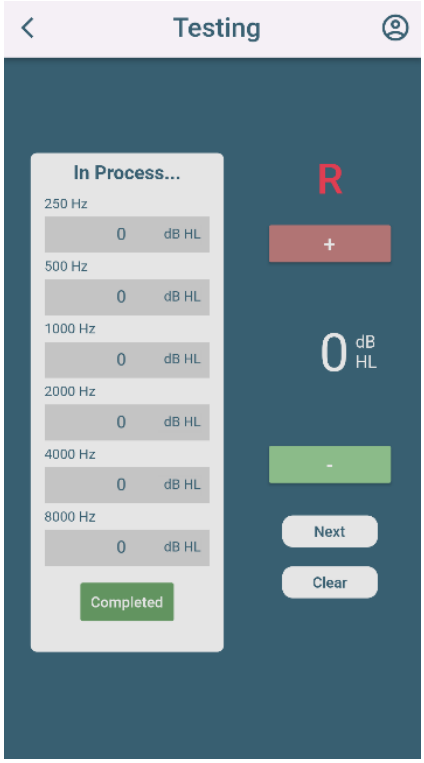
(ข)

รูปที่ 3.18 หน้า Debug Console ที่ระบุค่าเสียงบนโทรศัพท์ ณ ขณะนั้น

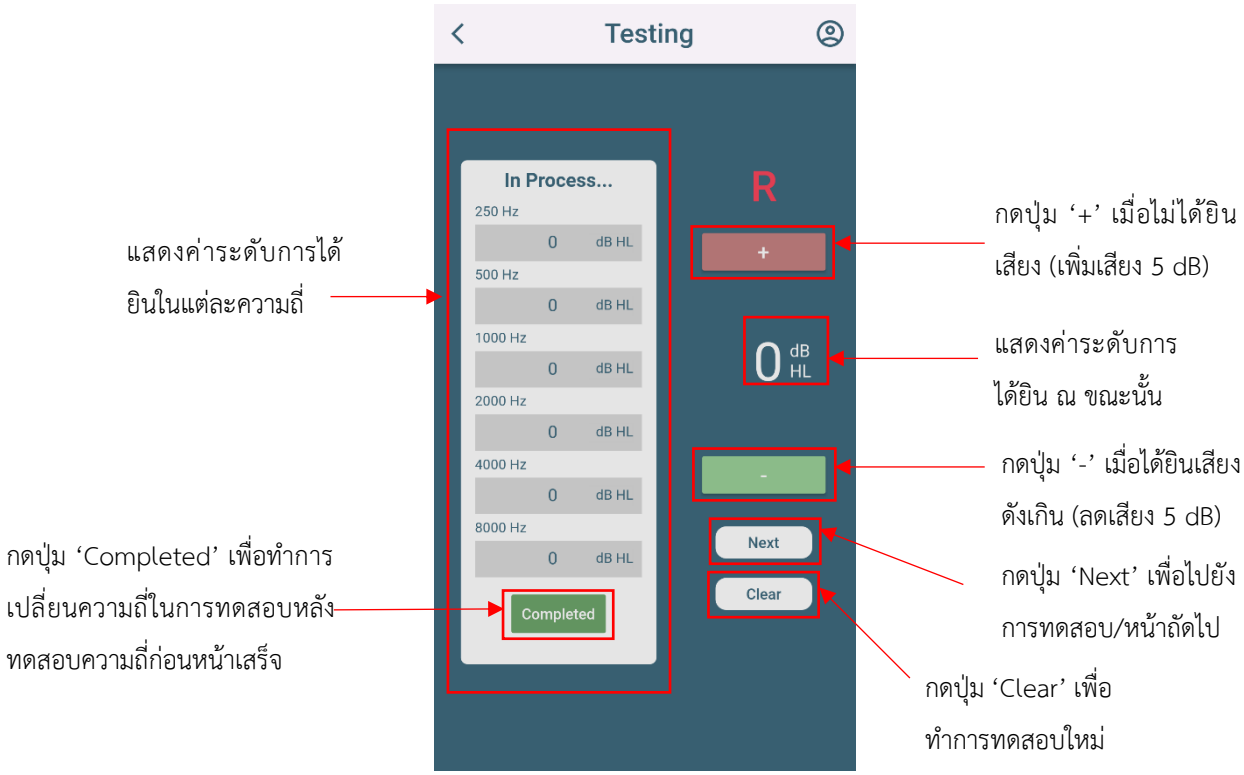
(ก) กรณีปรับเสียงไม่ถึง 50% (ข) กรณีปรับเสียงถึง 50%

หลังเสร็จสิ้นขั้นตอนการปรับเสียงโทรศัพท์จนได้ผลลัพธ์แสดงตามรูปที่ 3.18 (ข) ให้ทำการกดปุ่ม ‘Next’ เพื่อเข้าสู่การทดสอบสมรรถภาพการได้ยินโดยเริ่มที่หูขวา ตามรูปที่ 3.19 การแสดงผลผ่านหน้าจอทางด้านซ้าย จะเป็นการแสดงค่าระดับการได้ยินในแต่ละช่วงความถี่ เมื่อผู้ใช้ได้ยินเสียงให้กดปุ่ม ‘Completed’ เพื่อเปลี่ยนฟังเสียงในความถี่ถัดไป ทำเช่นเดียวกันจนครบทั้ง 6 ความถี่ และการแสดงผลผ่านหน้าจอทางด้านขวาจะเป็นการปรับระดับเสียงตามที่ผู้ใช้งานสามารถได้ยิน

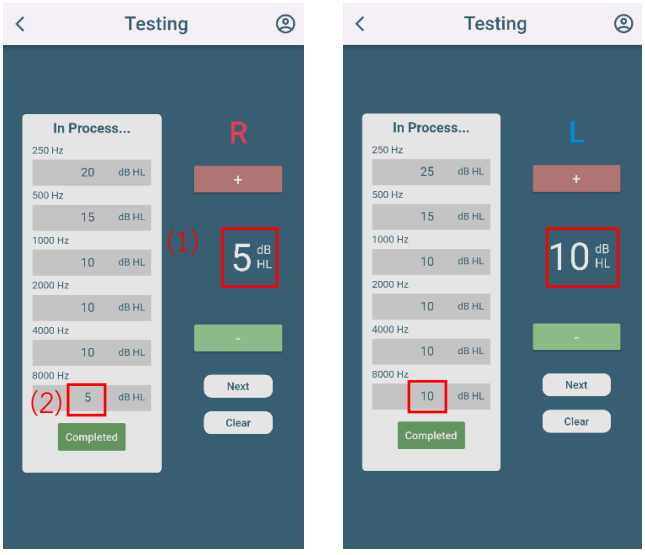
กด ‘+’ เพื่อเพิ่มเสียงขึ้น 5 dB เมื่อไม่ได้ยินเสียง และกด ‘-‘ เพื่อลดเสียงลง 5 dB เมื่อได้ยินเสียงดังจนเกินไป เมื่อทำการทดสอบครบทั้ง 6 ความถี่ให้กด ‘Next’ เพื่อทำการทดสอบสมรรถภาพการได้ยินของหูซ้าย ซึ่งใช้หลักการเดียวกันกับการทดสอบของหูขวา



รูปที่ 3.19 หน้าที่ใช้สำหรับทดสอบสรรถภาพการได้ยินของหูขวา



รูปที่ 3.20 หน้าที่ของปุ่มที่ใช้สำหรับทดสอบสรรถภาพการได้ยิน



(ก) (ข)

รูปที่ 3.21 ตัวอย่างผลการทดสอบสมรรถภาพการได้ยิน

(ก) หูขวา (ข) หูซ้าย

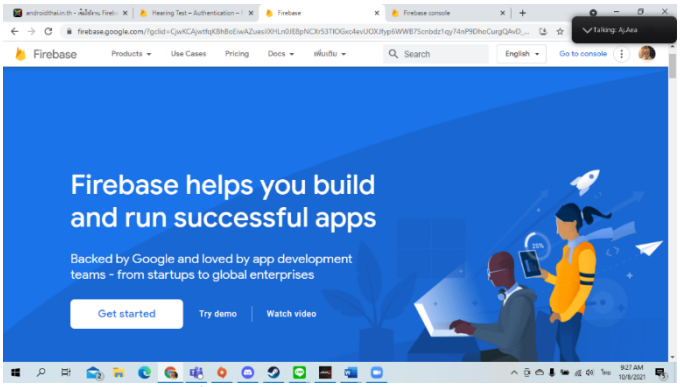
จาก (1) ดังรูปที่ 3.21 ที่เป็นส่วนแสดงค่าระดับการได้ยิน ณ ขณะนั้น จะเห็นได้ว่าเป็นการทดสอบสมรรถภาพการได้ยินของความถี่ 8000 เฮิร์ตซ์ซึ่งจะมีค่าตรงกับ (2) ที่เป็นการแสดงค่าระดับการได้ยินในแต่ละความถี่

3.1.5 การออกแบบฐานข้อมูล

การออกแบบฐานข้อมูลของแอปพลิเคชัน Hearing Test จะใช้ Firebase เป็นการจัดการในการเก็บฐานข้อมูลของผู้ใช้งาน

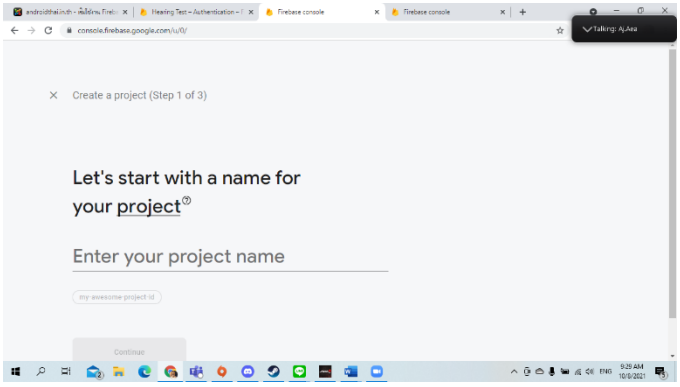
3.1.5.1 การสร้างฐานข้อมูลบน Firebase

1) เข้าไปที่หน้าเว็บไซต์ Firebase กด Get Start เพื่อเริ่มสร้างโครงงาน

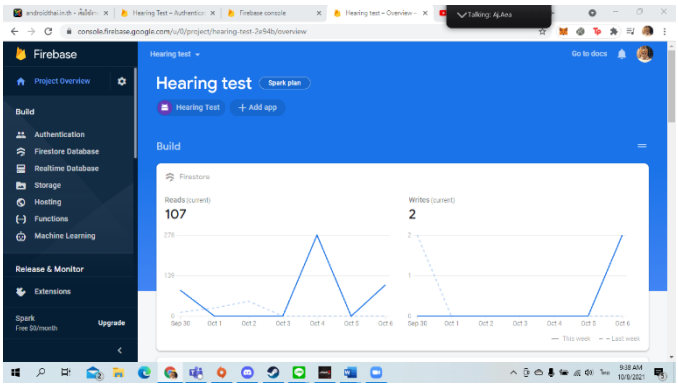


รูปที่ 3.22 หน้าเว็บไซต์ Firebase

2) ตั้งชื่อโครงงานจากนั้นกดสร้างโครงงาน เมื่อสร้างฐานข้อมูลเสร็จจะได้ดังรูปที่ 3.23

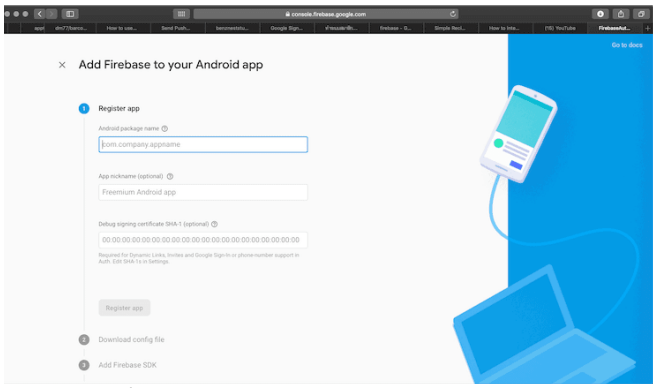


รูปที่ 3.23 หน้าเว็บไซต์สำหรับการสร้างโครงงาน



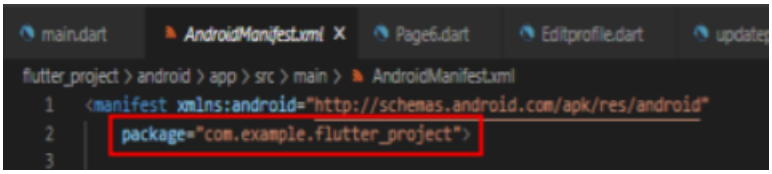
รูปที่ 3.24 การสร้างฐานข้อมูลเสร็จ

3) เลือกแพลตฟอร์มระบบปฏิบัติการแอนดรอยด์เพื่อทำการลงทะเบียนแอปพลิเคชัน Hearing Test



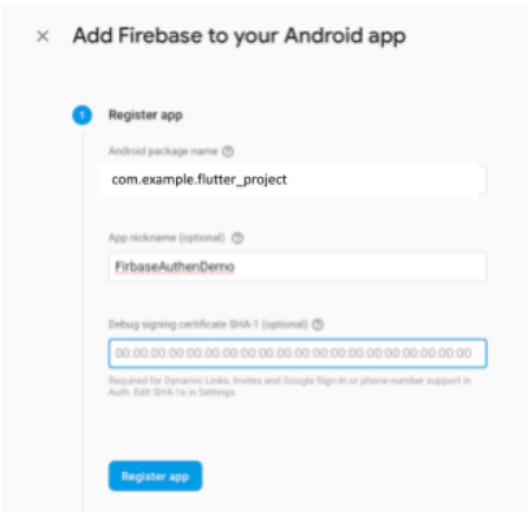
รูปที่ 3.25 การลงทะเบียนสร้างแอปพลิเคชัน

4) เข้าไปที่ Visual Studio ไปที่ Project App ที่ได้ทำการสร้างไว้และเข้าไปที่โฟลเดอร์ android โฟลเดอร์ app โฟลเดอร์ src โฟลเดอร์ main และ AndroidMainfest.xml แล้วตรวจสอบชื่อ Package



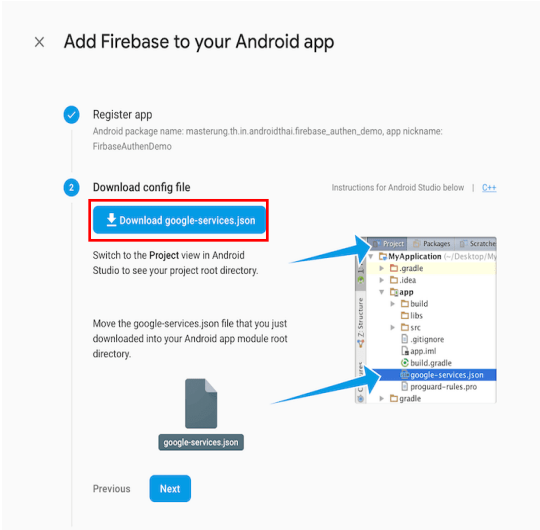
รูปที่ 3.26 ตัวอย่างการตรวจสอบชื่อ Package

5) คัดลอก Package ใส่ไว้ที่ Android package name

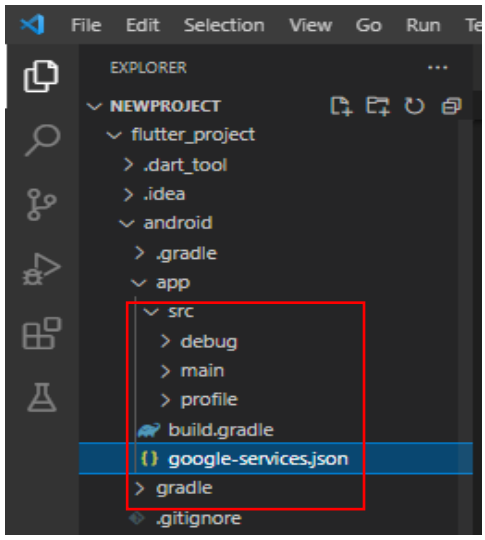


รูปที่ 3.27 ตัวอย่างการวาง Package ใน Android package name

6) ทำการดาวน์โหลดไฟล์ google-services.json แล้วนำไฟล์ไปวางในโครงงานที่โฟล์เดอร์ app ใน src

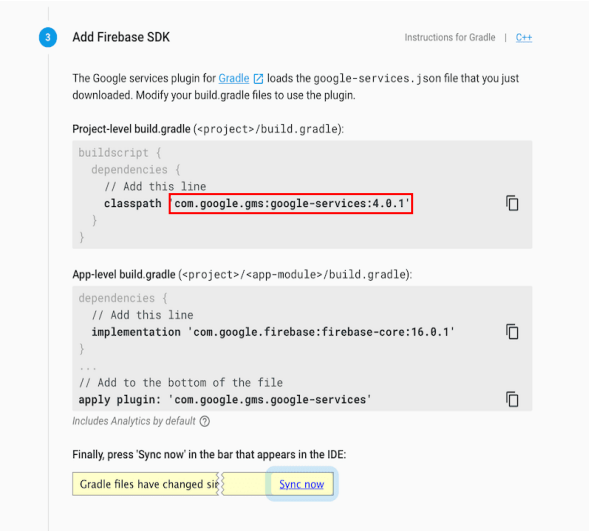


รูปที่ 3.28 ดาวน์โหลดไฟล์ google-services.json

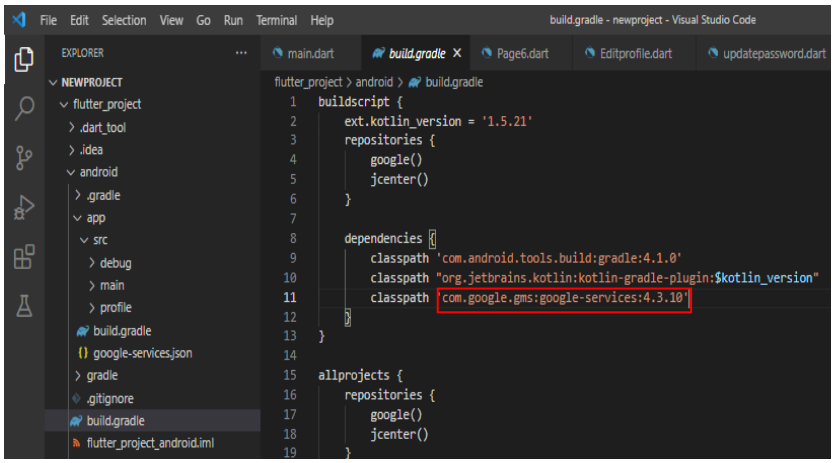


รูปที่ 3.29 ตัวอย่างการวาง google-serviecs.json ไว้ในโครงงาน

7) คัดลอกในส่วน classpath และนำไปวางไว้ในโครงงานใน buildgradle

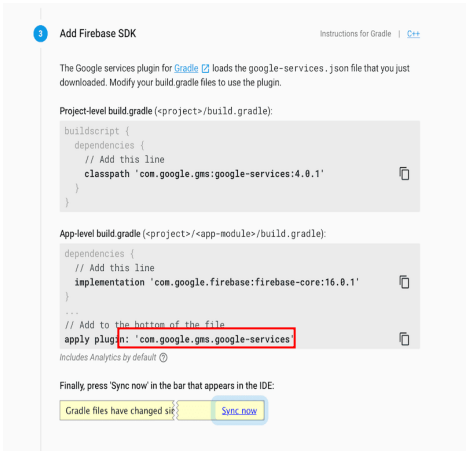


รูปที่ 3.30 ตัวอย่างการคัดลอก classpath

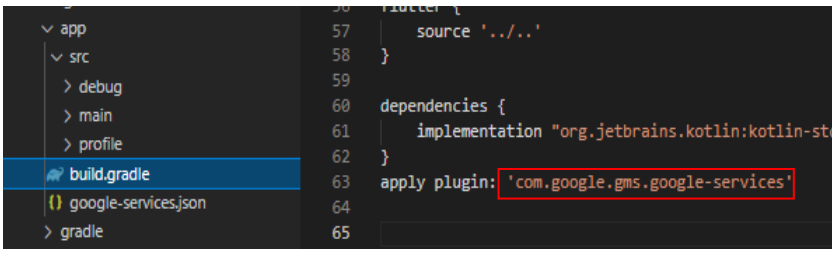


รูปที่ 3.31 ตัวอย่างการวาง classpath ใน buildgradle

8) คัดลอก apply plugin และนำไปวางไว้ในโครงงานในโฟลเดอร์ app ใน build.gradle

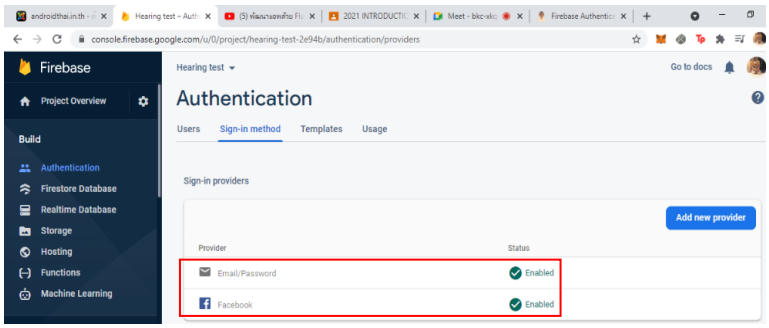


รูปที่ 3.32 ตัวอย่างการคัดลอก apply plugin



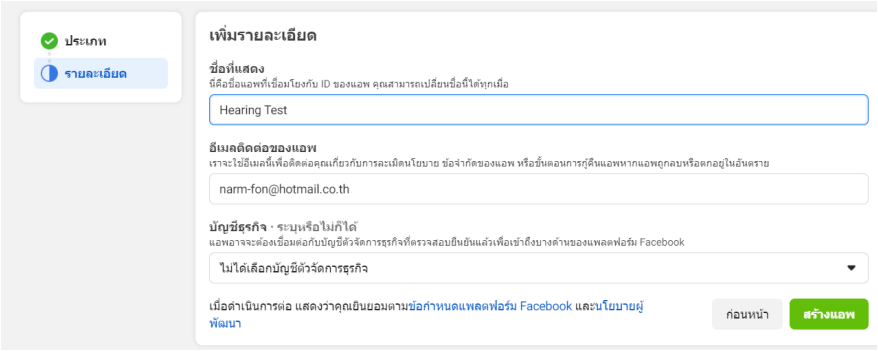
รูปที่ 3.33 ตัวอย่างการ วาง apply plugin ใน build.gradle

9) เปิดใช้งาน Authentication Sign-in method โดยจะเปิดใช้งานสำหรับ Email/Password และ Facebook



รูปที่ 3.34 ตัวอย่างการเปิดใช้งาน Authentication Sign-in

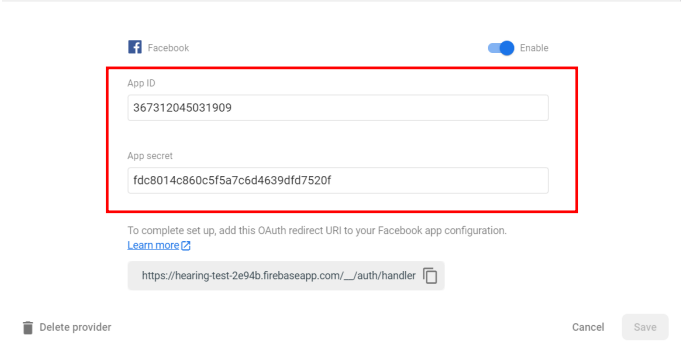
10) หลังจากที่เปิดใช้งานเข้าสู่ระบบด้วย Facebook แล้วให้ไปที่ Facebook for developers ทำการสร้างแอปพลิเคชัน



รูปที่ 3.35 ตัวอย่างการสร้างแอปพลิเคชันบน Facebook

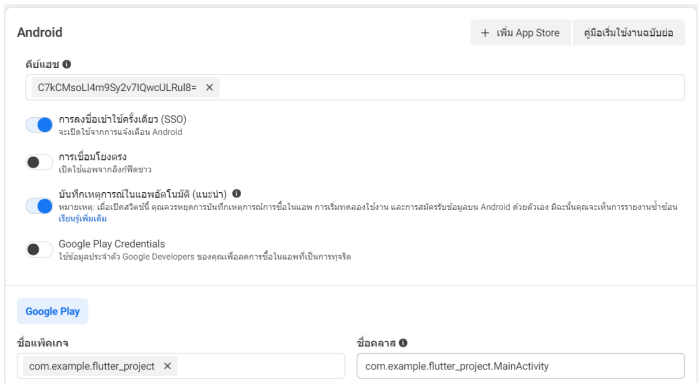
11) หลังจากที่สร้างแอปพลิเคชันสำเร็จให้ไปที่ข้อมูลพื้นฐาน คัดลอก ID ของแอปพลิเคชัน และข้อมูลลับของแอปพลิเคชันไปวางใน Firebase Authentication Sign-in method Facebook

รูปที่ 3.36 ข้อมูลพื้นฐานของแอปพลิเคชันบน Facebook



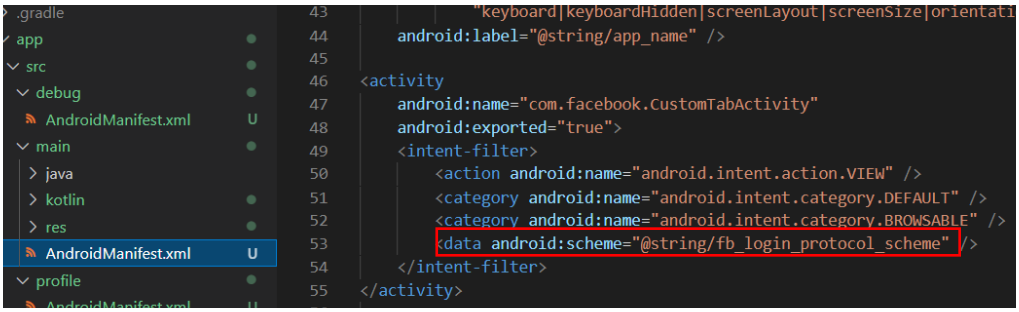
รูปที่ 3.37 ข้อมูลพื้นฐานใน Firebase Authentication Sign-in method Facebook

12) ตั้งค่าใน Facebook developers โดยต้องตั้งค่าให้ครบตามที่ facebook ต้องการ เช่น package name, class, hash key



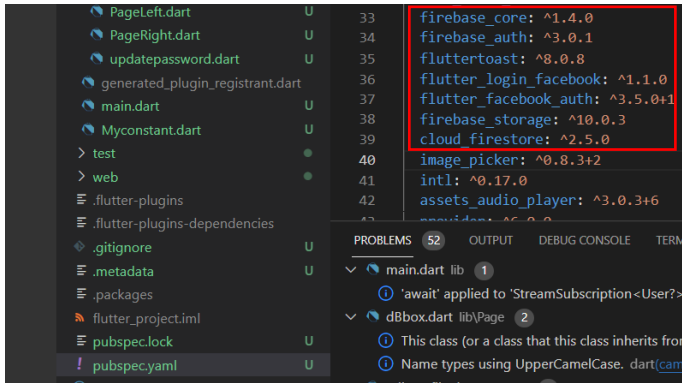
รูปที่ 3.38 การระบุค่าตามเงื่อนไขของ Facebook

13) กำหนดค่า Facebook ID ในโครงงานใน AndoridMainfest.xml



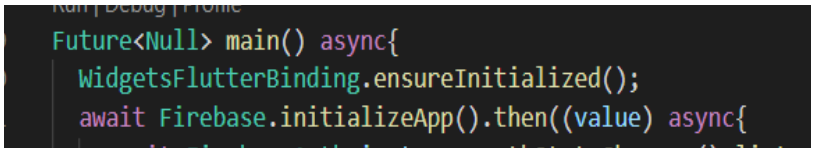
รูปที่ 3.39 Facebook ID ที่ใช้ในโครงงาน

14) ทำการติดตั้ง package ที่ pubspec.yaml เพื่อให้แอปพลิเคชันสามารถเชื่อมต่อโครงงานกับ Firebase ได้



รูปที่ 3.40 การติดตั้ง package

15) สร้างฟังก์ชัน Future ในหน้าหลักเพื่อทำการตรวจสอบว่าผู้ใช้งานแอปพลิเคชันได้มีการเชื่อมต่ออินเทอร์เน็ตอยู่ ซึ่งถ้าไม่ได้เชื่อมต่ออินเตอร์เน็ตผู้ใช้งานจะไม่สามารถสร้างบัญชีผู้ใช้หรือเข้าสู่ระบบได้เพราะจะไม่สามรถเชื่อมต่อกับ Firebase ได้

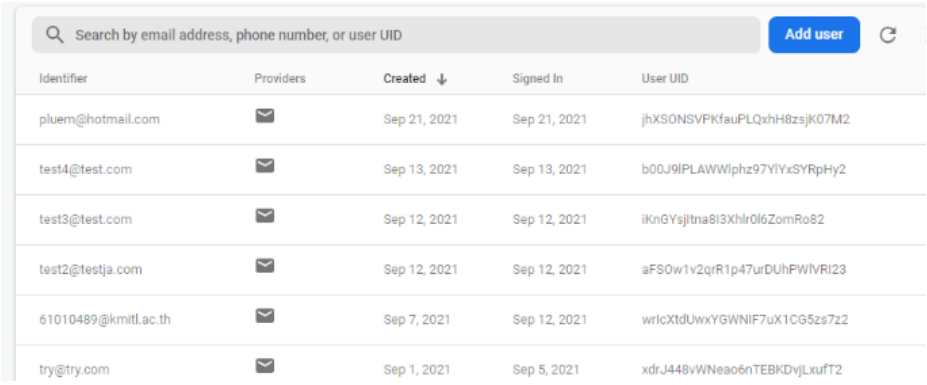


รูปที่ 3.41 การตรวจสอบการเชื่อมอินเทอร์เน็ตของผู้ใช้งาน

16) การเก็บข้อมูลการสร้างบัญชีผู้ใช้ด้วย Firebase Auth โดยใช้คำสั่ง createUserWithEmailAndPassword ในการสร้างบัญชีเพื่อเก็บข้อมูลอีเมลและรหัสผ่านไว้ใน Firebase เมื่อสร้างบัญชีสำเร็จผู้ใช้แต่ละคนจะมี uid เป็นของตัวเองโดยของแต่ละคนจะแตกต่างกัน

```
onPressed: () async {
  if (formKey.currentState!.validate()) {
    formKey.currentState!.save();
    try {
      await FirebaseAuth.instance
          .createUserWithEmailAndPassword(
              email: profile.email, password: profile.password)
          .then((value)
       {
        formKey.currentState!.reset();
        Fluttertoast.showToast(
            msg: 'User account has been created.',
            gravity: ToastGravity.TOP);

        Navigator.pushReplacement(context,
            MaterialPageRoute(builder: (context) {
          return information();
        })); // MaterialPageRoute
      });
    } on FirebaseAuthException catch (e) {
      print(e.message);
      Fluttertoast.showToast(
        msg: 'Email already in use.',
        gravity: ToastGravity.CENTER,
      );
```

รูปที่ 3.42 ตัวอย่างการสร้างบัญชีผู้ใช้งาน



รูปที่ 3.43 ข้อมูลผู้ใช้ที่สร้างบัญชีใน Firebase Auth

17) การเข้าสู่ระบบผู้ใช้ด้วยอีเมลและรหัสผ่านจะคล้ายกับหน้าสร้างบัญชีผู้ใช้ โดยจะใช้คำสั่งในการเข้าสู่ระบบเป็น signInWithEmailPassword โดย Firebase Auth จะมีอีเมลของผู้ใช้ที่ลงทะเบียนไว้แล้ว และในส่วนของการออกจากระบบบัญชีผู้ใช้จะใช้คำสั่ง signOut โดยใช้ Firebase Auth

```
onPressed: () async {
  if (formKey.currentState!.validate()) {
    formKey.currentState!.save();
    try {
      await FirebaseAuth.instance
          .signInWithEmailAndPassword(
              email: profile.email, password: profile.password)
          .then((value) {
        formKey.currentState!.reset();
        Navigator.push(context,
            MaterialPageRoute(builder: (context) {
          return MyHomePage(title: 'title');
        })); // MaterialPageRoute
      });
    } on FirebaseAuthException catch (e) {
      print(e.message);
      Fluttertoast.showToast(
          msg: e.code, gravity: ToastGravity.CENTER);
    }
```

รูปที่ 3.44 การเข้าสู่ระบบด้วยอีเมลและรหัสผ่าน

```
onPressed: () async{

await auth.signOut().then((value) async{
await  FacebookAuth.instance.logOut();
   Navigator.pushReplacement(context,
       MaterialPageRoute(builder: (context) {
     return Homescreen();
   })); // MaterialPageRoute
 });
},
```

รูปที่ 3.45 การออกจากระบบบัญชีผู้ใช้งาน

18) การสร้างบัญชีผู้ใช้ด้วย Facebook และการเข้าสู่ระบบด้วย Facebook จะใช้คำสั่ง signInWithCredential โดยใช้ Firebase Auth และในส่วนการออกจากระบบบัญชีผู้ใช้ FacebookAuth logout

```
final res = await fb.logIn(permissions: [
  FacebookPermission.publicProfile,
  FacebookPermission.email,
]);
```

รูปที่ 3.46 การประกาศใช้ facebooklogin ในหน้าสร้างบัญชีผู้ใช้และเข้าสู่ระบบผู้ใช้

```
switch (res.status) {
  case FacebookLoginStatus.success:
    final FacebookAccessToken? accessToken = res.accessToken;
    print('Access token: ${accessToken!.token}');
    final AuthCredential credential =
        FacebookAuthProvider.credential(accessToken.token);
    final result = await FirebaseAuth.instance
        .signInWithCredential(credential);

    print('${result.user} is now logged in');

    // Get profile data
    final profile = await fb.getUserProfile();
    print('Hello, ${profile!.name}! You ID: ${profile.userId}');

    // Get user profile image url
    final imageUrl = await fb.getProfileImageUrl(width: 100);
    print('Your profile image: $imageUrl');
```

รูปที่ 3.47 การเข้าสู่ระบบบัญชีผู้ใช้งานด้วย Facebook

```
onPressed: () async{

  await auth.signOut().then((value) async{
  await  FacebookAuth.instance.logOut();
     Navigator.pushReplacement(context,
         MaterialPageRoute(builder: (context) {
       return Homescreen();
     })); // MaterialPageRoute
   });
},
```

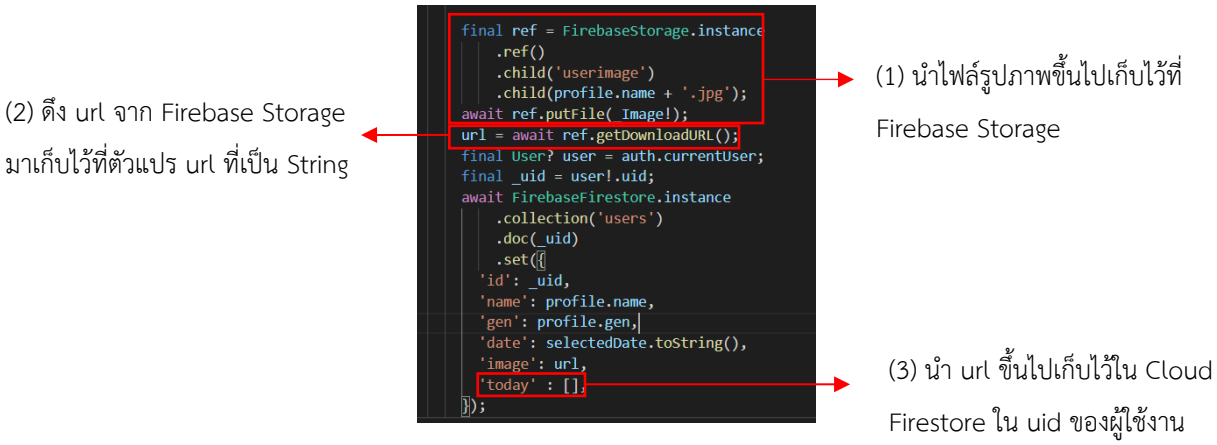
รูปที่ 3.48 การออกจากระบบบัญชีผู้ใช้งานด้วย Facebook

19) หลังจากที่ทำการสร้างบัญชีผู้ใช้แล้ว จะเข้าสู่หน้ากรอกข้อมูลส่วนตัวโดยหน้านี้จะมีการใส่รูปภาพ, ชื่อ, วันเกิด และเพศ ซึ่งหน้ากรอกข้อมูลส่วนตัวจะมีลักษณะเหมือนกับหน้าแก้ไขข้อมูลส่วนตัว

- รูปภาพที่ผู้ใช้เลือกจากอัลบั้มในโทรศัพท์หรือจากการถ่ายรูป รูปภาพส่วนนี้จะเก็บเป็นไฟล์แล้วนำขึ้นสู่ Firebase Storage โดยรูปภาพของผู้ใช้งานจะอยู่ในอัลบั้ม userimage โดยรูปภาพผู้ใช้งานชื่อไฟล์ของรูปภาพจะเป็นชื่อของผู้ใช้งาน หลังจากนำขึ้นสู่ Firebase Storage จะได้ url ของรูปภาพและนำค่า url ของรูปภาพลงมาเก็บใน Firestore Database ของข้อมูลผู้ใช้

```
File? _Image;
```

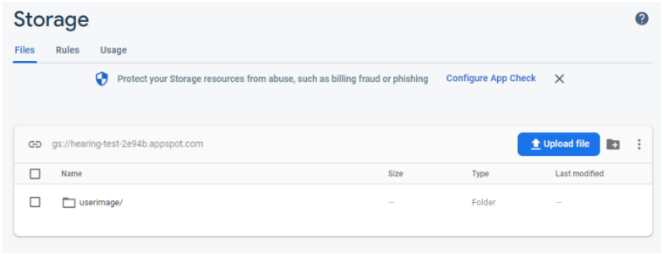
รูปที่ 3.49 ตัวอย่างการสร้างตัวแปร _Image เพื่อเก็บไฟล์รูปภาพ



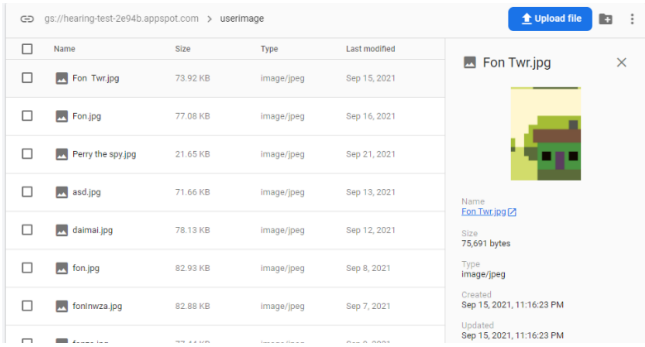

```
final ref = FirebaseStorage.instance
    .ref()
    .child('userimage')
    .child(profile.name + '.jpg');
await ref.putFile( Image!);
url = await ref.getDownloadURL();
final User? user = auth.currentUser;
final _uid = user!.uid;
await FirebaseFirestore.instance
    .collection('users')
    .doc(_uid)
    .set({
  'id': _uid,
  'name': profile.name,
  'gen': profile.gen,
  'date': selectedDate.toString(),
  'image': url,
  'today' : [],
});
```

รูปที่ 3.50 การนำไฟล์รูปภาพขึ้น Firebase Storage

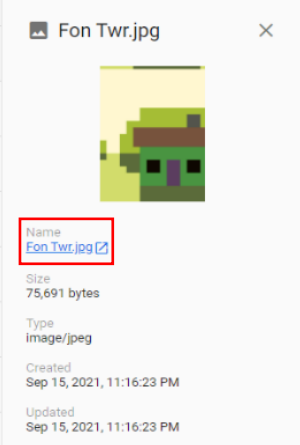
จากรูปที่ 3.50 (1) นำไฟล์รูปภาพเก็บขึ้นไปไว้ใน Firebase Storage โดยชื่อไฟล์รูปภาพจะเป็นของผู้ใช้งาน (2) ดึง url จาก Firebase Storage มาเก็บไว้ที่ตัวแปร url โดยที่ตัวแปร url จะเก็บค่าเป็น String (3) นำตัวแปร url ขึ้นไปเก็บไว้ใน Cloud Firestore โดยจะเก็บใน uid ของผู้ใช้งาน



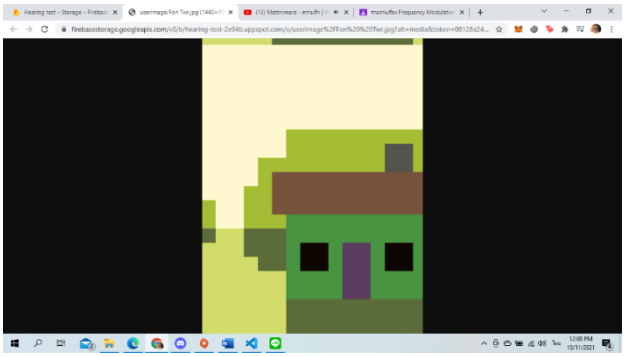
รูปที่ 3.51 อัลบั้ม userimage ที่เก็บไฟล์รูปภาพผู้ใช้งาน



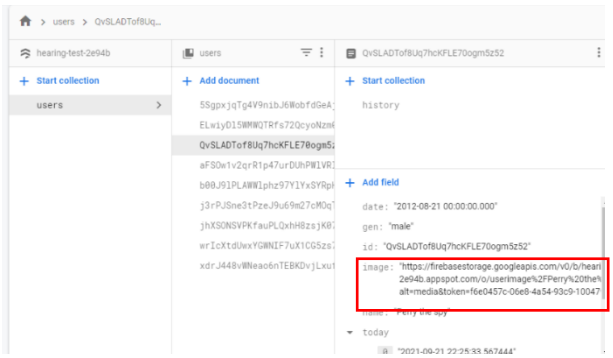
รูปที่ 3.52 ตัวอย่างชื่อไฟล์รูปภาพของผู้ใช้งานแต่ละผู้ใช้งาน



รูปที่ 3.53 ตัวอย่าง url รูปภาพของผู้ใช้งาน



รูปที่ 3.54 ตัวอย่างรูปภาพของผู้ใช้งานที่ได้จากการกรอกข้อมูลของผู้ใช้



รูปที่ 3.55 ตัวอย่าง url รูปภาพของผู้ใช้งานที่เก็บไว้ใน Cloud Firestore

- ชื่อผู้ใช้งาน, วันเกิด และเพศ จะเก็บไว้ในตัวแปร profile.name, profile.date และ profile.gen ซึ่งจะเก็บเป็น String และนำขึ้นสู่ Cloud Firestore ของ uid ผู้ใช้งาน

```
await FirebaseFirestore.instance
    .collection('users')
    .doc(_uid)
    .set({
  'id': _uid,
  'name': profile.name,
  'gen': profile.gen,
  'date': selectedDate.toString(),
  'image': url,
  'today' : [],
});
```

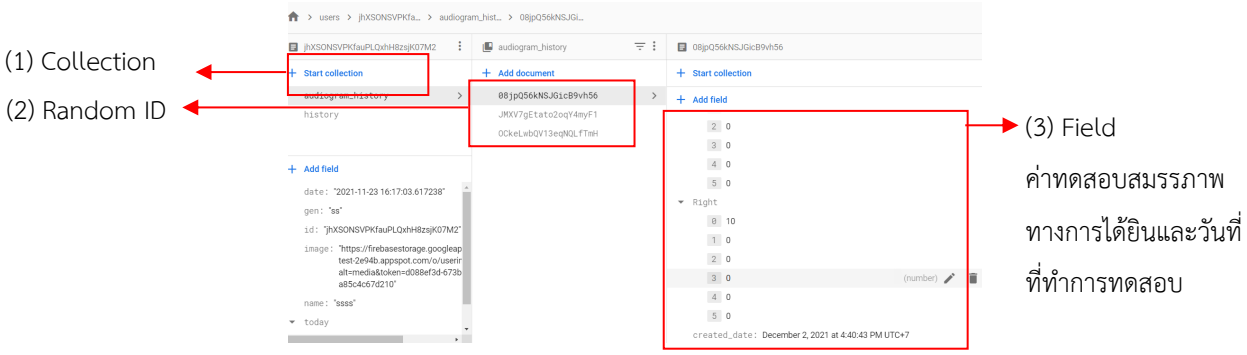
รูปที่ 3.56 การนำชื่อผู้ใช้งาน, วันเกิด และเพศขึ้นสู่ Cloud Firestore ตาม uid

```
date: "2012-08-21 00:00:00.000"
gen: "male"
id: "QvSLADTof8Uq7hcKFLE70ogm5z52"
image: "https://firebasestorage.googleapis.com/v0/b/hea
        2e94b.appspot.com/o/userimage%2FPerry%20th
        alt=media&token=f6e0457c-06e8-4a54-93c9-1004
name: "Perry the spy"
```

รูปที่ 3.57 ตัวอย่างข้อมูลของผู้ใช้งานบน Cloud Firestore

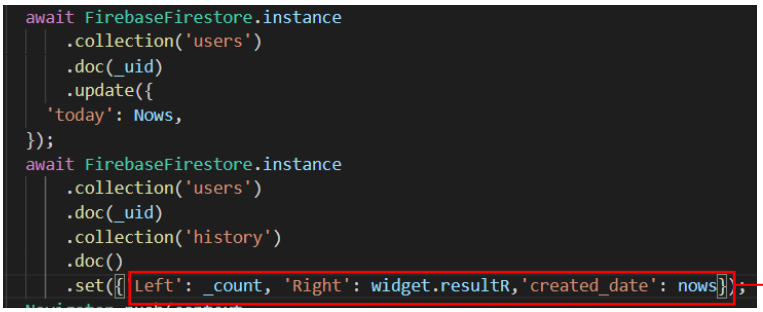
20) การส่งค่าผลที่ได้จากการทดสอบสมรรถภาพทางการได้ยินไปยังFirebase

- สร้าง collection audiogram_history แล้วสร้าง document เป็น random id โดยภายใน random id จะมี field หูซ้ายและหูขวาที่ทำการเก็บค่าการทดสอบสมรรถภาพทางการได้ยิน และมี created_date เป็นตัวเก็บวันที่ที่ทำการทดสอบ



รูปที่ 3.58 การเก็บข้อมูลในการทดสอบสมรรถภาพทางการได้ยินบน Firebase

- การส่งค่าการทดสอบสมรรถภาพทางการได้ยินขึ้นไปเก็บไว้บน Cloud Firestore โดยค่าทดสอบจะเก็บใน (3) จากรูปที่ ในหูซ้าย, หูขวาและเก็บวันที่ที่ทำการทดสอบ




```
await FirebaseFirestore.instance
    .collection('users')
    .doc(_uid)
    .update({
  'today': Nows,
});
await FirebaseFirestore.instance
    .collection('users')
    .doc(_uid)
    .collection('history')
    .doc()
    .set({'Left': _count, 'Right': widget.resultR,'created_date': nows});
```

รูปที่ 3.59 การเก็บข้อมูลในการทดสอบสมรรถภาพทางการได้ยิน

3.1.5.2 การดึงข้อมูลจากฐานข้อมูล

1) หน้าประวัติผู้ใช้งานจะทำการดึงข้อมูลผู้ใช้งานจากฐานข้อมูล Firebase ที่ทำการเก็บข้อมูลผู้ใช้ไว้ใน Cloud Firestore ทำการดึงข้อมูลผู้ใช้ลงมาเก็บไว้ที่ตัวแปร profile.name, profile.date, profile.gen และ _userimage

```
void getdata() async {
  // ignore: unused_local_variable
  User? user = auth.currentUser;

  _uid = user!.uid;
  print('user.email ${user.email}');
  final DocumentSnapshot userDoc =
      await FirebaseFirestore.instance.collection('users').doc(_uid).get();
  setState(() {
    profile.name = userDoc.get('name');
    print('name ${profile.name}');
    profile.gen = userDoc.get('gen');
    profile.date = userDoc.get('date');
    _userimage = userDoc.get('image');
    date = profile.date.split("");
  });
```

รูปที่ 3.60 การดึงข้อมูลผู้ใช้งานจากฐานข้อมูล

2) แสดงรูปภาพผู้ใช้งานเมื่อทำการดึงลงมาจากฐานข้อมูล

```
CircleAvatar(
  backgroundImage: NetworkImage(_userimage ??
      'https://cdn.pixabay.com/photo/2015/10/05/22/37/blank-profile-picture-973460_1280.png'),
  radius: 70,
),
```

รูปที่ 3.61 การดึงรูปภาพผู้ใช้งานมาแสดง

3) แสดงชื่อผู้ใช้งานเมื่อทำการดึงลงมาจากฐานข้อมูล

```
Text(
  profile.name,
  style: TextStyle(
      fontSize: 20,
      color: Myconstant.light,
      fontWeight: FontWeight.bold),
),
```

รูปที่ 3.62 การดึงชื่อผู้ใช้งานมาแสดงเป็นข้อความ

4) แสดงวันเกิดผู้ใช้งานเมื่อทำการดึงลงมาจากฐานข้อมูล

```
Text(
  "${profile.date.split(" ")[0]}",
  style: TextStyle(
      fontSize: 18, color: Myconstant.light),
  textAlign: TextAlign.center,
),
```

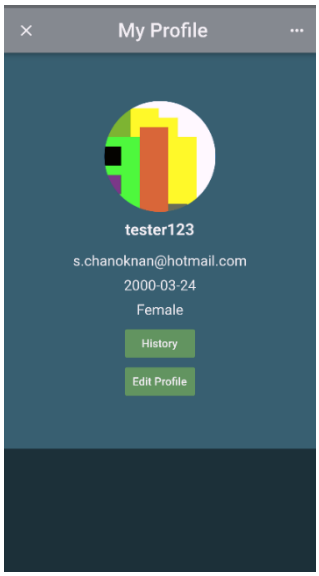
รูปที่ 3.63 การดึงวันเกิดผู้ใช้งานมาแสดงเป็นข้อความ

5) แสดงเพศของผู้ใช้งานเมื่อทำการดึงลงมาจากฐานข้อมูล

```
Text(
  profile.gen,
  style: TextStyle(
      fontSize: 18, color: Myconstant.light),
  textAlign: TextAlign.center,
),
```

รูปที่ 3.64 การดึงเพศของผู้ใช้งานมาแสดงเป็นข้อความ

6) แสดงข้อมูลของผู้ใช้งานในแอปพลิเคชัน



รูปที่ 3.65 ตัวอย่างหน้าข้อมูลของผู้ใช้งาน

7) เรียกฟังก์ชัน Firebase เพื่อเรียกข้อมูลของผู้ใช้งาน โดยที่หน้า History จะแสดงวันที่ที่ผู้ใช้งานทำการทดสอบสมรรถภาพทางการได้ยิน

```
await FirebaseFirestore.instance
    .collection('users')
    .doc(_uid)
    .collection('audiogram_history')
    .get()
    .then(
  (value) {
    setState(
      () {
        if (value.docs.length > 0) {
          value.docs.forEach(
            (element) {
              FirebaseFirestore.instance
                  .collection('users')
                  .doc(_uid)
                  .collection('audiogram_history')
                  .doc(element.id)
                  .get();
              today = element.get('created_date');
              print('object ${today}');
              DateTime date = today.toDate();
              time.add(date);
            },
```

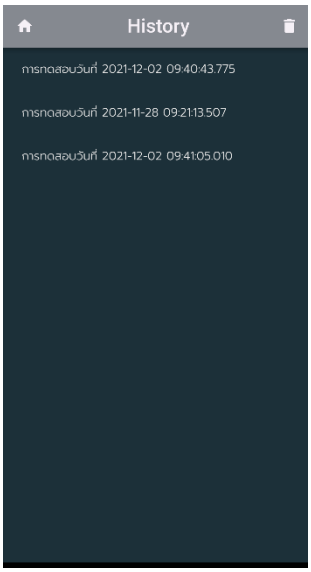
วันที่ที่ทำการทดสอบสมรรถภาพทางการได้ยิน

รูปที่ 3.66 วันที่ที่ผู้ใช้งานทำการทดสอบสมรรถภาพทางการได้ยิน

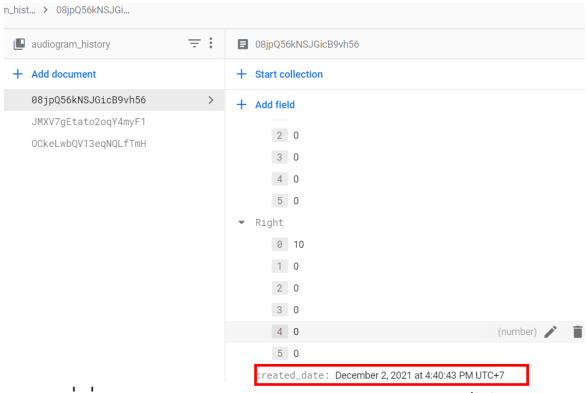
```
child: Text(
  'การทดสอบวันที่ ${time[index]}',
  style: TextStyle(
      color: Myconstant.gray,
      fontFamily: 'Prompt',
      fontSize: 15), // TextStyle
```

แสดงวันที่ที่ทำการทดสอบสมรรถภาพทางการได้ยิน

รูปที่ 3.67 การแสดงวันที่ที่การทดสอบสมรรถภาพทางการได้ยิน



รูปที่ 3.68 หน้า History ของผู้ใช้งาน



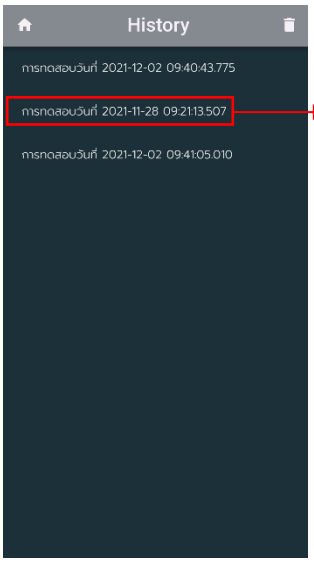
รูปที่ 3.69 วันที่ที่ทำการทดสอบสมรรถภาพทางการได้ยินบน Firebase

8) เรียกฟังก์ชัน Firebase เพื่อเรียกข้อมูลค่าการทดสอบสมรรถภาพทางการได้ยินของหูซ้ายและหูขวาของผู้ใช้งาน

```
Future getdata2(DateTime pickedDate) async {
  User? user = auth.currentUser;
  _uid = user!.uid;
  await FirebaseFirestore.instance
      .collection('users')
      .doc(_uid)
      .collection('audiogram_history')
      .where("created_date", isEqualTo: pickedDate)
      .get()
      .then((userDoc2) {
    setState(() {
      if (userDoc2.docs.length > 0) {
        userDoc2.docs.forEach((element) {
          history = List.from(element.get('Left'));
          history2 = List.from(element.get('Right'));
          print('my Left ${history}');
          print('my Right ${history2}');
        });
```

รูปที่ 3.70 ข้อมูลค่าการทดสอบสมรรถภาพทางการได้ยินของหูซ้ายและหูขวา

9) หน้า History สามารถกดเข้าไปดูผลการทดสอบสมรรถภาพทางการได้ยิน โดยเมื่อทำการกดวันที่ที่ทำการทดสอบจะทำการเรียกข้อมูลใน Cloud Firestore ของผู้ใช้ แล้วทำการส่งค่าไปยังหน้า Audiogram เพื่อแสดงผลการทดสอบสมรรถภาพทางการได้ยินในการทดสอบสมรรถภาพทางการได้ยินของวันที่ที่ทำการทดสอบ

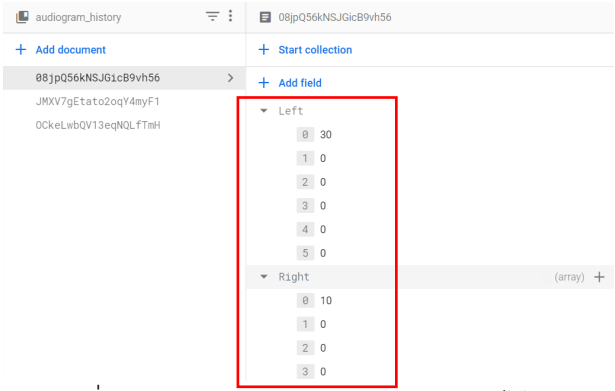


เมื่อทำการกดจะเรียกฟังก์ชัน Firebase เพื่อทำการส่งค่าการทดสอบสมรรถภาพทางการได้ยินไปยังหน้า Audiogram

รูปที่ 3.71 ตัวอย่างหน้าประวัติของผู้ใช้งาน

```
onPressed: () async {
  await getdata2(time[index]);

  Navigator.push(
      context,
      MaterialPageRoute(
          builder: (context) => Audiogram(
                result: history2,
                result2: history,
                sumall: history2[1] +
                    history2[2] +
                    history2[3] +
                    history[1] +
                    history[2] +
                    history[3],
              ))); // Audiogram // MaterialP
```

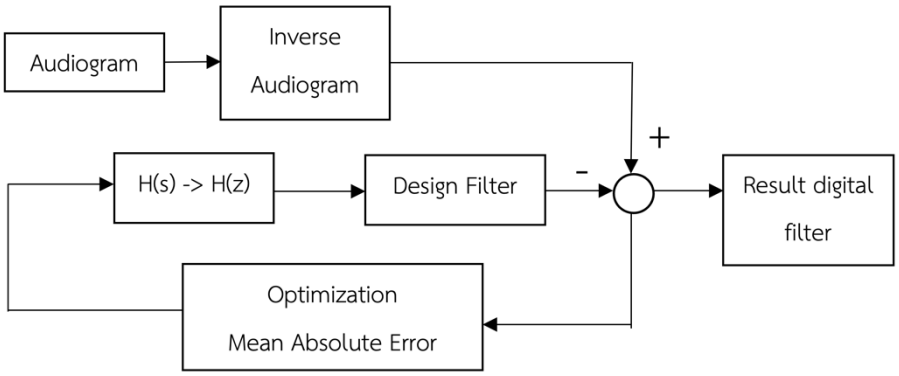
รูปที่ 3.72 การส่งค่าผลการทดสอบสมรรถภาพทางการได้ยินไปยังหน้า Audiogram



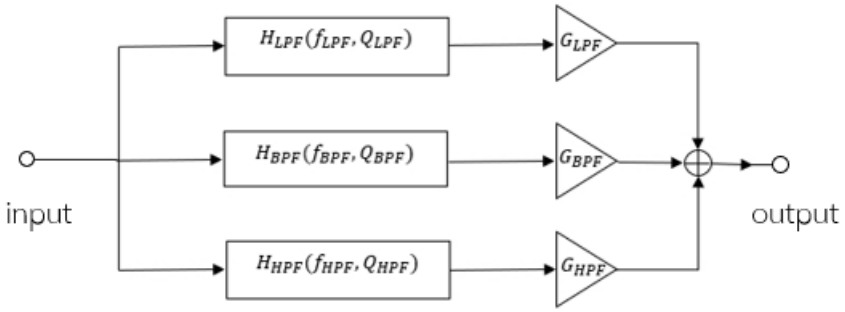
รูปที่ 3.73 ค่าทดสอบสมรรถภาพทางการได้ยิน

3.1.6 การออกแบบวงจรกรองความถี่สำหรับเครื่องช่วยฟัง

การออกแบบวงจรกรองสัญญาณความถี่แบบดิจิทัลสำหรับเครื่องช่วยฟังจะใช้โปรแกรมแมทแลป และโปรแกรมไพธอน โดยมีวงจรกรองสัญญาณความถี่ต้นแบบมาจากวงจรกรองแบบความถี่แอนะล็อกที่ใช้สมการไบควอดเดรติก และใช้การแปลงไบลิเนียร์สำหรับแปลงระนาบ s (s-plane) ให้เป็นระนาบ z (z-plane) เพื่อให้เป็นวงจรกรองความถี่แบบดิจิทัล โดยในปริญญานิพนธ์นี้จะใช้เป็นวงจรกรองสัญญาณความถี่แบบดิจิทัลประเภท IIR เนื่องจากมีการคำนวณที่มีความซับซ้อนน้อยกว่าและมีประสิทธิภาพมากกว่าวงจรกรองสัญญาณความถี่ประเภท FIR ที่อันดับเท่า ๆกัน เมื่อพิจารณาที่ผลตอบสนองทางความถี่ โดยบล็อกไดอะแกรมโครงสร้างการออกแบบวงจรกรองสัญญาณความถี่แบบดิจิทัลดังรูปที่ 3.74



รูปที่ 3.74 บล็อกไดอะแกรมโครงสร้างการออกแบบวงจรกรองความถี่แบบดิจิทัล



รูปที่ 3.75 โครงสร้างของการออกแบบวงจรกรองสัญญาณความถี่สำหรับเครื่องช่วยฟัง

3.1.6.1 การกำหนดพารามิเตอร์เพื่อใช้สำหรับออกแบบวงจรกรองความถี่

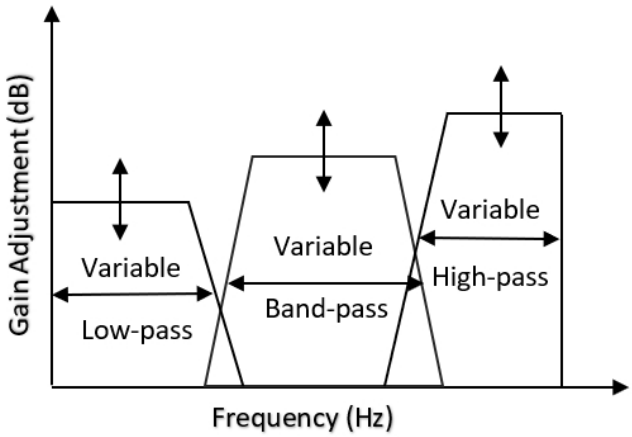
วงจรกรองสัญญาณแอนะล็อก ต้นแบบที่ใช้คือ สมการไบควอตเดรติก โดยมีฟังก์ชันการถ่ายโอน (Transfer function) ได้ดังสมการที่ 3.1

$$H(z, x) = G_{LPF}H_{LPF}\left(z, f_{LPF}, Q_{LPF}\right)+G_{BPF}H_{BPF}\left(z, f_{BPF}, Q_{BPF}\right)+G_{HPF}H_{HPF}\left(z, f_{HPF}, Q_{HPF}\right) \quad (3.1)$$

โดย x คือเวกเตอร์ของพารามิเตอร์ที่เป็นตัวกำหนดลักษณะของวงจรกรองสัญญาณให้มีผลตอบสนองกับขนาดรูปแบบที่ไปใช้ชดเชยการสูญเสียทางการได้ยิน ซึ่งการออกแบบโดยวงจรกรองสัญญาณแอนะล็อก ต้นแบบเป็นสมการไบควอตเดรติกจะได้เวกเตอร์ x ดังนี้

$$X = \left[f_{LPF}\, f_{BPF}\, f_{HPF}\, Q_{LPF}\, Q_{BPF}\, Q_{HPF}\, G_{LPF}\, G_{BPF}\, G_{HPF}\right] \quad (3.2)$$

โดยเวกเตอร์ x มี 9 พารามิเตอร์ที่มาจากการหาค่าที่เหมาะสมที่สุด โดยมีค่าความถี่คัตออฟของแต่ละวงจรกรองความถี่ [f_{LPF} f_{BPF} f_{HPF}] ค่าคุณภาพของแต่ละวงจรกรองความถี่ [Q_{LPF} Q_{BPF} Q_{HPF}] และค่าอัตราการขยายของแต่ละวงจรกรองความถี่ [G_{LPF} G_{BPF} G_{HPF}] ซึ่งจะประกอบกันเป็นดังรูปที่ 3.76 เพื่อนำมาใช้กำหนดลักษณะของวงจรกรองสัญญาณให้มีผลตอบสนองทางขนาดรูปแบบที่ไปใช้ชดเชยการสูญเสียทางการได้ยินตามรูปที่ 3.76



รูปที่ 3.76 แนวคิดการสร้างผลตอบสนองทางขนาดรูปแบบที่ไปใช้ชดเชยการสูญเสียทางการได้ยิน

โดยจะทำการสังเกตกราฟการได้ยินเพื่อทดลองกำหนดค่า f_{LPF} f_{BPF} f_{HPF} และ Q_{LPF} Q_{BPF} Q_{HPF} และค่า G_{LPF} G_{BPF} G_{HPF} ที่ทำหน้าที่เสมือนวงจร Equalizer โดยแต่ละตัวจะถูกกำหนดเพื่อให้วงจรกรองความถี่ที่ออกแบบมาใกล้เคียงกับกราฟการได้ยินต้นแบบ

3.1.6.2 การออกแบบวงจรกรองความถี่โดยใช้หลักการหาค่าที่เหมาะสมที่สุด

สำหรับการออกแบบวงจรกรองความถี่แบบดิจิทัลที่ได้ค่า f, Q, G มาจากการหาค่าที่เหมาะสมที่สุด (Optimization) ผ่านการเรียกใช้คำสั่ง fminsearch() ในโปรแกรมแมทแลป ซึ่งคำสั่งนี้จะใช้หลักการ Nelder Mead ที่ได้กล่าวไว้ในหัวข้อที่ 2.16.2 ในบทที่ 2

fminsearch() เป็นคำสั่งที่ใช้สำหรับการหาค่าต่ำสุด (Minimum) สำหรับการหาค่าที่เหมาะสมที่สุดเมื่อฟังก์ชันที่ใช้นั้นไม่มีเงื่อนไข (Unconstrained) และเป็นกำหนดการที่ไม่เป็นเชิงเส้น (Nonlinear Programming) ซึ่งคำสั่ง fminsearch() จะใช้หลักการ Nelder Mead ในการหาค่าที่เหมาะสมที่สุด

- ตัวอย่างการเรียกใช้คำสั่ง fminsearch()

1. x = fminsearch(fun, x0)

โดย x0 คือ การกำหนดจุดเริ่มต้นที่ x0 และ การหาค่าต่ำสุดสัมพัทธ์ของฟังก์ชันfun โดยที่ x0 สามารถเป็นสเกลาร์ เวกเตอร์ หรือเมทริกซ์

fun คือ ฟังก์ชัน handle (Function handle) หรือเรียกว่า ฟังก์ชันที่ไม่ระบุชื่อ (Anonymous function) ที่มีอินพุต X และให้ค่ากลับมาในรูปสเกลาร์

2. [x, fval] = fminsearch(...)

โดย fminsearch() จะส่งค่าของฟังก์ชันวัตถุประสงค์ (Objective function) กลับมาโดยอธิบายในรูปของ ‘fun’ ที่จุด *x* ยกตัวอย่างเช่น ให้ fun เป็นฟังก์ชันวัตถุประสงค์จะได้ดังนี้

X = fminsearch(sin, 3)

เป็นการหาค่าต่ำสุดของฟังก์ชัน sin ที่เข้าใกล้ 3 ในกรณีนี้ sin คือฟังก์ชันที่จะให้ค่าสเกลาร์กลับมา โดยที่ฟังก์ชัน sin จะหาค่าที่จุด x

- การประยุกต์หลักการหาค่าที่เหมาะสมที่สุด (Optimization)

การประยุกต์หลักการหาค่าที่เหมาะสมที่สุด (Optimization) โดยการเขียนบนโปรแกรมแมทแลปเพื่อประยุกต์ใช้กับการออกแบบวงจรกรองความถี่ของโครงงานสามารถทำได้ตามขั้นตอนดังนี้

1. การสร้างตัวแปร (Design variable)

สร้างตัวแปรที่ต้องการหาเพื่อเป็นทำเป็นวงจรกรองความถี่ที่ออกแบบประสิทธิภาพมากที่สุดเช่นกำหนด ค่า f0 = x1 ค่า Q = x2 เพื่อนำไปใช้ในกระบวนการถัดไป

2. ฟังก์ชันวัตถุประสงค์ (Objective function)

ในการสร้างฟังก์ชันเพื่อนำมาคำนวณจะต้องเป็นกระบวนการที่มีตัวแปรที่ได้กำหนดค่าในขั้นตอน Design variable และจำต้องมีการคำนวณค่าของความแตกต่างระหว่างวงจรกรองความถี่ในอุดมคติ กับ วงจรกรองความถี่ ที่ได้ทำการออกแบบ โดยใช้วิธีการคำนวณความผิดพลาด Mean Absolute Error เป็นการทดสอบในขั้นต้น ซึ่งสามารเรียกใช้ฟังก์ชันดังกล่าวตามรูปที่ 3.24

ดังนั้นจึงสามารถสร้าง Objective function ได้โดยการสร้าง วงจรกรองความถี่ที่ขึ้นกับวงจรกรองความถี่ในอุดมคติและการคำนวณค่าความผิดพลาดด้วยวิธี Mean Absolute Error ตามสมการที่ 3.3

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \qquad (3.3)$$

โดย n = จำนวนจุดของข้อมูลทั้งหมดที่ใช้คำนวณ

y_i = ค่าที่ได้จากทฤษฎี

x_i = ค่าที่ได้จากการทดลอง

```
function [MAE] = Meanabserror(xi,yi)
    np = length(xi);
    ns = length(yi);
    meann = sum(abs(xi-yi));
    if np == ns
        MAE = meann/np;
    else
        MAE = inf;
    end
```

รูปที่ 3.77 ฟังก์ชันที่ใช้คำนวณค่า Mean Absolute Error

3. เงื่อนไข (Constrained)

เนื่องจากปริญญานิพนธ์นี้ไม่สามารถควบคุมขอบเขตของการสุ่มค่าได้ จึงจัดเป็น Unconstraint หรือไม่มีขอบเขตกำหนด จึงจะต้องใช้ทฤษฎี Nelder Mead เพื่อควบคุมการสุ่มให้ได้ค่า Error ที่ต้องการ โดยใช้ function fminserach ในการใช้คำนวณ

```
fun = @(x) Objectivepossssss(x(1),x(2),x(3),x(4),x(5),x(6),x(7),x(8),x(9));
[row,col] = fminsearch(fun,[4000,0.5,1,4000,0.5,1,4000,0.5,1]);
```

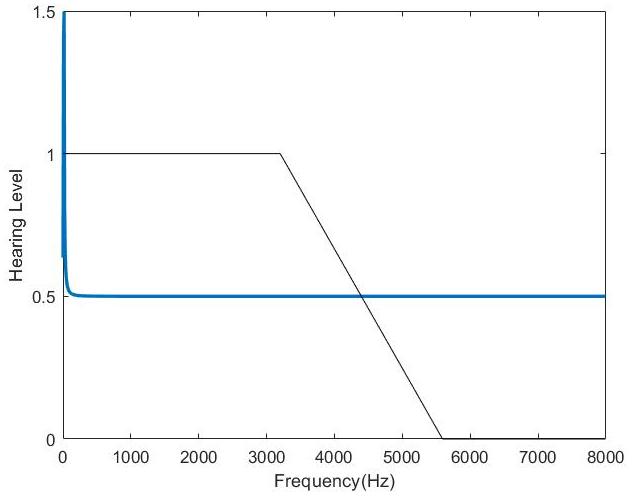
รูปที่ 3.78 การกำหนดค่าเริ่มต้นสำหรับคำสั่ง fminsearch()

3.1.6.3 การสังเกตเพื่อกำหนดค่าเริ่มต้น

เนื่องจากการทำงานของการ Optimization จำเป็นต้องมีค่าเริ่มต้น (Initial) หรือค่าเริ่มต้นของการสุ่มด้วยทฤษฎี Nelder-Mead ปริญญานิพนธ์นี้จึงมีข้อสงสัยเกี่ยวกับค่า Initial จึงได้ทำการทดลองเพื่อให้เห็นความแตกต่างในการกำหนดค่า Initial นี้ โดยขั้นแรกจะเป็นการกำหนดค่า Initial ที่เป็นค่าเดียวกันทั้งหมด

```
[row,col] = fminsearch(fun,[10,10,10,10,10,10,10,10,10]);
```

รูปที่ 3.79 ฟังก์ชัน fminsearch ที่กำหนดค่า Initial เป็นค่าเดียวกันทั้งหมด

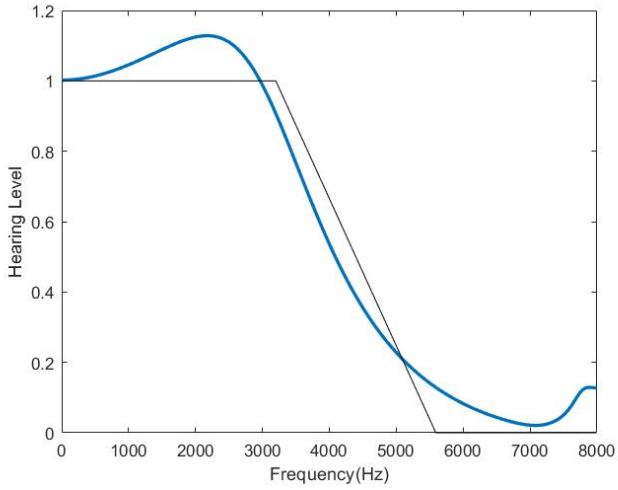


รูปที่ 3.80 Magnitude response ของฟังก์ชันที่กำหนดค่า Initial เป็นค่าเดียวกันทั้งหมด

ขั้นต่อไปจะกำหนดค่า Initial เป็นค่าประมาณที่เหมาะสมกับแต่ละ Parameter โดยจะให้ค่าความถี่จะมีค่าเป็น 4000 Hz ค่าของ Quality factor จะมีค่าเป็น 0.5 และค่าของ Scaling factor หรือ Gain จะมีค่าเป็น 1

```
[row,col] = fminsearch(fun,[4000,0.5,1,4000,0.5,1,4000,0.5,1]);
```

รูปที่ 3.81 ฟังก์ชัน fminsearch ที่กำหนดค่า Initial เป็นค่าประมาณที่เหมาะสม

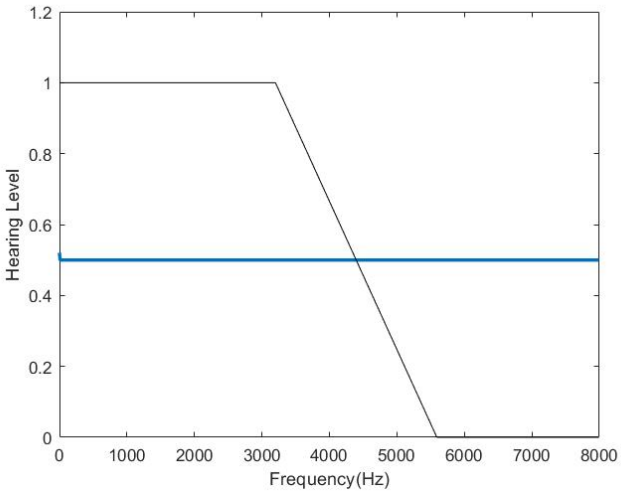


รูปที่ 3.82 Magnitude response ของฟังก์ชันที่กำหนดค่า Initial เป็นค่าประมาณที่เหมาะสม

ขั้นต่อไปได้ทำการลองเปลี่ยนค่า Initial เป็นค่าที่ทางโปรแกรม MATLAB ได้สุ่มค่ามาให้ในฟังก์ชัน rand() โดยจะมีค่าจาก 0 ถึง 1

```
[row,col] = fminsearch(fun,[rand(1,9)]);
```

รูปที่ 3.83 ฟังก์ชัน fminsearch ที่กำหนดค่า Initial เป็นค่าสุ่ม



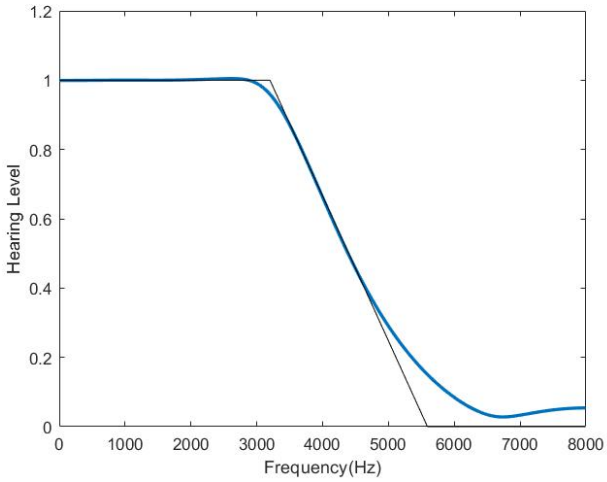
รูปที่ 3.84 Magnitude response ของฟังก์ชันที่กำหนดค่า Initial เป็นค่าสุ่ม

จากกการสังเกตผลของการกำหนดค่า Initial ข้างต้น จะเห็นได้ว่าการกำหนดค่าที่เหมาะสมจะทำให้ Magnitude response มีลักษณะที่ใกล้เคียงกับตัวต้นแบบ เนื่องจากในการ Optimization เพื่อหาจุดต่ำสุดของฟังก์ชัน จะขึ้นอยู่กับค่า Initial ของฟังก์ชัน fminsearch() เนื่องจากการเปลี่ยนค่า Initial ค่าต่ำสุดที่ฟังก์ชันคำนวณได้จะมีค่าแตกต่างกัน จึงต้องควบคุมการกำหนดค่า Initial ให้เหมาะสมกับการสุ่มในแต่ละครั้ง

และจากการสร้างวงจรกรองความถี่จะเห็นได้ว่า Error function ที่ใช้คือ Mean Absolute Error เพื่อ Magnitude response ที่มีลักษณะเหมือนกับตัวต้นแบบที่สุด

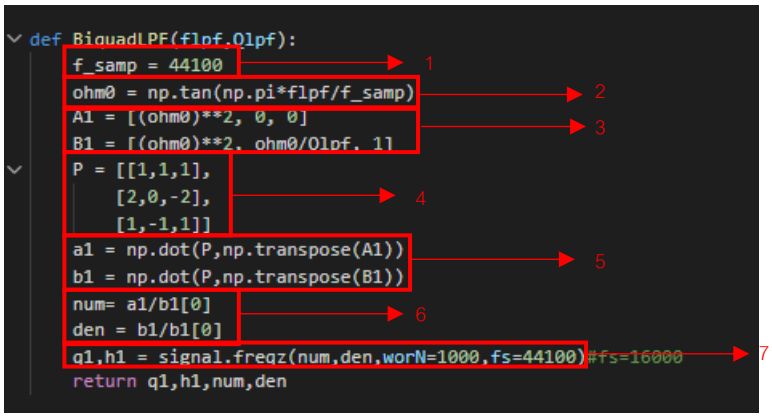
- การคำนวณค่าความผิดพลาด

ในการสร้างฟังก์ชันที่ใช้สำหรับคำนวณค่าความผิดพลาด ปริญญานิพนธ์นี้ได้เลือกวิธีการคำนวณค่าความผิดพลาด Mean Absolute Error โดยมีสูตรที่ใช้ในการคำนวณตามสมการที่ 3.3 โดยจะได้ Magnitude response ดังนี้



รูปที่ 3.85 Magnitude response ที่ใช้ Mean absolute error

โดยการนำเอาการออกแบบวงจรกรองสัญญาณความถี่แบบดิจิทัลสำหรับเครื่องช่วยฟังไปใช้บนแอปพลิเคชันจะใช้ภาษาไพธอนในการออกแบบวงจรกรองสัญญาณความถี่แบบดิจิทัลเพื่อความสะดวกในการประยุกต์ใช้บน Flutter โดยการออกแบบวงจรกรองสัญญาณความถี่แบบดิจิทัลด้วยภาษาไพธอน จะเหมือนกับที่เขียนใน MATLAB โดยเราจะออกแบบวงจรกรองสัญญาณความถี่ต่ำผ่าน, วงจรกรองสัญญาณความถี่สูงผ่าน และวงจรกรองสัญญาณผ่านแถบความถี่



รูปที่ 3.86 ฟังก์ชันที่ใช้ออกแบบวงจรกรองสัญญาณความถี่แบบต่ำผ่าน

จากรูปที่ 3.86

1) กำหนด $\mathrm{f\_samp} = 44100\ \mathrm{Hz}$ (Sampling Frequency)

2) หาค่า Ω_0 จากความสัมพันธ์ $\Omega_0 = \tan\left(\pi \frac{\mathrm{f}}{\mathrm{F}}\right)$

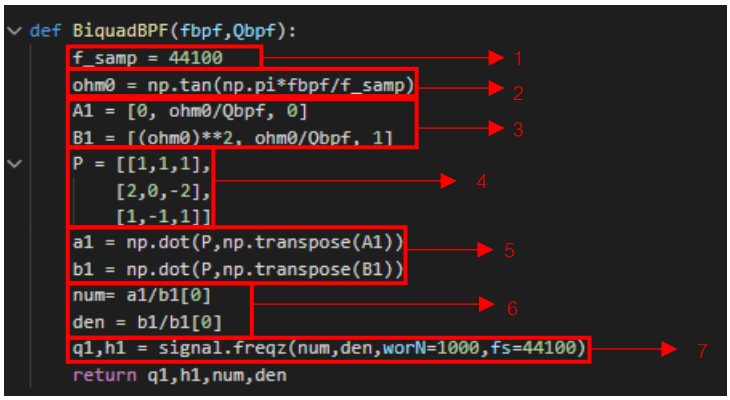
3) ค่า A คือ เศษของสัมประสิทธิ์ (Numerator coefficient) ในสมการ $\mathrm{H(s)}$ ในสมการที่ (1), (2) และ (3) ค่า B คือ ส่วนของสัมประสิทธิ์ (Denominator coefficient) ในสมการ $\mathrm{H(s)}$ ในสมการที่ (1), (2) และ (3)

4) P คือ ปาสคาลเมตริกซ์ (Pascal Matrix)

5) นำค่า A คูณกับ Pascal Matrix และนำค่า B คูณกับ Pascal Matrix

6) ตัวแปร num มาจาก Numerator คือค่าเศษสัมประสิทธิ์ในสมการ $\mathrm{H(z)}$ ที่ต้องหาค่า และ den มาจาก Denominator คือค่าส่วนสัมประสิทธิ์ในสมการ $\mathrm{H(z)}$ ที่จะต้องหาค่า วิธีการคือจะนำค่าสัมประสิทธิ์ของส่วนที่อยู่หน้าตัวกำลังสูงสุดหารทั้งสมการซึ่งคือค่า $\mathrm{b(0)}$ จาก Matrix b นำค่าที่ได้หารค่า a ทั้งหมดและหารค่า b ทั้งหมด

7) ค่า h1 คือค่า Frequency response มีค่าเป็น Complex number ค่า q1 คือค่า angular frequency vector ซึ่งมีค่า 0 - 9990 จะใช้ function $\mathrm{signal.freqz()}$ โดยมี 4 ตัวแปรคือ num คือค่าในข้อที่ 6, den คือค่าในข้อที่ 6 , worN คือค่าที่ต้องกำหนดเพราะเนื่องจากวงจรกรองความถี่ที่สร้างเป็น IIR filter design ซึ่งมีค่าไม่จำกัดจึงต้องกำหนดค่าที่ 1000 จุด, $\mathrm{f_s}$ คือค่าจากข้อที่ 1



รูปที่ 3.87 ฟังก์ชันที่ใช้ออกแบบวงจรกรองสัญญาณผ่านแถบความถี่

จากรูปที่ 3.87

1) กำหนด $\mathrm{f\_samp} = 44100\ \mathrm{Hz}$ (Sampling Frequency)

2) หาค่า Ω_0 จากความสัมพันธ์ $\Omega_0 = \tan\left(\pi \frac{\mathrm{f}}{\mathrm{F}}\right)$

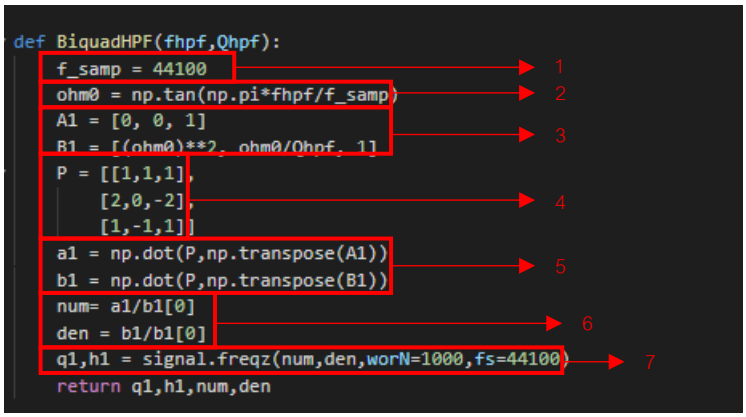
3) ค่า A คือ เศษของสัมประสิทธิ์ (Numerator coefficient) ในสมการ $\mathrm{H(s)}$ ในสมการที่ (1), (2) และ (3) ค่า B คือ ส่วนของสัมประสิทธิ์ (Denominator coefficient) ในสมการ $\mathrm{H(s)}$ ในสมการที่ (1), (2) และ (3)

4) P คือ ปาสคาลเมตริกซ์ (Pascal Matrix)

5) นำค่า A คูณกับ Pascal Matrix และนำค่า B คูณกับ Pascal Matrix

6) ตัวแปร num มาจาก Numerator คือค่าเศษสัมประสิทธิ์ในสมการ $\mathrm{H(z)}$ ที่ต้องหาค่า และ den มาจาก Denominator คือค่าส่วนสัมประสิทธิ์ในสมการ $\mathrm{H(z)}$ ที่จะต้องหาค่า วิธีการคือจะนำค่าสัมประสิทธิ์ของส่วนที่อยู่หน้าตัวกำลังสูงสุดหารทั้งสมการซึ่งคือค่า $\mathrm{b(0)}$ จาก Matrix b นำค่าที่ได้หารค่า a ทั้งหมดและหารค่า b ทั้งหมด

7) ค่า h1 คือค่า Frequency response มีค่าเป็น Complex number ค่า q1 คือค่า angular frequency vector ซึ่งมีค่า 0 - 9990 จะใช้ function $\mathrm{signal.freqz()}$ โดยมี 4 ตัวแปรคือ num คือค่าในข้อที่ 6, den คือค่าในข้อที่ 6 , worN คือค่าที่ต้องกำหนดเพราะเนื่องจากวงจรกรองความถี่ที่สร้างเป็น IIR filter design ซึ่งมีค่าไม่จำกัดจึงต้องกำหนดค่าที่ 1000 จุด, $\mathrm{f_s}$ คือค่าจากข้อที่ 1



รูปที่ 3.88 ฟังก์ชันที่ใช้ออกแบบวงจรกรองสัญญาณความถี่สูงผ่าน

จากรูปที่ 3.88

1) กำหนด $\mathrm{f\_samp} = 44100\ \mathrm{Hz}$ (Sampling Frequency)

2) หาค่า Ω_0 จากความสัมพันธ์ $\Omega_0 = \tan\left(\pi \frac{\mathrm{f}}{\mathrm{F}}\right)$

3) ค่า A คือ เศษของสัมประสิทธิ์ (Numerator coefficient) ในสมการ $\mathrm{H(s)}$ ในสมการที่ (1), (2) และ (3) ค่า B คือ ส่วนของสัมประสิทธิ์ (Denominator coefficient) ในสมการ $\mathrm{H(s)}$ ในสมการที่ (1), (2) และ (3)

4) P คือ ปาสคาลเมตริกซ์ (Pascal Matrix)

5) นำค่า A คูณกับ Pascal Matrix และนำค่า B คูณกับ Pascal Matrix

6) ตัวแปร num มาจาก Numerator คือค่าเศษสัมประสิทธิ์ในสมการ $\mathrm{H(z)}$ ที่ต้องหาค่า และ den มาจาก Denominator คือค่าส่วนสัมประสิทธิ์ในสมการ $\mathrm{H(z)}$ ที่จะต้องหาค่า วิธีการคือจะนำค่าสัมประสิทธิ์ของส่วนที่อยู่หน้าตัวกำลังสูงสุดหารทั้งสมการซึ่งคือค่า $\mathrm{b(0)}$ จาก Matrix b นำค่าที่ได้หารค่า a ทั้งหมดและหารค่า b ทั้งหมด

7) ค่า h1 คือค่า Frequency response มีค่าเป็น Complex number ค่า q1 คือค่า angular frequency vector ซึ่งมีค่า 0 - 9990 จะใช้ function $\mathrm{signal.freqz()}$ โดยมี 4 ตัวแปรคือ num คือค่าในข้อที่ 6, den คือค่าในข้อที่ 6 , worN คือค่าที่ต้องกำหนดเพราะเนื่องจากวงจรกรองความถี่ที่สร้างเป็น IIR filter design ซึ่งมีค่าไม่จำกัดจึงต้องกำหนดค่าที่ 1000 จุด, $\mathrm{f_s}$ คือค่าจากข้อที่ 1

ในส่วนของ function fminsearch ในไพธอนจะใช้ทฤษฎี Nelder Mead เพื่อควบคุมการสุ่มให้ได้ค่า Error ที่ต้องการ เห็นดังรูปที่ 3.89

```
banana = lambda x: ErrorFunction(x[0],x[1],x[2],x[3],x[4],x[5],x[6],x[7],x[8], Data[i])
opt = scipy.optimize.fmin(func=banana, x0=[1599,0.299,9.5,4000,0.899,10,8000,0.5,10],xtol=1e-4,ftol=1e-4,full_output=True)
```

รูปที่ 3.89 การกำหนดค่าเริ่มต้นสำหรับคำสั่ง fminsearch

จากรูปที่ 3.89 โดยมีการกำหนดค่า Initial เป็นค่าประมาณที่เหมาะสมที่สุดในการสร้างฟังก์ชันที่ใช้สำหรับคำนวณค่าความผิดพลาดในไพธอน

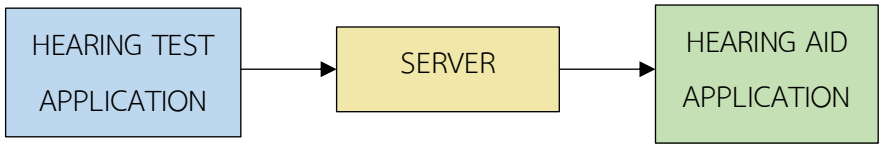
```
def ErrorFunction(fL,QL,GL,fB,QB,GB,fH,QH,GH, newval2):
    q,h_lpf,num_lpf,den_lpf = BiquadLPF(fL,QL)
    HL = h_lpf*GL
    q,h_bpf,num_bpf,den_bpf = BiquadBPF(fB,QB)
    HB = h_bpf*GB
    q,h_hpf,num_hpf,den_hpf = BiquadHPF(fH,QH)
    HH = h_hpf*GH
    desired_flt = abs(HL+HB+HH)
    h_desired = 20*np.log10(desired_flt)
    octave_freq = [250,500,1000,2000,4000,8000]
    threshold_dB = newval2
    spc = np.linspace(250,8000,1000)
    h_ideal = np.interp(spc,octave_freq,threshold_dB)
    err = mean_absolute_error(h_ideal,h_desired)
    return err
```

รูปที่ 3.90 ฟังก์ชันสำหรับคำนวณค่าความผิดพลาด

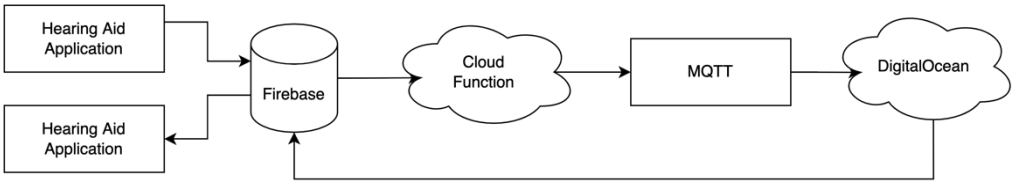
จากรูปที่ 3.90 การคำนวณค่าความผิดพลาดที่เลือกมาใช้คือ Mean Absolute Error โดยจะใช้ตัวต้นแบบการสูญเสียการได้ยินและวงจรกรองสัญญาณความถี่ที่ได้ออกแบบมาคำนวณหาค่าความผิดพลาด

3.1.7 การออกแบบเซิร์ฟเวอร์สำหรับคำนวณค่าสัมประสิทธิ์

ในการออกแบบวงจรกรองความถี่แบบดิจิทัลจะถูกติดตั้งให้อยู่บนเซิร์ฟเวอร์เนื่องจากปริญญานิพนธ์มีจุดประสงค์ให้การออกแบบกรองความถี่เป็นเหมือนกับการบริการอย่างนึง เพื่อให้ผู้ใช้สามารถนำข้อมูลจากศูนย์ตรวจสมรรถภาพการได้ยินมาออกแบบวงจรกรองความถี่แบบดิจิทัลเพื่อนำไปสร้างเครื่องช่วยฟังในแอปพลิเคชันต่อไป โดยในการออกแบบเซิร์ฟเวอร์จะมีการความสัมพันธ์แสดงดังรูปที่ 3.91



รูปที่ 3.91 บล็อกไดอะแกรมการเชื่อมต่อระหว่างแอปพลิเคชันทดสอบการได้ยิน เซิร์ฟเวอร์ และแอปพลิเคชันเครื่องช่วยฟัง



รูปที่ 3.92 บล็อคไดอะแกรมการทำงานของเซิร์ฟเวอร์

ในการเชื่อมต่อของเซิร์ฟเวอร์จะมีบริการของ Google Cloud Platform อยู่ 3 บริการคือ Firebase, Cloud function และ DigitalOcean และบริการ MQTT เพื่อใช้ในการบริหารจัดการการเข้าออกของข้อมูล โดยเซิร์ฟเวอร์จะทำงานก็ต่อเมื่อผู้ใช้งานมีผลการทดสอบการได้ยินส่งมาที่ฐานข้อมูลจึงจะเริ่มทำงานการออกแบบวงจรกรองความถี่ ซึ่งจะมีการทำงานเป็นลำดับดังนี้

3.1.7.1 Cloud Function

บริการ Cloud Function เป็นบริการของ Google Cloud Platform เพื่อใช้ในการตรวจจับข้อมูลเข้าออก เนื่องจากปริญญานิพนธ์ต้องการให้เป็นระบบการทำงานที่ไม่มีการสั่งการจากทางผู้ใช้ จึงจะเป็นการทำงานโดยอัตโนมัติโดย Cloud Function จะทำหน้าที่ตรวจจับข้อมูลเมื่อผู้ใช้ส่งผลการทดสอบสมรรถภาพการได้ยินไปยังฐานข้อมูลผ่านบริการ Firebase จากนั้นบริการ Cloud Function จะทำการตรวจจับและดึงข้อมูลที่ถูกสร้างขึ้นมาใหม่ส่งไปให้บริการ MQTT เพื่อนำไปออกแบบวงจรกรองความถี่ต่อไป ซึ่งมีโปรแกรมของ Cloud Function ในการสั่งการการตรวจจับแสดงดังรูปที่ 3.93

```
exports.createUserAudioGram = functions
  .region("asia-southeast1")
  .firestore.document("users/{userId}/audiogram_history/{audiogram_historyId}")
  .onCreate((snapshot, __) => {
    functions.logger.info(snapshot.data());

    let trigg_message = {
      userID: __.params.userId,
      audiogramID: __.params.audiogram_historyId,
      lValue: snapshot.data().Left,
      rValue: snapshot.data().Right,
    };

    client.publish("/audiogram", JSON.stringify(trigg_message), function (err) {
      if (err) {
        functions.logger.error("Error:" + err);
        response.send("Error:" + err);
        reject();
      }
    });

    return Promise.resolve();
  });

exports.updateUserAudioGram = functions
  .region("asia-southeast1")
  .firestore.document("users/{userId}/audiogram_history/{audiogram_historyId}")
  .onUpdate((snapshot, __) => {
    functions.logger.info(snapshot.before.data());
    functions.logger.info(snapshot.after.data());
    return Promise.resolve();
  });
```

รูปที่ 3.93 การเขียนโปรแกรมในส่วนของ Cloud Function

โดยการตรวจจับจะตรวจจับก็ต่อเมื่อมีการสร้างข้อมูลใหม่นั่นคือผลการทดสอบสมรรถภาพการได้ยิน จะดึงข้อมูลเก็บไว้ในตัวแปรชื่อ ‘lvalue’ และ ‘rvalue’ และนำตัวแปรทั้ง 2 ส่งไปให้ MQTT เพื่อให้บริหารจัดการการส่งข้อมูลต่อไป

3.1.7.2 MQTT

บริการ MQTT เป็นบริการที่ใช้ในการบริหารจัดการการส่งข้อมูลไม่ให้มีความติดขัดของการส่งข้อมูล โดยเนื่องจากส่วนของโปรแกรมในการออกแบบวงจรกรองความถี่จะมีการประมวลผลที่ใช้เวลานานจึงทำให้เกิดการติดขัดของข้อมูลได้ จึงใช้บริการ MQTT ในการจัดการข้อมูลให้สามารถส่งข้อมูลได้ สะดวกมากยิ่งขึ้นโดยสามารถเขียนโปรแกรมให้อยู่ในส่วนเดียวกันกับ Cloud Function เพื่อให้สามารถเชื่อมต่อกันได้

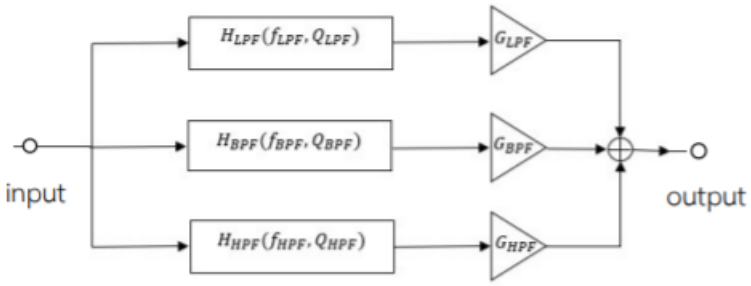
```
const functions = require("firebase-functions");
const admin = require("firebase-admin");
const mqtt = require("mqtt");

admin.initializeApp();
      You, a week ago • 25/11/2021
const client = mqtt.connect('mqtt://ggaomyqh:3wjA27NFU3ET@m16.cloudmqtt.com:16319/');

client.on("connect", function () {
  functions.logger.info("Ready To Rock!!");
});

client.on("reconnect", () => {
  functions.logger.info("MQTT client is reconnecting...");
});

client.on("disconnect", () => {
  functions.logger.info("MQTT client is disconnecting...");
});

client.on("error", function () {
  functions.logger.error("Can't connect");
  client.reconnect();
});
```

รูปที่ 3.94 ส่วนของโปรแกรมการทำงานของ MQTT

เมื่อมีการส่งข้อมูลจาก Cloud Function มาที่ MQTT ต่อไปจะนำข้อมูลที่ได้นำส่งไปที่ Cloud Run เพื่อออกแบบวงจรกรองความถี่ต่อไป

3.1.7.3 DigitalOcean

บริการ DigitalOcean ถูกใช้ในการเป็นคลาวด์เซิร์ฟเวอร์เพื่อใช้สำหรับการออกแบบวงจรกรองความถี่แบบดิจิทัล จากข้อมูลที่ถูกส่งมาจากบริการ MQTT โดยโปรแกรมการออกแบบจะเป็นการออกแบบด้วยสมการไบควอดเดรติก เพื่อออกแบบวงจรกรองความถี่ 3 แบบ คือ วงจรกรองความถี่ต่ำ วงจรกรองความถี่สูง และวงจรกรองความถี่แถบผ่าน เพื่อใช้ในการออกแบบวงจรกรองความถี่เพื่อชดเชยการสูญเสียการได้ยิน



รูปที่ 3.95 ระบบการทำงานของวงจรกรองความถี่แบบดิจิทัล

ในการออกแบบวงจรกรองความถี่จะใช้วิธี Optimization ในการหาวงจรกรองความถี่แบบดิจิทัลที่เหมาะสมที่สุด โดยใช้การคำนวณหาค่าความผิดพลาดด้วยสมการ Mean absolute error เป็นเกณฑ์ในการหาค่าความผิดพลาดต่ำสุด ตามสมการที่ 3.3 เป็นวงจรกรองความถี่ที่เหมาะสมที่สุด

```
def BiquadLPF(flpf,Qlpf):
    f_samp = 20000
    ohm0 = np.tan(np.pi*flpf/f_samp)
    A1 = [(ohm0)**2, 0, 0]
    B1 = [(ohm0)**2, ohm0/Qlpf, 1]
    P = [[1,1,1],
        [2,0,-2],
        [1,-1,1]]
    a1 = np.dot(P,np.transpose(A1))
    b1 = np.dot(P,np.transpose(B1))
    num= a1/b1[0]
    den = b1/b1[0]
    q1,h1 = signal.freqz(num,den,worN=1000,fs=44100)
    return q1,h1,num,den

def BiquadBPF(fbpf,Qbpf):
    f_samp = 20000
    ohm0 = np.tan(np.pi*fbpf/f_samp)
    A1 = [0, ohm0/Qbpf, 0]
    B1 = [(ohm0)**2, ohm0/Qbpf, 1]
    P = [[1,1,1],
        [2,0,-2],
        [1,-1,1]]
    a1 = np.dot(P,np.transpose(A1))
    b1 = np.dot(P,np.transpose(B1))
    num= a1/b1[0]
    den = b1/b1[0]
    q1,h1 = signal.freqz(num,den,worN=1000,fs=20000)
    q1,h1 = signal.freqz(num,den,worN=1000,fs=44100)

def BiquadHPF(fhpf,Qhpf):
    f_samp = 20000
    ohm0 = np.tan(np.pi*fhpf/f_samp)
    A1 = [0, 0, 1]
    B1 = [(ohm0)**2, ohm0/Qhpf, 1]
    P = [[1,1,1],
        [2,0,-2],
        [1,-1,1]]
    a1 = np.dot(P,np.transpose(A1))
    b1 = np.dot(P,np.transpose(B1))
    num= a1/b1[0]
    den = b1/b1[0]
    q1,h1 = signal.freqz(num,den,worN=1000,fs=20000)
    q1,h1 = signal.freqz(num,den,worN=1000,fs=44100)
```

รูปที่ 3.96 โปรแกรมการออกแบบวงจรกรองความถี่ด้วยสมการ Biquadratic

```
banana = lambda x: ErrorFunction(x[0],x[1],x[2],x[3],x[4],x[5],x[6],x[7],x[8], Data[i])
opt = scipy.optimize.fmin(func=banana, x0=[1599,0.299,9.5,4000,0.899,10,8000,0.5,10],xtol=1e-4,ftol=1e-4,full_output=True)
print('Current function value: ',opt[1])
print('Iterations: ',opt[2])
print('Function evaluations: ',opt[3])

xopt = opt[0]
print('Coefficients: ',xopt);
flpf = abs(xopt[0])
Qlpf = abs(xopt[1])
Glpf = abs(xopt[2])
fbpf = abs(xopt[3])
Qbpf = abs(xopt[4])
Gbpf = abs(xopt[5])
fhpf = abs(xopt[6])
Qhpf = abs(xopt[7])
Ghpf = abs(xopt[8])
q,h_lpf,num_lpf,den_lpf = BiquadLPF(flpf,Qlpf)

q,h_bpf,num_bpf,den_bpf = BiquadBPF(fbpf,Qbpf)

q,h_hpf,num_hpf,den_hpf  = BiquadHPF(fhpf,Qhpf)

#sending patient's data back to firebase
UTC = pytz.utc
Lcoeff.append((num_lpf+den_lpf+Glpf).tolist())
Bcoeff.append((num_bpf+den_bpf+Gbpf).tolist())
Hcoeff.append((num_hpf+den_hpf+Ghpf).tolist())
```

รูปที่ 3.97 การเขียนโปรแกรมออกแบบวงจรกรองความถี่โดยอาศัยหลักการ Optimization

```
def ErrorFunction(fL,QL,GL,fB,QB,GB,fH,QH,GH, newval2):
    q,h_lpf,num_lpf,den_lpf = BiquadLPF(fL,QL)
    HL = h_lpf*GL
    q,h_bpf,num_bpf,den_bpf = BiquadBPF(fB,QB)
    HB = h_bpf*GB
    q,h_hpf,num_hpf,den_hpf = BiquadHPF(fH,QH)
    HH = h_hpf*GH
    desired_flt = abs(HL+HB+HH)
    h_desired = 20*np.log10(desired_flt)
    octave_freq = [250,500,1000,2000,4000,8000]
    threshold_dB = newval2
    spc = np.linspace(250,8000,1000)
    h_ideal = np.interp(spc,octave_freq,threshold_dB)
    err = mean_absolute_error(h_ideal,h_desired)
    return err
```

รูปที่ 3.98 การเขียนโปรแกรมโปรแกรมคำนวณค่าความผิดพลาดด้วยสมการ Mean absolute error

โดยระบบจะรับค่าผลการทดสอบสมรรถภาพการได้ยินจากบริการ MQTT เพื่อนำมาออกแบบวงจรกรองความถี่แบบดิจิทัล และคำนวณหาค่าสัมประสิทธิ์เพื่อส่งกลับไปที่ฐานข้อมูลที่ Firebase และนำไปใช้สำหรับการชดเชยการสูญเสียการได้ยินในแอปพลิเคชันเครื่องช่วยฟัง

```
cred = credentials.Certificate("hearing-test-2e94b-firebase-adminsdk-6qp9b-5a2383d7ad.json")
firebase_admin.initialize_app(cred)
db = firestore.client()

host = os.getenv('CLOUDMQTT_URL')
url = urlparse(host)
topic = url.path[1:]

def on_connect(self, client, userdata, rc):
    logging.info('connected')
    self.subscribe("/"+topic, 0)


def on_message(client, userdata, msg):
    logging.debug(msg.payload.decode("utf-8", "strict"))
    m_in = json.loads(msg.payload.decode("utf-8", "strict"))
    user_id = get(m_in, 'userID')
    a_id = get(m_in, 'audiogramID')
    leftValue = get(m_in, 'lValue')
    rightValue = get(m_in, 'rValue')

    Data = [leftValue,rightValue]
    Lcoeff=[]
    Bcoeff=[]
    Hcoeff=[]
```

รูปที่ 3.99 การเขียนโปรแกรมที่รับค่ามาจากบริการ MQTT

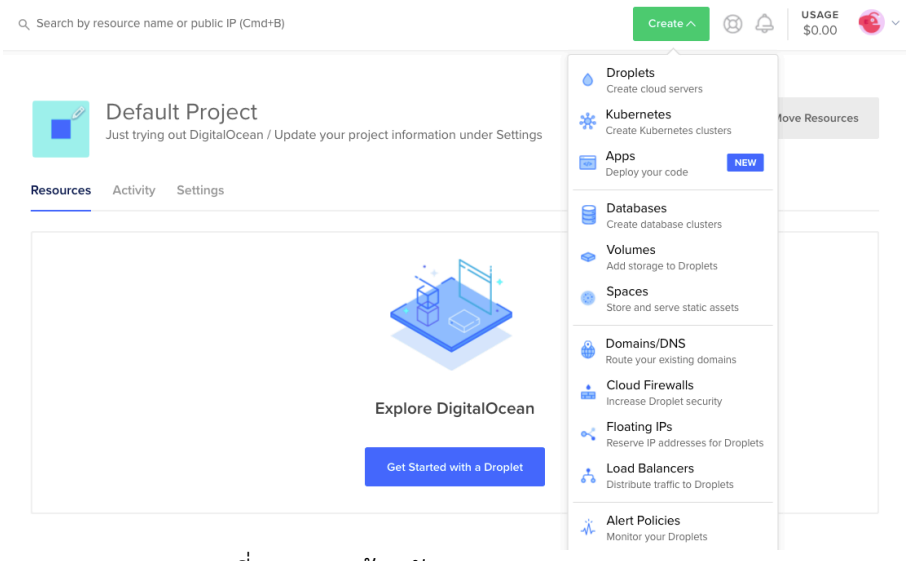
```
doc_result = db.collection(u'users').document(user_id).collection(u'audiogram_history').document(a_id)
doc_result.update({
    u'LPFcoeff_Left': Lcoeff[0], # array
    u'BPFcoeff_Left': Bcoeff[0],
    u'HPFcoeff_Left':  Hcoeff[0],
    u'LPFcoeff_Right': Lcoeff[1],
    u'BPFcoeff_Right': Bcoeff[1],
    u'HPFcoeff_Right':  Hcoeff[1],
    u'updated_at': datetime.now(UTC),
})
```

รูปที่ 3.100 การเขียนโปรแกรมส่งค่าสัมประสิทธิ์ไปที่ฐานข้อมูล

- วิธีการสร้างคลาวด์เซิร์ฟเวอร์บน DigitalOcean

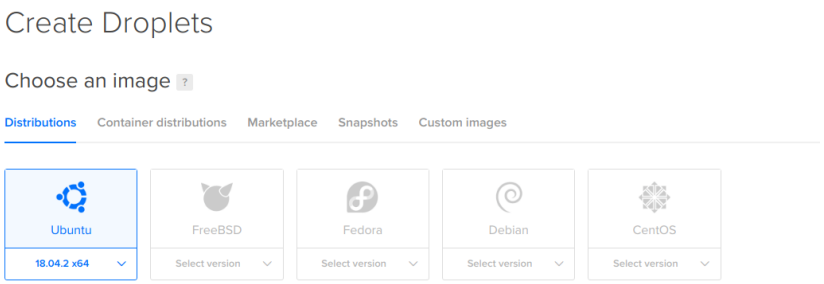
สำหรับบริการที่เลือกใช้สำหรับการสร้างคลาวด์เซิร์ฟเวอร์คือ บริการ Droplet โดยการสร้าง Droplet นั้นจะใช้คำสั่งของระบบปฏิบัติการลีนุ๊กซ์ (Linux) โดยมีขั้นตอนในการสร้างดังต่อไปนี้

1) เมื่อทำการเข้าสู่ระบบผู้ใช้งาน และเข้าสู่หน้าหลักของ DigitalOcean กดปุ่ม ‘Create’ ตามรูปที่ 3.101 และเลือกบริการ Droplets



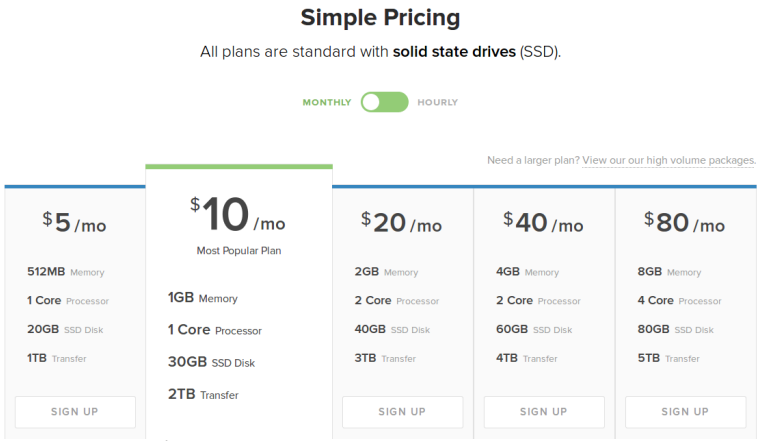
รูปที่ 3.101 หน้าหลักของ DigitalOcean [34]

2) เลือก Image ที่ต้องการใช้งานบน Droplet โดยในปริญญานิพนธ์นี้เลือกใช้ Image ในรูปแบบ Distribution แบบ Ubuntu แสดงดังรูปที่ 3.103



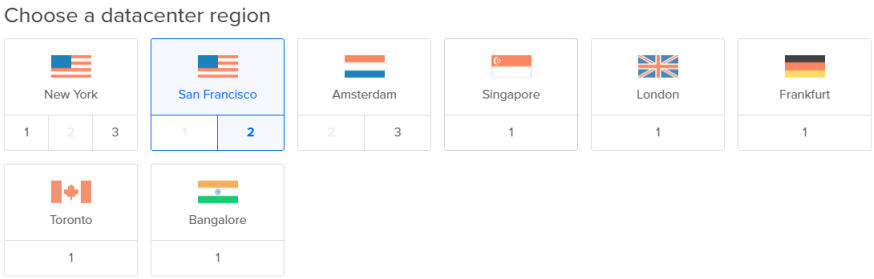
รูปที่ 3.102 หน้าแสดง Image ที่ใช้งานได้บน DigitalOcean [34]

3) เลือกค่าบริการที่ต้องการใช้ตามการใช้งาน โดยในที่นี้เลือกใช้บริการพื้นฐานซึ่งมีหน่วยความจำ 512 MB โปรเซสเซอร์ 1 คอร์ หน่วยความจำ SSD 20 GB โดยมีสิทธิให้ทดลองใช้ฟรี 2 เดือน



รูปที่ 3.103 ค่าบริการของ DigitalOcean [35]

3) เลือก Datacenter ที่ต้องการใช้งาน โดยในที่นี้เลือกใช้ Datacenter ของประเทศสิงคโปร์



รูปที่ 3.104 Datacenter ที่เปิดให้บริการ [34]

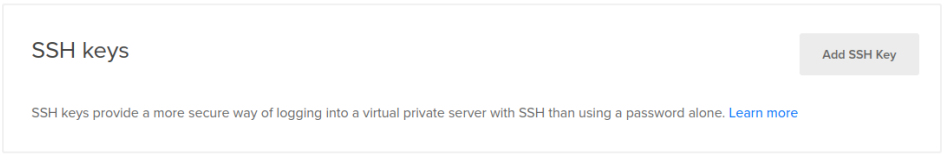
4) ทำการยืนยันตัวตน (Authentication) โดยเลือกใช้ SSH keys โดยสามารถสร้างได้จาก PowerShell บนเครื่องคอมพิวเตอร์ซึ่งมีขั้นตอนดังนี้

1. พิมพ์ ssh-key gen บน PowerShell
2. ระบบจะขึ้นข้อความสำหรับการยืนยันให้ทำการกด enter จนระบบแสดงข้อความดังรูปที่ 3.106

```
Your identification has been saved in /home/username/.ssh/id_rsa.
Your public key has been saved in /home/username/.ssh/id_rsa.pub.
The key fingerprint is:
a9:49:EX:AM:PL:E3:3e:a9:de:4e:77:11:58:b6:90:26 username@203.0.113.0
The key's randomart image is:
+--[ RSA 2048]----+
|     ..o         |
|   E o= .        |
|    o. o         |
|        ..       |
|      ..S        |
|     o o.        |
|   =o.+.         |
|. =++..          |
|o=++.            |
+-----------------+
```

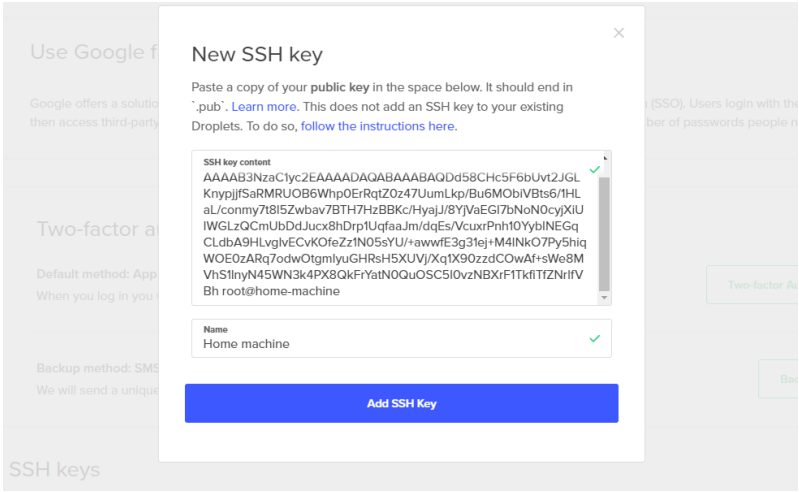
รูปที่ 3.105 ตัวอย่าง SSH key [36]

3. ทำการอัพโหลด SSH key ที่ได้ โดยกดเข้าไปที่ Add SSH key แสดงดังรูปที่ 3.106



รูปที่ 3.106 หน้าที่ใช้สำหรับการเพิ่ม SSH key [36]

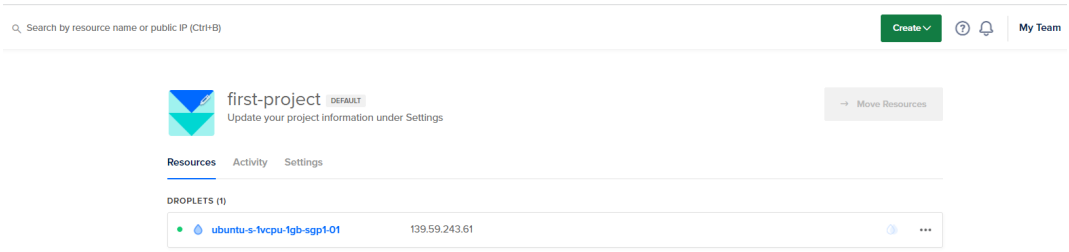
4. นำ SSH key ที่ได้วางลงบนกล่องข้อความแสดงดังรูปที่ 3.108



รูปที่ 3.107 ตัวอย่างการเพิ่ม SSH key [36]

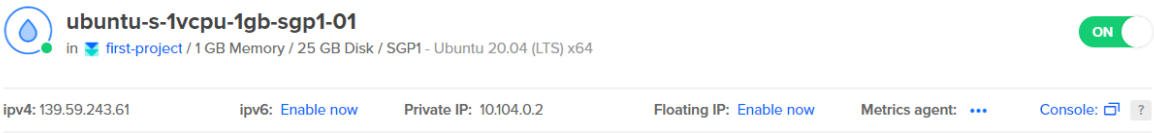
5. เมื่อทำการสร้าง SSH key เสร็จให้ทำการพิมพ์ cat ~/.ssh/id_rsa.pub ที่ PowerShell เพื่อสร้าง key ที่จะใช้เชื่อมต่อกับ GitHub สำหรับการอัปโหลดโค้ดที่ต้องการใช้ประมวลผล

6. การสร้าง Droplet เสร็จเป็นที่เรียบร้อย



รูปที่ 3.108 Droplet ที่ใช้สำหรับประมวลผลบนคลาวด์เซิร์ฟเวอร์ [36]

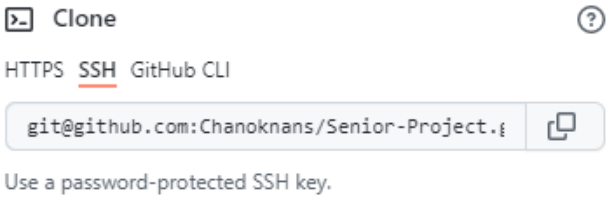
7. กดเข้าไปที่ Droplet ที่ได้ทำการสร้างจากนั้นกดที่ Console เพื่อทำการอัปโหลดโค้ดที่ต้องการใช้งาน



รูปที่ 3.109 หน้าหลักของ Droplet ที่ใช้สำหรับประมวลผลคลาวด์เซิร์ฟเวอร์ [36]

8. ทำการติดตั้ง git และ Python บน Ubuntu ของคลาวด์เซิร์ฟเวอร์ให้เรียบร้อย และทำการเข้าสู่ระบบของ git ผ่านคำสั่ง git config —global user.name “/*your username*/” และ git config —global user.email “/*your email*/”

9. ทำการโคลนโปรเจคของโค้ดที่ได้ทำการอัปโหลดไว้บน git ในรูปของ SSH link แสดงดังรูปที่ 3.110 และใช้คำสั่ง git clone “/*ssh link*/”



รูปที่ 3.110 SSH link บน GitHub

5) สามารถใช้บริการคลาวด์เซิร์ฟเวอร์ด้วยบริการของ DigitalOcean เป็นที่เรียบร้อย

- การใช้งานโปรแกรมบนคลาวด์เซิร์ฟเวอร์

เมื่อทำการสร้าง Droplet ผ่านบริการ DigitalOcean เสร็จเป็นที่เรียบร้อย สามารถใช้งานโปรแกรมสำหรับคำนวณค่าสัมประสิทธิ์ที่ใช้สำหรับออกแบบวงจรกรองความถี่แบบดิจิทัลบนคลาวด์เซิร์ฟเวอร์ได้ โดยวิธีใช้งานแสดงขั้นตอนดังต่อไปนี้

1) เมื่อเข้าสู่หน้าคอนโซลของคลาวด์เซิร์ฟเวอร์ให้ทำการพิมพ์คำสั่งเพื่อแสดงไปยัง path ที่ใช้เก็บโค้ดที่ใช้สำหรับการประมวลผลโดยพิมพ์คำสั่ง cd Senior-Project/api

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~# cd Senior-Project/api
root@ubuntu-s-1vcpu-1gb-sgp1-01:~/Senior-Project/api#
```

รูปที่ 3.111 คำสั่งที่ใช้สำหรับแสดง path ที่ใช้เก็บข้อมูล

2) ตรวจสอบไฟล์ที่ถูกจัดเก็บอยู่ว่ามีโค้ดโปรแกรมที่ใช้สำหรับประมวลผลหรือไม่ โดยในที่นี้โค้ดที่ใช้สำหรับคำนวณค่าสัมประสิทธิ์ที่ใช้สำหรับออกแบบวงจรกรองความถี่แบบดิจิทัลคือ ไฟล์ filtering.py ไฟล์requirement.txt และไฟล์ hearing-test-2e94b-firebase-adminsdk-6qp9b-5a2383d7ad.json แสดงดังรูปที่ 3.112

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~/Senior-Project/api# ls
Dockerfile  filtering.py  hearing-test-2e94b-firebase-adminsdk-6qp9b-5a2383d7ad.json
requirements.txt
```

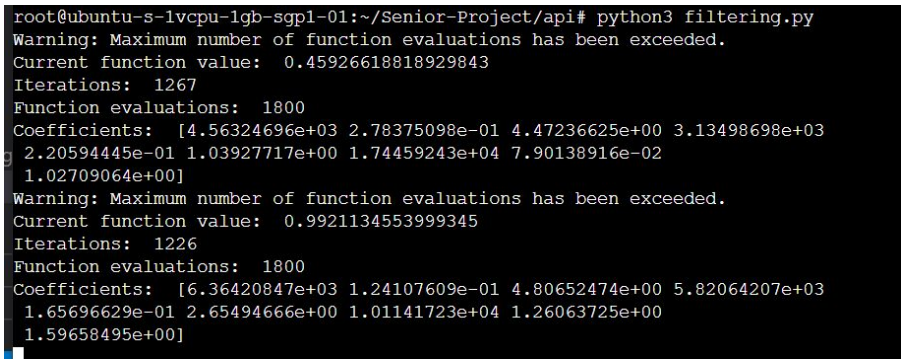
รูปที่ 3.112 ไฟล์ที่ถูกจัดเก็บอยู่ในโฟลเดอร์ api

3) พิมพ์คำสั่ง python3 filtering.py เพื่อเริ่มใช้งานคลาวด์เซิร์ฟเวอร์

```
root@ubuntu-s-1vcpu-1gb-sgp1-01:~/Senior-Project/api# python3 filtering.py
```

รูปที่ 3.113 คำสั่งที่ใช้สำหรับรันโปรแกรม

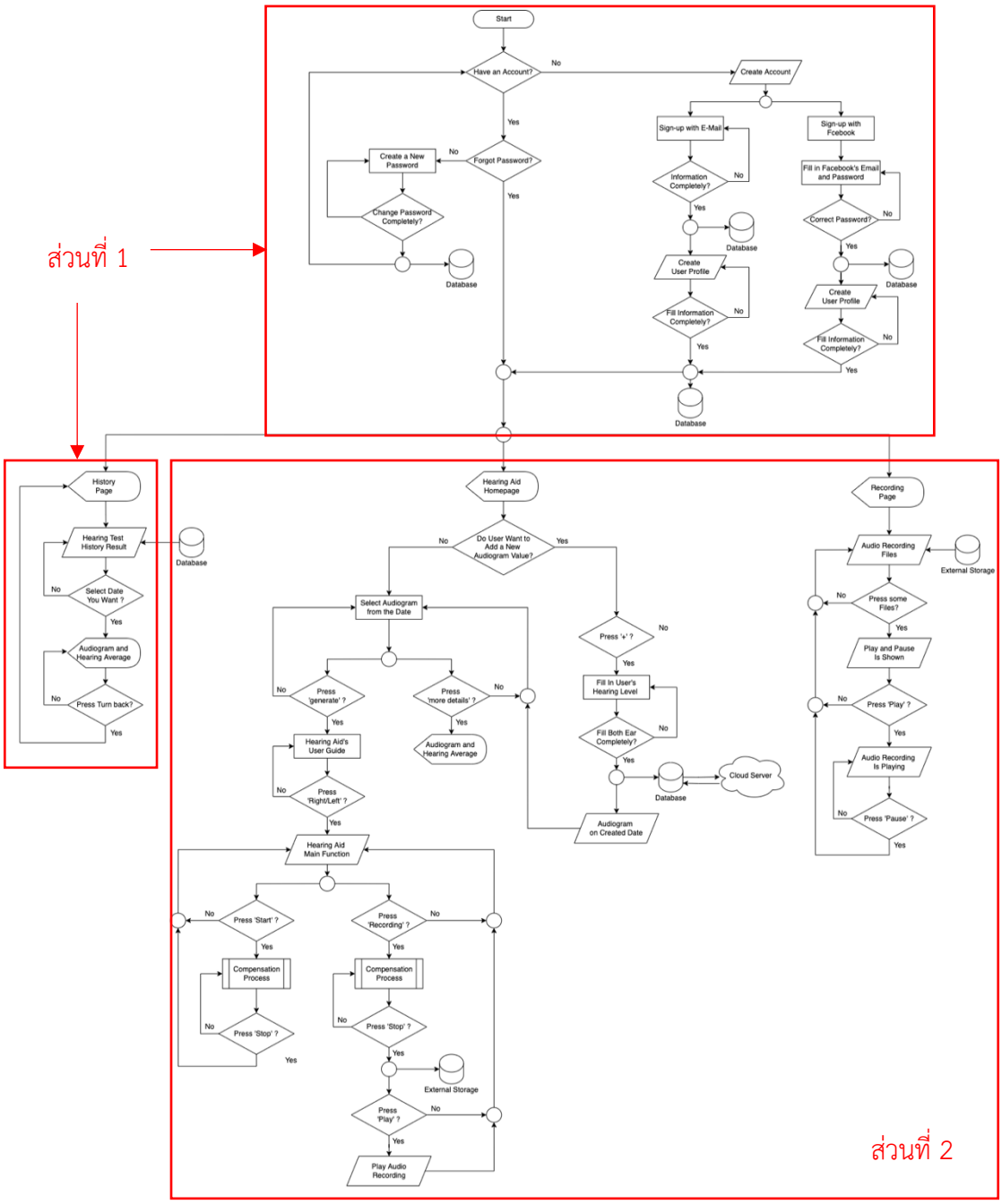
4) เมื่อมีการส่งผลทดสอบค่าสมรรถภาพการได้ยินจากผู้ใช้งานแอปพลิเคชันเครื่องช่วยฟัง ระบบจะแสดงค่าที่ได้จากการคำนวณค่าสัมประสิทธิ์ที่ใช้สำหรับออกแบบวงจรกรองความถี่แบบดิจิทัลบนเซิร์ฟเวอร์และส่งค่าที่ได้จากการคำนวณกลับไปยังผู้ใช้งานเพื่อนำไปใช้ต่อในกระบวนการถัดไป



รูปที่ 3.114 ตัวอย่างค่าที่ได้จากการคำนวณค่าสัมประสิทธิ์บนคลาวด์เซิร์ฟเวอร์

3.1.8 การออกแบบระบบการทำงานของแอปพลิเคชันเครื่องช่วยฟัง

จากการศึกษาระบบการทำงานของเครื่องช่วยฟัง ปริญญานิพนธ์นี้ได้ทำการออกแบบระบบและวิธีการชดเชยการสูญเสียการได้ยินบนโมบายแอปพลิเคชันชันชันโดยใช้ภาษา Flutter ในการเขียนแอปพลิเคชัน และใช้ Firebase เป็นฐานข้อมูลสำหรับเก็บค่าการทดสอบสมรรถภาพการได้ยินจากแอปพลิเคชันการทดสอบการได้ยิน และประวัติของผู้ใช้งาน โดยภาพรวมของระบบการทำงานสำหรับแอปพลิเคชันเครื่องช่วยฟังเป็นไปตามรูปที่ 3.116

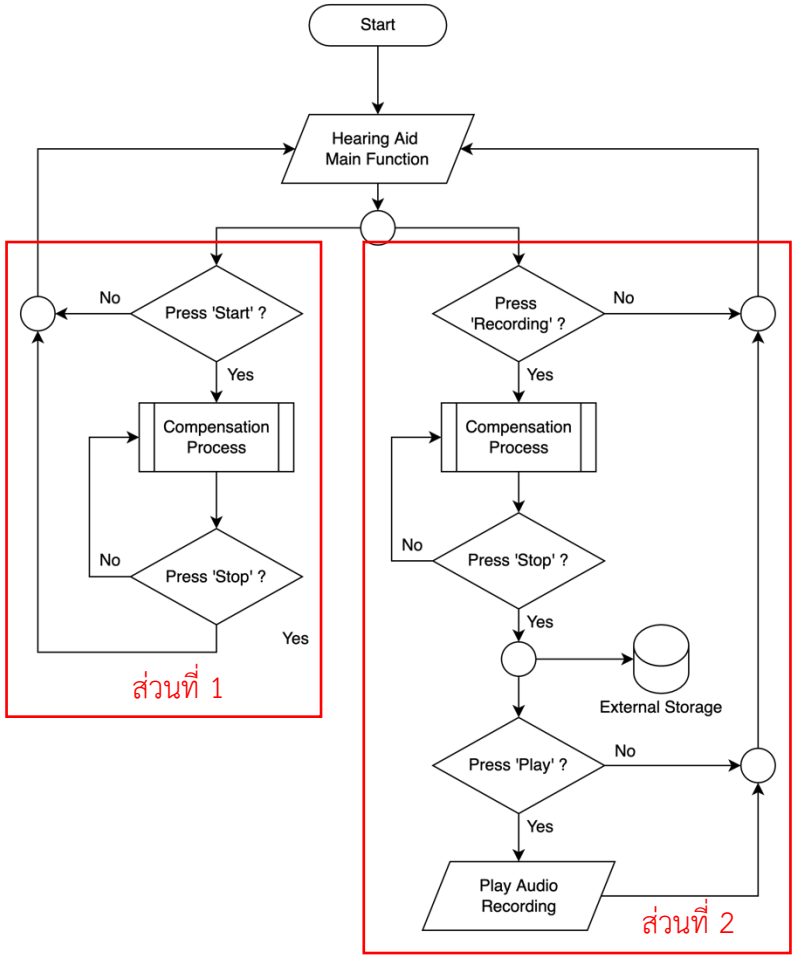


รูปที่ 3.115 แผนภาพระบบการทำงานแอปพลิเคชันเครื่องช่วยฟัง

จากรูปที่ 3.115 แสดงแผนภาพระบบการทำงานแอปพลิเคชันเครื่องช่วยฟัง (Hearing Aid Application) โดยการทำงานหลักของระบบแอปพลิเคชันเครื่องช่วยฟังสามารถแบ่งออกได้เป็น 2 ส่วนได้แก่ ส่วนที่ 1 คือ ส่วนที่ใช้จัดเก็บประวัติการทดสอบสมรรถภาพการได้ยินจากแอปพลิเคชันการทดสอบการได้ยิน และประวัติผู้ใช้งาน ส่วนที่ 2 คือ ส่วนการทำงานหลักของแอปพลิเคชันเครื่องช่วยฟังและส่วนที่ใช้สำหรับการแสดงผลการบันทึกเสียงที่ผ่านการชดเชยการสูญเสียการได้ยิน ซึ่งเป็นส่วนเสริมจากการทำงานหลักของแอปพลิเคชันเครื่องช่วยฟังในกรณีที่ผู้ใช้งานต้องการบันทึกเสียง ณ ขณะนั้น

3.1.8.1 การออกแบบระบบการทำงานหลักของแอปพลิเคเครื่องช่วยฟัง

ในส่วนระบบการทำงานหลักของแอปพลิเคชันเครื่องช่วยฟังสามารถแบ่งระบบการทำงานได้แสดงตามรูปที่ 3.116 ซึ่งจะมีการแบ่งฟังก์ชันการทำงานย่อยได้ออกเป็น 2 ส่วน คือ ส่วนที่ 1 ส่วนการทำงานของแอปพลิเคชันเครื่องช่วยฟังแบบเรียลไทม์ และส่วนที่ 2 ส่วนที่ใช้สำหรับการบันทึกเสียงที่ผ่านการชดเชยการสูญเสียการได้ยิน ซึ่งเป็นส่วนเสริมจากส่วนที่ 1 ในกรณีที่ผู้ใช้งานต้องการเก็บบันทึกเสียงที่ผ่านการชดเชย ณ ขณะนั้น



รูปที่ 3.116 แผนภาพระบบการทำงานหลักของแอปพลิเคชันเครื่องช่วยฟัง

จากรูปที่ 3.116 จะเห็นได้ว่าระบบการทำงานทั้งสองส่วนนั้นมีลักษณะการทำงานที่คล้ายกัน แต่แตกต่างกันที่ข้อมูลที่ใช้ในการประมวลผล และลักษณะการจัดเก็บข้อมูล โดยในส่วนที่ 1 จะมีการนำเสียงที่รับเข้าจากไมโครโฟนมาประมวลผล และจัดเก็บไฟล์เสียงแบบชั่วคราว (Temporary Storage) เพื่อให้การจัดการเสียงเป็นไปอย่างต่อเนื่องแสดงตามรูปที่ 3.117 (ก) และ 3.117 (ข) แตกต่างจากส่วนที่ 2 ที่มีการนำเสียงรับจากไมโครโฟนมาจัดเก็บ และจัดเก็บไฟล์เสียงบนพื้นที่จัดเก็บข้อมูลในโทรศัพท์ (External Storage) แสดงตามรูปที่ 3.118 (ก) และ 3.118 (ข) และนำไปใช้ต่อในหน้าแสดงผลไฟล์จัดเก็บบันทึกเสียงแสดงดังรูปที่ 3.119

```
void startPlayer() async {
  HomePageCubit homePageCubit = BlocProvider.of<HomePageCubit>(context);
  assert(_mRecorderIsInited && _mPlayer2!.isStopped || _mPlayer!.isPlaying);
  await _mPlayer!.startPlayerFromStream(
    codec: Codec.pcm16,
    numChannels: 1,
    sampleRate: tSampleRate,
  );
  var sink = await createFile2();
```

(ก)

```
Future<IOSink> createFile2() async {
  var tempDir = await getTemporaryDirectory();
  _mPath = '${tempDir.path}/${DateTime.now().toString()}.pcm';
  var outputFile = File(_mPath!);
  if (outputFile.existsSync()) {
    await outputFile.delete(); // sent file to local storage
  }
  return outputFile.openWrite();
}
```

(ข)

รูปที่ 3.117 ข้อมูลที่ใช้ในการประมวลผลและลักษณะการจัดเก็บข้อมูลของระบบการทำงานของแอปพลิเคชันเครื่องช่วยฟังแบบเรียลไทม์

(ก) คำสั่งในการเขียนโปรแกรมสำหรับการนำเสียงที่รับเข้าจากไมโครโฟน

(ข) คำสั่งในการเขียนโปรแกรมสำหรับการจัดเก็บไฟล์เสียงแบบชั่วคราว

```dart
Future<void> record() async {
  HomePageCubit homePageCubit = BlocProvider.of<HomePageCubit>(context);
  assert(_mRecorderIsInited && _mPlayer2!.isStopped || _mPlayer!.isPlaying);
  var sink = await createFile();
  var recordingDataController = StreamController<Food>();
  _mRecordingDataSubscription =
      recordingDataController.stream.listen((buffer) async {
```

(ก)

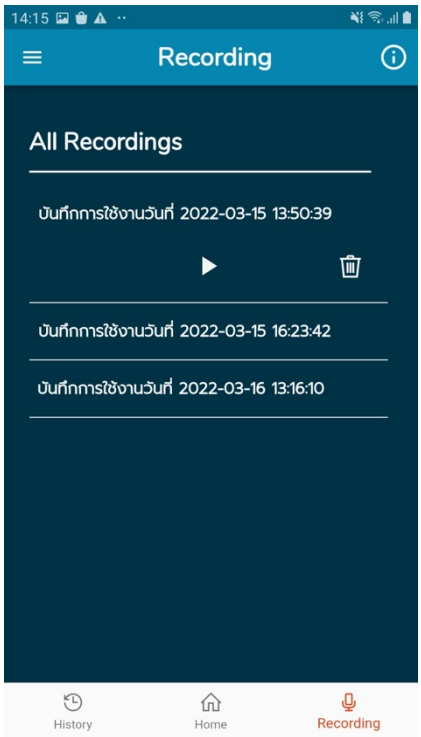
```dart
Future<IOSink> createFile() async {
  var tempDir = await getExternalStorageDirectory();
  var testdir =
      await Directory('${tempDir!.path}/sound').create(recursive: true);
  _mPath = '${testdir.path}/${DateTime.now().toString()}.pcm';
  print(_mPath);
  var outputFile = File(_mPath!);
  if (outputFile.existsSync()) {
    await outputFile.delete(); // sent file to local storage
  }
  return outputFile.openWrite();
}
```

(ข)

รูปที่ 3.118 ข้อมูลที่ใช้ในการประมวลผลและลักษณะการจัดเก็บข้อมูลของระบบการทำงานของการบันทึกเสียงที่ผ่านการชดเชยการสูญเสียการได้ยิน

(ก) คำสั่งในการเขียนโปรแกรมสำหรับการนำเสียงรับจากไมโครโฟนมาจัดเก็บ

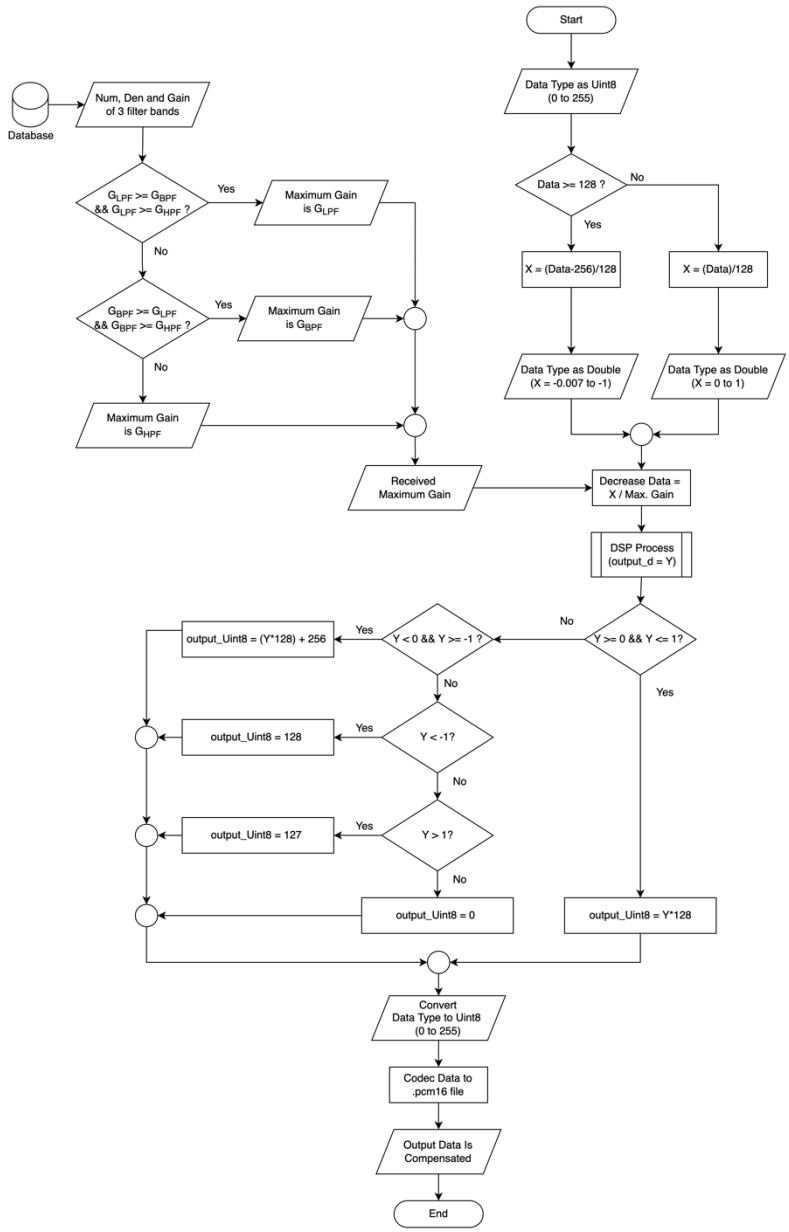
(ข) คำสั่งในการเขียนโปรแกรมสำหรับจัดเก็บไฟล์เสียงบนพื้นที่จัดเก็บข้อมูลในโทรศัพท์



รูปที่ 3.119 หน้าแสดงผลไฟล์จัดเก็บบันทึกเสียง

3.1.8.2 การออกแบบระบบการแปลงรูปแบบสัญญาณของข้อมูล

ในกระบวนการชดเชยการสูญเสียสมรรถภาพการได้ยินจะนำข้อมูลเสียงจากไมโครโฟนมาประมวลผลกับวงจรกรองความถี่แบบดิจิทัลที่ถูกออกแบบจากค่าการสูญเสียสมรรถภาพการได้ยิน โดยข้อมูลขาเข้าจะต้องเป็นข้อมูลในรูป Double ที่มีค่าอยู่ที่ช่วง -1 ถึง 1 เพื่อให้สอดคล้องกับการประมวลผลสัญญาณกับวงจรกรองความถี่แบบดิจิทัล เนื่องจากข้อมูลขาเข้าดังกล่าวจะถูกเก็บข้อมูลให้อยู่ในรูป Uint8 (Uint8: Unsigned Integer 8 bit) ที่มีค่าข้อมูลอยู่ในช่วง 0 ถึง 255 จึงต้องนำข้อมูลขาเข้านำมาแปลงข้อมูลให้กลับมาอยู่ในช่วง -1 ถึง 1 เพื่อนำไปใช้ในกระบวนการชดเชยการสูญเสียสมรรถภาพการได้ยิน โดยจะมีระบบการแปลงรูปแบบสัญญาณข้อมูลแสดงดังรูปที่ 3.120



รูปที่ 3.120 แผนภาพระบบการแปลงรูปแบบสัญญาณข้อมูล

จากข้อมูลที่มาจากไมโครโฟนจะมีค่าข้อมูลอยู่ในรูปแบบ Uint8 ที่มีค่าอยู่ในช่วง 0 ถึง 255 เนื่องจากโทรศัพท์จะมีการรับข้อมูลมาเป็นในรูปแบบ PCM16 ซึ่งจะมีข้อมูลต่อ 16 บิท ต่อ 1 ชุดข้อมูล โดยในระบบของการรับข้อมูลจะแบ่งออกเป็น 2 ประเภท คือ Mono และ Stereo จะมีรูปแบบข้อมูลดังรูปที่ 3.121

16-bit Mono	*LSB*	*MSB*	*LSB*	*MSB*
	Sample 1		Sample 2	

16-bit Stereo	***Left Channel***		***Right Channel***	
	LSB	*MSB*	*LSB*	*MSB*
	Sample 1		Sample 1	

รูปที่ 3.121 รูปแบบของข้อมูลของ PCM16 ในระบบ Mono และ Stereo [37]

โดยในการศึกษาได้ใช้รูปแบบ Mono เนื่องจากมีจำนวนข้อมูลในการประมวลผลน้อยกว่า และในการใช้งานจริง ผู้ใช้งานจำเป็นต้องใส่หูฟังเพียง 1 ข้างเท่านั้นเพื่อให้เกิดความปลอดภัยของผู้ใช้งาน จากรูปที่ 3.121 จะเห็นได้ว่าข้อมูลต่อ 1 ชุดข้อมูลจะประกอบไปด้วย LSB (Least Significant Bit) และ MSB (Most Significant Bit) ซึ่งจำเป็นที่ต้องแยกข้อมูลเป็น 2 ส่วนคือส่วนของข้อมูล LSB และส่วนของข้อมูล MSB เพื่อนำไปประมวลผลกับวงจรกรองความถี่แบบดิจิทัล เนื่องจากข้อมูลที่มาจากไมโครโฟนเป็นข้อมูลรูปแบบ Uint8 จึงต้องแปลงรูปแบบของข้อมูลให้เป็นรูปแบบ Double และทำการ Normalized เพื่อให้ข้อมูลมีค่าอยู่ในช่วง -1 ถึง 1 โดยใช้รูปแบบ 2's Complement โดยเนื่องจากข้อมูลเป็นข้อมูล 8 บิต ในการแปลงจะใช้หลักเกณฑ์ Q7 หรือการให้ข้อมูลบิทที่ 7 เป็นข้อมูลที่บ่งบอกเครื่องหมายของข้อมูล สามารถสังเกตได้จากตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างการแปลงข้อมูลด้วย 2's Complement

ข้อมูล 8 บิต	ข้อมูลจากไมโครโฟน	ข้อมูลที่ผ่าน 2's Complement
0000 0000	0	0
0000 0001	1	1
0111 1110	126	126
0111 1111	127	127
1000 0000	128	−128
1000 0001	129	−127
1111 1110	254	−2
1111 1111	255	−1

แสดงว่าข้อมูลที่ถูกแปลงแบบ 2’s Complement จะมีข้อมูลอยู่ในช่วง -128 ถึง 127 จากนั้นจะทำการ Normalized โดยการหารข้อมูลด้วย 128 เพื่อให้ข้อมูลมีค่าอยู่ในช่วง -1 ถึง 1 ซึ่งจะสามารถเขียนโปรแกรมการแปลงรูปแบบข้อมูลได้ดังรูปที่ 3.122

```
double reduce(int value) {
  double y;
  if (value < 128) {
    y = value / 128;
  } else {
    y = (value - 256) / 128;
  }
  return y;
}
```

รูปที่ 3.122 ฟังก์ชันการแปลงรูปแบบข้อมูลจาก Uint8 เป็น Double

โดยจะมีค่าที่ถูกแปลงจาก Uint8 เป็น Double ด้วย 2’s Complement และทำการ Normalized เพื่อแปลงค่าจากช่วง 0 ถึง 255 ให้เป็นค่าในช่วง -1 ถึง 1 ซึ่งจะแสดงดังตารางที่ 3.2

ตารางที่ 3.2 การแปลงข้อมูลจาก Uint8 เป็น Double

ข้อมูล Uint8	ข้อมูล Double	ข้อมูล Uint8	ข้อมูล Double	ข้อมูล Uint8	ข้อมูล Double
0	0	**33**	0.2578125	**66**	0.515625
1	0.0078125	**34**	0.265625	**67**	0.5234375
2	0.015625	**35**	0.2734375	**68**	0.53125
3	0.0234375	**36**	0.28125	**69**	0.5390625
4	0.03125	**37**	0.2890625	**70**	0.546875
5	0.0390625	**38**	0.296875	**71**	0.5546875
6	0.046875	**39**	0.3046875	**72**	0.5625
7	0.0546875	**40**	0.3125	**73**	0.5703125
8	0.0625	**41**	0.3203125	**74**	0.578125
9	0.0703125	**42**	0.328125	**75**	0.5859375
10	0.078125	**43**	0.3359375	**76**	0.59375
11	0.0859375	**44**	0.34375	**77**	0.6015625
12	0.09375	**45**	0.3515625	**78**	0.609375
13	0.1015625	**46**	0.359375	**79**	0.6171875
14	0.109375	**47**	0.3671875	**80**	0.625
15	0.1171875	**48**	0.375	**81**	0.6328125
16	0.125	**49**	0.3828125	**82**	0.640625
17	0.1328125	**50**	0.390625	**83**	0.6484375
18	0.140625	**51**	0.3984375	**84**	0.65625
19	0.1484375	**52**	0.40625	**85**	0.6640625
20	0.15625	**53**	0.4140625	**86**	0.671875
21	0.1640625	**54**	0.421875	**87**	0.6796875
22	0.171875	**55**	0.4296875	**88**	0.6875
23	0.1796875	**56**	0.4375	**89**	0.6953125
24	0.1875	**57**	0.4453125	**90**	0.703125
25	0.1953125	**58**	0.453125	**91**	0.7109375
26	0.203125	**59**	0.4609375	**92**	0.71875
27	0.2109375	**60**	0.46875	**93**	0.7265625
28	0.21875	**61**	0.4765625	**94**	0.734375
29	0.2265625	**62**	0.484375	**95**	0.7421875
30	0.234375	**63**	0.4921875	**96**	0.75
31	0.2421875	**64**	0.5	**97**	0.7578125
32	0.25	**65**	0.5078125	**98**	0.765625

ข้อมูล Uint8	ข้อมูล Double	ข้อมูล Uint8	ข้อมูล Double	ข้อมูล Uint8	ข้อมูล Double
99	0.7734375	**132**	-0.96875	**165**	-0.7109375
100	0.78125	**133**	-0.9609375	**166**	-0.703125
101	0.7890625	**134**	-0.953125	**167**	-0.6953125
102	0.796875	**135**	-0.9453125	**168**	-0.6875
103	0.8046875	**136**	-0.9375	**169**	-0.6796875
104	0.8125	**137**	-0.9296875	**170**	-0.671875
105	0.8203125	**138**	-0.921875	**171**	-0.6640625
106	0.828125	**139**	-0.9140625	**172**	-0.65625
107	0.8359375	**140**	-0.90625	**173**	-0.6484375
108	0.84375	**141**	-0.8984375	**174**	-0.640625
109	0.8515625	**142**	-0.890625	**175**	-0.6328125
110	0.859375	**143**	-0.8828125	**176**	-0.625
111	0.8671875	**144**	-0.875	**177**	-0.6171875
112	0.875	**145**	-0.8671875	**178**	-0.609375
113	0.8828125	**146**	-0.859375	**179**	-0.6015625
114	0.890625	**147**	-0.8515625	**180**	-0.59375
115	0.8984375	**148**	-0.84375	**181**	-0.5859375
116	0.90625	**149**	-0.8359375	**182**	-0.578125
117	0.9140625	**150**	-0.828125	**183**	-0.5703125
118	0.921875	**151**	-0.8203125	**184**	-0.5625
119	0.9296875	**152**	-0.8125	**185**	-0.5546875
120	0.9375	**153**	-0.8046875	**186**	-0.546875
121	0.9453125	**154**	-0.796875	**187**	-0.5390625
122	0.953125	**155**	-0.7890625	**188**	-0.53125
123	0.9609375	**156**	-0.78125	**189**	-0.5234375
124	0.96875	**157**	-0.7734375	**190**	-0.515625
125	0.9765625	**158**	-0.765625	**191**	-0.5078125
126	0.984375	**159**	-0.7578125	**192**	-0.5
127	0.9921875	**160**	-0.75	**193**	-0.4921875
128	1	**161**	-0.7421875	**194**	-0.484375
129	-0.9921875	**162**	-0.734375	**195**	-0.4765625
130	-0.984375	**163**	-0.7265625	**196**	-0.46875
131	-0.9765625	**164**	-0.71875	**197**	-0.4609375

ข้อมูล Uint8	ข้อมูล Double	ข้อมูล Uint8	ข้อมูล Double	ข้อมูล Uint8	ข้อมูล Double
198	-0.453125	**218**	-0.296875	**238**	-0.140625
199	-0.4453125	**219**	-0.2890625	**239**	-0.1328125
200	-0.4375	**220**	-0.28125	**240**	-0.125
201	-0.4296875	**221**	-0.2734375	**241**	-0.1171875
202	-0.421875	**222**	-0.265625	**242**	-0.109375
203	-0.4140625	**223**	-0.2578125	**243**	-0.1015625
204	-0.40625	**224**	-0.25	**244**	-0.09375
205	-0.3984375	**225**	-0.2421875	**245**	-0.0859375
206	-0.390625	**226**	-0.234375	**246**	-0.078125
207	-0.3828125	**227**	-0.2265625	**247**	-0.0703125
208	-0.375	**228**	-0.21875	**248**	-0.0625
209	-0.3671875	**229**	-0.2109375	**249**	-0.0546875
210	-0.359375	**230**	-0.203125	**250**	-0.046875
211	-0.3515625	**231**	-0.1953125	**251**	-0.0390625
212	-0.34375	**232**	-0.1875	**252**	-0.03125
213	-0.3359375	**233**	-0.1796875	**253**	-0.0234375
214	-0.328125	**234**	-0.171875	**254**	-0.015625
215	-0.3203125	**235**	-0.1640625	**255**	-0.0078125
216	-0.3125	**236**	-0.15625		
217	-0.3046875	**237**	-0.1484375		

จากตารางที่ 3.2 จะเห็นได้ว่าเมื่อแปลงข้อมูลให้อยู่ในรูป Double ซึ่งจะมีค่าอยู่ในช่วง -1 ถึง 1 ซึ่งสามารถนำไปใช้กับวงจรกรองความถี่แบบดิจิทัลที่ได้ออกแบบ โดยหลังจากนำข้อมูลผ่านวงจรกรองความถี่แบบดิจิทัลแล้ว ข้อมูลจะมีค่าที่อยู่ในช่วง -1 ถึง 1 โดยเป็นรูปแบบ Double แต่ในการทำให้ข้อมูลนั้นออกมาเป็นเสียงเพื่อให้ผู้ใช้งานได้ยินเสียงที่ถูกชดเชยการสูญเสียสมรรถภาพการได้ยิน จำเป็นต้องแปลงรูปแบบของข้อมูลกลับเป็น Uint8 ดังเดิม โดยจะทำการแปลงข้อมูลย้อนกลับโดยยังอิงการ Normalized และทฤษฎี 2’s Complement เพื่อให้ข้อมูลกลับมาอยู่ในช่วง 0 ถึง 255 ดังเดิม โดยสามารถเขียนโปรแกรมได้ดังรูปที่ 3.123

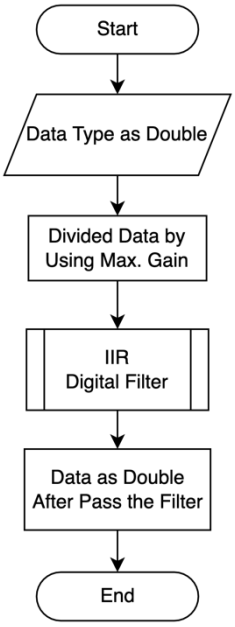
```
int scaling(num value) {
  int y;
  if (value >= 0 && value <= 1) {
    y = (value * 128).round();
  } else if (value < 0 && value >= -1) {
    y = ((value * 128) + 256).floor();
  } else if (value < -1) {
    y = 128;
  } else if (value > 1) {
    y = 127;
  } else {
    y = 0;
  }
  return y;
}
```

รูปที่ 3.123 ฟังก์ชันการแปลงรูปแบบข้อมูลจาก Double เป็น Uint8

หลังจากการแปลงข้อมูลกลับเป็นรูปแบบ Uint8 นั้นสามารถนำข้อมูลที่ได้ไปใช้ในการชดเชยการสูญเสียสมรรถภาพการได้ยินผ่านทางหูฟังของผู้ใช้งาน หรือนำไปบันทึกเป็นไฟล์ข้อมูลที่ถูกชดเชยการสูญเสียสมรรถภาพการได้ยินได้

3.1.8.3 การออกแบบระบบการชดเชยการสูญเสียการได้ยิน

สำหรับระบบที่ใช้สำหรับการชดเชยการสูญเสียการได้ยินจะนำข้อมูลที่ได้จากสัญญาณข้อมูลจากรูป Uint 8 ที่ถูกแปลงเป็นสัญญาณในรูป Double ในช่วง -1 ถึง 1 มาใช้ในการประมวลผลกับวงจรกรองความถี่แบบดิจิทัลชนิดผลตอบสนองของอิมพัลส์จำนวนไม่จำกัด (IIR: Infinite Impulse Response) ซึ่งการสร้างวงจรกรองความถี่แบบดิจิทัลสามารถออกแบบได้จากการประมวลผลบนคลาวด์เซิร์ฟเวอร์ที่ใช้สำหรับคำนวณค่าสัมประสิทธิ์เทอมเศษ (Numerator), ค่าสัมประสิทธิ์ทอมส่วน (Denominator) และค่าอัตราการขยาย (Gain) ของวงจรกรองความถี่ต่ำผ่าน (Low pass filter), วงจรกรองความถี่แถบผ่าน (Band pass filter) และวงจรกรองความถี่สูงผ่าน (High pass filter) โดยจะมีระบบการชดเชยการสูญเสียการได้ยินแสดงดังรูปที่ 3.124



รูปที่ 3.124 แผนภาพระบบการชดเชยการสูญเสียการได้ยิน

ในการชดเชยการสูญเสียสมรรถภาพการได้ยินด้วยวงจรกรองความถี่แบบดิจิทัล อาจจะทำให้ผลลัพธ์ของกระบวนการมีค่ามากกว่า 1 หรือน้อยกว่า -1 ได้ ซึ่งจะทำให้เสียงที่ออกมาทางหูฟังจะเกิดการ Overflow มีผลทำให้เสียงที่ออกมามีคุณภาพต่ำและเกิดเสียงที่แตก โดยในแต่ละวงจรกรองความถี่แบบดิจิทัลจะมีอัตราการขยายขนาดของสัญญาณเพื่อให้สามารถชดเชยการสูญเสียสมรรถภาพการได้ยินหรือค่าสัมประสิทธิ์เทอมส่วนที่มีค่าไม่ได้อยู่ในช่วง -1 ถึง 1 ซึ่งจะทำให้ข้อมูลเกิดการ Overflow จึงจำเป็นต้องลดขนาดของข้อมูลก่อนที่จะนำไปใช้ โดยจะถูกลดลงเป็นจำนวนเดียวกับค่าอัตราการขยายสูงสุดในแต่ละวงจรกรองความถี่แบบดิจิทัล เพื่อลดการเกิด Overflow และทำให้ข้อมูลที่ได้ออกมามีคุณภาพมากขึ้น

โดยหลังจากการปรับขนาดของข้อมูลเสียงที่จะทำการชดเชย จะสามารถนำข้อมูลดังกล่าวไปใช้ในกระบวนการชดเชยการสูญเสียสมรรถภาพการได้ยินด้วยวงจรกรองความถี่แบบดิจิทัล โดยจะนำค่าสัมประสิทธิ์เทอมส่วน (Numerator) และค่าสัมประสิทธิ์เทอมเศษ (Denominator) มาใช้ร่วมกับสมการผลต่างสืบเนื่อง (Difference Equations) แสดงดังรูปที่ 3.95 โดยค่าสัมประสิทธิ์เทอมเศษและสัมประสิทธิ์เทอมส่วนของวงจรกรองความถี่มีจำนวนอย่างละ 3 ค่า จึงสามารถเขียนสมการผลต่างสืบเนื่องได้ดังสมการที่ 3.4 และ 3.5

$$H(z) = \frac{a_0+a_1z^{-1}+a_2z^{-2}}{b_0+b_1z^{-1}+b_2z^{-2}} \quad (3.4)$$

$$y(n) = a_0x(n) + a_1x(n-1) + a_2x(n-2) - b_1y(n-1) - b_2y(n-2) \quad (3.5)$$

โดยจากสมการที่ 3.5 จะเห็นได้ว่าต้องนำข้อมูลเสียงที่จะทำการชดเชยมาเข้าสมการเพื่อที่จะทำการคำนวณหาค่าข้อมูลเสียงที่ถูกชดเชยแล้วในแต่ละตำแหน่ง ซึ่งจำนวนของข้อมูลขาเข้าจะมีจำนวนเท่ากับจำนวนข้อมูลขาออก โดยจะสามารถเขียนในโปรแกรมได้ดังรูปที่ 3.125

```
List<num> convolve(List<num> data, List Num, List Den, int len) {
  List<num> y = List<num>.generate(len, (indexs) => 0);
  List<num> x = data;
  for (int n = 2; n < len; n++) {
    y[n] = (Num[0] * x[n]) +
        (Num[1] * x[n - 1]) +
        (Num[2] * x[n - 2]) -
        (Den[1] * y[n - 1]) -
        (Den[2] * y[n - 2]);
  }
  return y;
}
```

รูปที่ 3.125 คำสั่งในการเขียนโปรแกรมระบบการชดเชยการสูญเสียด้วยสมการผลต่างสืบเนื่อง

หลังจากการที่นำข้อมูลเสียงที่จะทำการชดเชยผ่านวงจรกรองความถี่แบบดิจิทัลด้วยสมการผลต่างสืบเนื่อง จะได้ข้อมูลที่ถูกชดเชยแล้วแต่ยังไม่สามารถนำมาชดเชยได้ เนื่องจากขนาดข้อมูลยังมีค่าที่ยังต่ำและยังไม่สามารถเพิ่มระดับของเสียงให้สามารถชดเชยได้ทุกความถี่ จึงมีค่าอัตราขยายให้กับข้อมูลที่ผ่านวงจรกรองความถี่แบบดิจิทัลของแต่ละวงจร ก่อนที่จะนำค่ามารวมกัน ดังรูปที่ 3.95 โดยจะสามารถเขียนเป็นโปรแกรมได้ดังรูปที่ 3.126

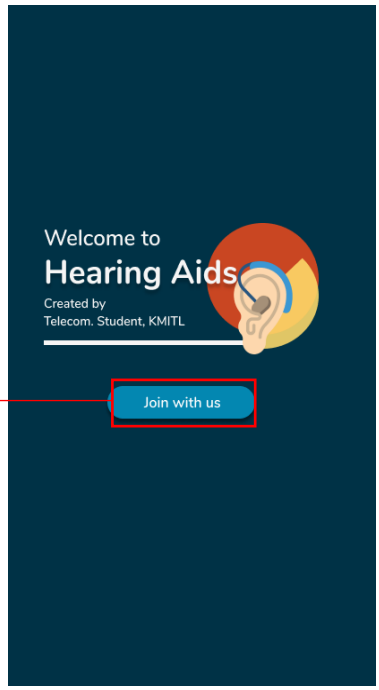
```
Future<List<num>> generateSampleRate(List<num> data, int len) async {
  // TODO: do calculation here!
  List<num> lpfConvolve = await convolve(
    data,
    state.lpfCoeff!.getRange(0, 3).toList(),
    state.lpfCoeff!.getRange(3, 6).toList(),
    len,
  );
  List<num> bpfConvolve = await convolve(
    data,
    state.bpfCoeff!.getRange(0, 3).toList(),
    state.bpfCoeff!.getRange(3, 6).toList(),
    len,
  );
  List<num> hpfConvolve = await convolve(
    data,
    state.hpfCoeff!.getRange(0, 3).toList(),
    state.hpfCoeff!.getRange(3, 6).toList(),
    len,
  );
  List<num> x = [];
  for (var i = 0; i < lpfConvolve.length; i++) {
    x.add((lpfConvolve[i] * state.lpfCoeff![6]) +
        (bpfConvolve[i] * state.bpfCoeff![6]) +
        (hpfConvolve[i] * state.hpfCoeff![6]));
  }
  return x;
}
```

รูปที่ 3.126 คำสั่งในการเขียนโปรแกรมระบบการทำงานวงจรกรองความถี่แบบดิจิทัล

โดยหลังจากนำสัญญาณข้อมูลในรูป Double ผ่านวงจรกรองความถี่แบบดิจิทัลจะได้สัญญาณข้อมูลในรูป Double หลังผ่านวงจรกรองความถี่ และนำไปแปลงสัญญาณกลับตามระบบของรูปที่ 3.120 เพื่อใช้ในการชดเชยการสูญเสียสมรรถภาพการได้ยินเพื่อให้ผู้ใช้งานได้ยินเสียงที่ถูกชดเชยตามความเสียหายของหูผู้ใช้งาน

3.1.9 การออกแบบแอปพลิเคชันผ่านภาษาโปรแกรม Flutter ของแอปพลิเคชันเครื่องช่วยฟัง

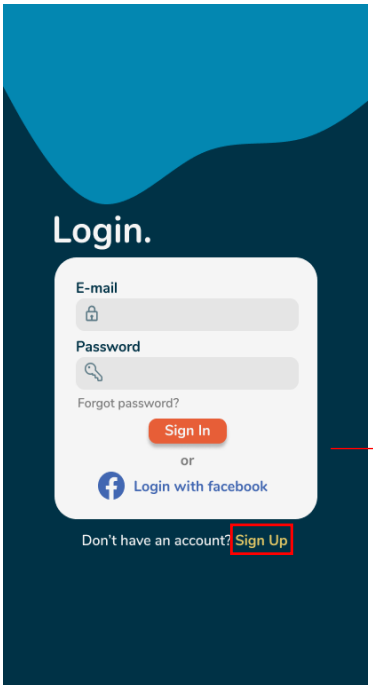
3.1.9.1 หน้าสำหรับเข้าสู่แอปพลิเคชัน



กดเพื่อเข้าสู่หน้าเข้าสู่ระบบ

รูปที่ 3.127 หน้าสำหรับเข้าสู่แอปพลิเคชัน

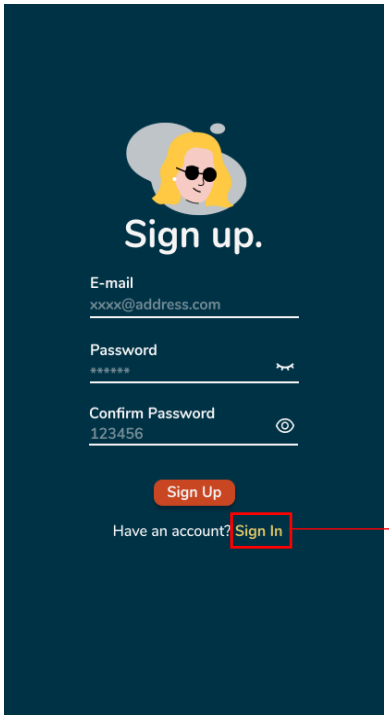
3.1.9.2 หน้าเข้าสู่ระบบผู้ใช้งาน



กดเพื่อเข้าสู่หน้าลงทะเบียนผู้ใช้ใหม่

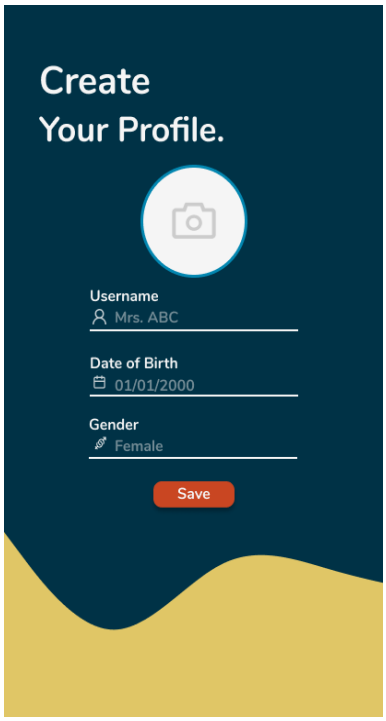
รูปที่ 3.128 หน้าเข้าสู่ระบบผู้ใช้งาน

3.1.9.3 หน้าลงทะเบียนผู้ใช้งานใหม่



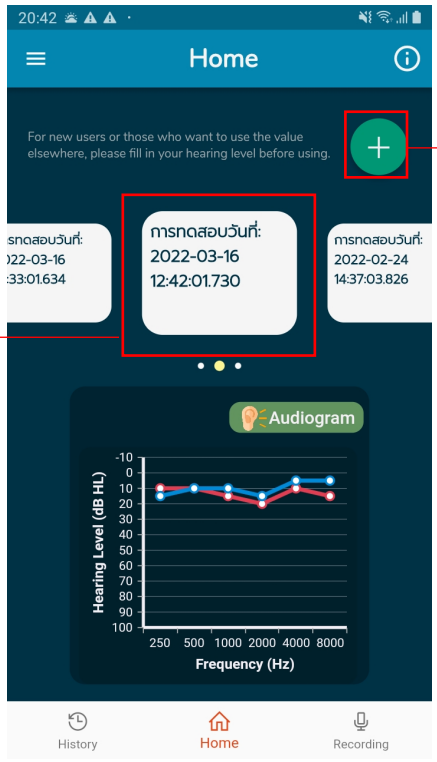
รูปที่ 3.129 หน้าลงทะเบียนผู้ใช้งาน

3.1.9.4 หน้ากรอกประวัติผู้ใช้งานใหม่



รูปที่ 3.130 หน้ากรอกประวัติผู้ใช้งานใหม่

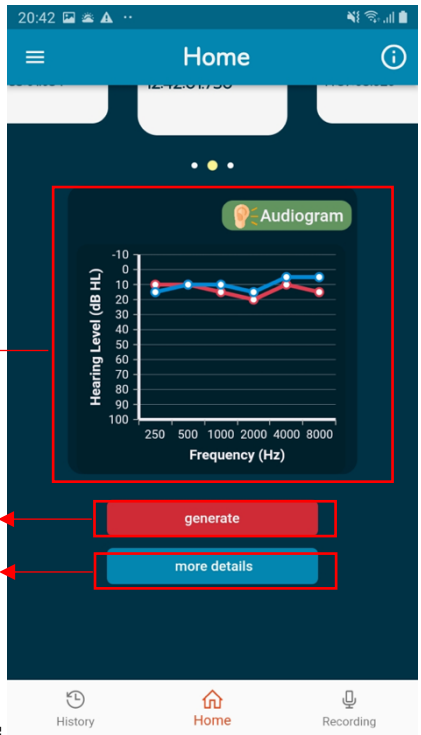
3.1.9.5 หน้าหลักของตัวแอปพลิเคชัน



กดปุ่มเพื่อทำการใส่ค่า Audiogram

กดกล่องสีขาว เพื่อแสดงกราฟการได้ยิน และเพื่อเลือกใช้ ในฟังก์ชันเครื่องช่วยฟัง

รูปที่ 3.131 หน้าหลักของแอปพลิเคชันส่วนบน



กราฟ Audiogram ของวันที่ทำการทดสอบการได้ยิน

กดเพื่อไปยังหน้าเครื่องช่วยฟัง

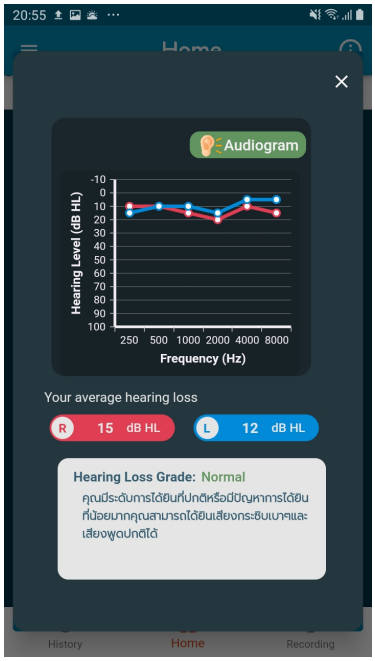
กดเพื่อไปยังรายละเอียดของ Audiogram

รูปที่ 3.132 หน้าหลักของแอปพลิเคชันส่วนล่าง

3.1.9.6 เมื่อกดปุ่ม ‘generate’ ระบบจะแสดงวิธีการใช้งานของแอปพลิเคชันเครื่องช่วยฟังแสดงตามรูปที่ 3.133

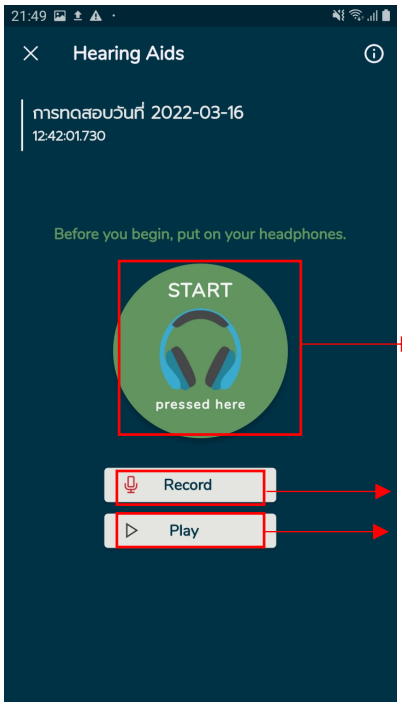
รูปที่ 3.133 วิธีการใช้งานของแอปพลิเคชันเครื่องช่วยฟัง

3.1.9.7 เมื่อกดปุ่ม ‘more details’ ระบบจะแสดงหน้ารายละเอียดของกราฟการได้ยิน

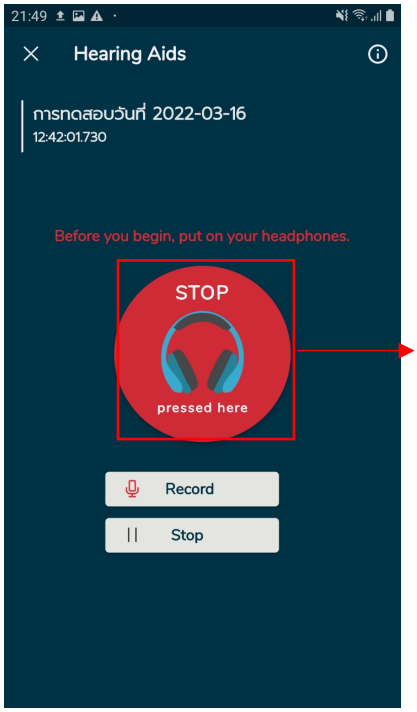


รูปที่ 3.134 หน้ารายละเอียดของกราฟการได้ยิน

3.1.9.8 หน้าฟังก์ชันการทำงานของเครื่องช่วยฟัง



รูปที่ 3.135 หน้าเครื่องช่วยฟัง

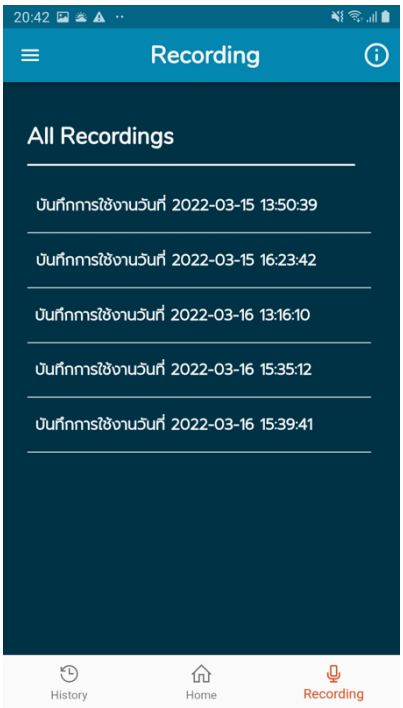


รูปที่ 3.136 หน้าเครื่องช่วยฟังเมื่อทำการกดใช้ฟังก์ชันเครื่องช่วยฟัง

3.1.9.9 หน้าแสดงประวัติค่าการทดสอบสมรรถภาพการได้ยิน

รูปที่ 3.137 หน้าแสดงประวัติค่าการทดสอบสมรรถภาพการได้ยิน

3.1.9.10 หน้าแสดงไฟล์บักทึกเสียงที่ผ่านการชดเชยการสูญสียการได้ยิน



รูปที่ 3.138 หน้าแสดงไฟล์บักทึกเสียงที่ผ่านการชดเชยการสูญสียการได้ยิน

3.2 เครื่องมือที่ใช้ในการทดสอบ

ในปริญญานิพนธ์นี้ มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลอง ดังนี้

3.2.1 โทรศัพท์เคลื่อนที่

โทรศัพท์ยี่ห้อ Samsung รุ่น Galaxy J7 Core ขนาดหน้าจอ 5.5 นิ้ว ระบบปฏิบัติการ Android เวอร์ชั่น 7.0 ช่อง Micro-USB เป็น Wi-Fi 802.11 เพื่อรองรับการสร้างแอปพลิเคชันจากภาษา Flutter ให้ทดสอบประสิทธิภาพของแอปพลิเคชัน และทดสอบการทดสอบสมรรถภาพการได้ยิน

รูปที่ 3.139 โทรศัพท์ Samsung Galaxy J7 Core [21]

3.2.2 สาย USB 2.0 Micro Type B

สาย USB 2.0 Micro Type B เป็นสายประเภทไมโครตัวสายจะมีความยืดหยุ่น หัวต่อมีขนาดเล็ก และมีความหนาเป็นครึ่งนึงของสาย Mini USB โดยสายชนิดนี้ถูกออกแบบมาให้มีขนาดเล็กใช้เชื่อมต่อกับโทรศัพท์อย่าสมาร์ทโฟน กล้องดิจิทัล โดยที่อีกด้านของสายจะเป็นหัวต่อแบบ USB Type A เพื่อใช้เชื่อมต่อกับคอมพิวเตอร์ หรือที่ชาร์จแบตเตอรี่ เนื่องจากมีความเร็วสูงในการเชื่อมต่อ (480 Mbit/s)

รูปที่ 3.140 สาย USB 2.0 Micro Type B [22]

3.2.3 หูฟังขนาด 3.5 มิลลิเมตร

ช่องเสียบหูฟังที่พบมากที่สุดและมาตรฐานคือหูฟังที่มีเส้นผ่านศูนย์กลาง 3.5 มม. ซึ่งเรียกอีกอย่างว่าแจ็คมินิ มันเป็นแจ็คควอเตอร์ดั้งเดิมที่มีขนาดเล็กกว่าซึ่งยังคงใช้กันอย่างแพร่หลาย เช่นเดียวกับแจ็คหูฟังขนาด 6.5 มม. 3.5 มม. รองรับโมโนและสเตอริโอ อย่างไรก็ตามฟังก์ชันอื่นที่แจ็คหูฟัง 3.5 มม. มีคือไมโครโฟนที่รองรับการโทร

รูปที่ 3.141 หูฟังขนาด 3.5 มิลลิเมตร [23]

3.3 การจัดเก็บผลการทดลอง

3.3.1 การสร้างบัญชีผู้ใช้งานและการเข้าสู่ระบบบัญชีผู้ใช้งาน

3.3.1.1 การสร้างบัญชีผู้ใช้งานผ่านอีเมลและรหัสผ่าน

3.3.1.2 กรณีผู้ใช้งานใส่ระบุอีเมลผิดหรือรหัสผ่านไม่ถูกต้อง

3.3.1.3 กรณีผู้ใช้งานลืมรหัสผ่าน

3.3.1.4 กรณีผู้ใช้งานทำการเปลี่ยนรหัสผ่าน

3.3.2 การทดสอบสมรรถภาพการได้ยินด้วยสัญญาณความถี่เดียว

3.3.3 การวัดประสิทธิภาพของแอปพลิเคชันสำหรับการทดสอบการได้ยิน

3.3.3.1 การทดสอบความเสถียรภาพของแอปพลิเคชัน

3.3.3.2 การวัดแนวโน้มผลทดสอบสมรรถภาพการได้ยินระหว่างแอปพลิเคชันกับศูนย์การทดสอบสมรรถภาพการได้ยิน

3.3.4 ผลลัพธ์ที่ได้จากการใช้งานแอปพลิเคชันการทดสอบสมรรถภาพการได้ยิน

3.3.5 ผลลัพธ์ที่ได้จากการออกแบบวงจรกรองความถี่แบบดิจิทัลสำหรับเครื่องช่วยฟัง

3.3.5.1 กราฟการได้ยินที่นำมาใช้ทดสอบการออกแบบวงจรกรองความถี่แบบดิจิทัลสำหรับเครื่องช่วยฟัง

3.3.5.2 ผลลัพธ์ที่ได้จากกราฟการได้ยินที่นำมาใช้ทดสอบการออกแบบวงจรกรองความถี่แบบดิจิทัลสำหรับเครื่องช่วยฟัง

3.3.5.3 การนำค่าการทดสอบสรรถภาพการได้ยินจากฐานข้อมูลมาใช้ในการออกแบบวงจรกรองความถี่แบบดิจิทัล

3.3.6 การทดสอบคลาวด์เซิร์ฟเวอร์ที่ใช้สำหรับการออกแบบวงจรกรองความถี่

3.3.6.1 การวัดเวลาที่ใช้ทั้งระบบบนแอปพลิเคชันเครื่องช่วยฟัง

3.3.7 ผลลัพธ์ที่ได้จากการใช้งานแอปพลิเคชันเครื่องช่วยฟัง

บทที่ 4

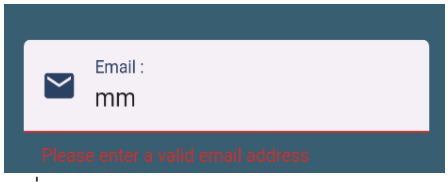
ผลการทดลอง

4.1 การสร้างบัญชีผู้ใช้งานและการเข้าสู่ระบบบัญชีผู้ใช้งาน

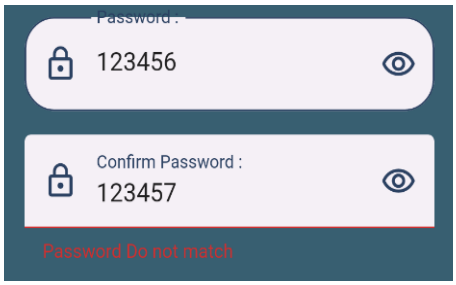
4.1.1 การสร้างบัญชีผู้ใช้งานผ่านอีเมลและรหัสผ่าน

ในการทำงานของแอปพลิเคชันการทดสอบสมรรถภาพการได้ยิน ผู้ใช้งานจำเป็นต้องสร้างบัญชีผู้ใช้งานขึ้นก่อนที่เข้าสู่ตัวระบบหลัก ซึ่งการสร้างบัญชีผู้ใช้งานสามารถสร้างได้ทั้งจากอีเมลและการเข้าสู่ระบบผ่าน Facebook โดยสร้างระบบของบัญชีผู้ใช้งานเพื่อเป็นการเก็บประวัติการทดสอบสมรรถภาพการได้ยินของผู้ใช้งาน เพศ และวันเกิด เพื่อใช้งานต่อในส่วนของแอปพลิเคชันสำหรับเครื่องช่วยฟัง และดูแนวโน้มอายุและเพศของผู้ใช้งาน เพื่อปรับปรุงและพัฒนาแอปให้เหมาะสมมากยิ่งขึ้น

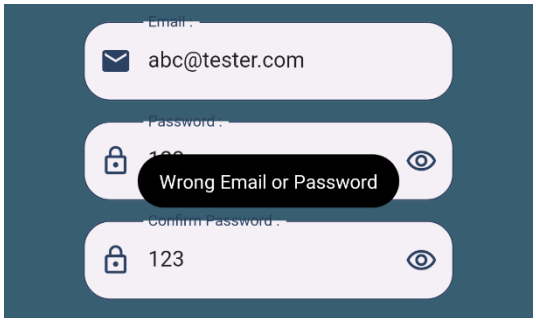
การสร้างบัญชีผู้ใช้งานผ่านอีเมลและรหัสผ่าน ผู้ใช้งานจะต้องตั้งรหัสผ่านให้ความยาวอย่างน้อย 6 ตัวอักษร และต้องระบุแอดเดรสของอีเมลให้ครบถ้วน



รูปที่ 4.1 กรณีไม่ได้ระบุแอดเดรสของอีเมล



รูปที่ 4.2 กรณีระบุรหัสผ่านไม่ตรงตามรหัสผ่านที่สร้าง



รูปที่ 4.3 กรณีระบุรหัสผ่านไม่ครบ 6 ตัวอักษร

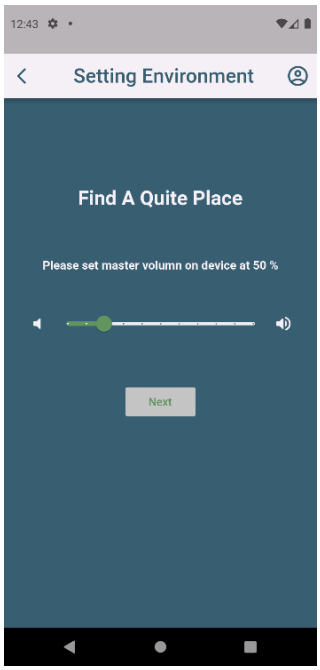
4.1.2 กรณีผู้ใช้งานใส่ระบุอีเมลผิดหรือรหัสผ่านไม่ถูกต้องในการเข้าสู่ระบบบัญชีผู้ใช้

ในกรณีที่ผู้ใช้งานเข้าสู่ระบบด้วยอีเมล ถ้าหากผู้ใช้งานกรอกรหัสผ่านหรืออีเมลไม่ตรงตามที่ลงทะเบียนไว้ระบบจะทำการแจ้งเตือนข้อความแสดงดังรูปที่ 4.4 เพื่อแจ้งเตือนให้ผู้ใช้งานกรอกส่วนที่ผิดใหม่อีกครั้ง

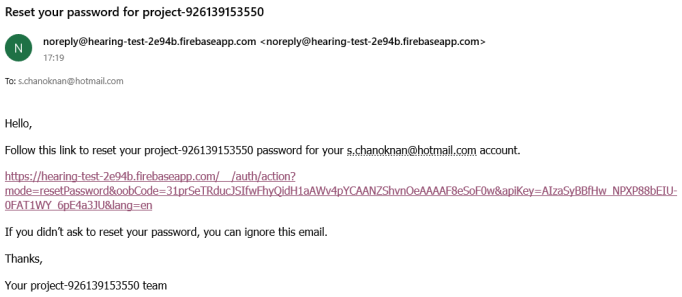
รูปที่ 4.4 กรณีใส่อีเมลหรือรหัสผ่านไม่ถูกต้อง

4.1.3 กรณีผู้ใช้งานลืมรหัสผ่านให้ทำการกรอกอีเมลที่ใช้สมัครบัญชีผู้ใช้แล้วกด Send Email

ในกรณีที่ผู้ใช้งานลืมรหัสผ่าน ให้กดปุ่ม ‘Forgot Password’ ในหน้า Login เมื่อกดปุ่มดังกล่าวแล้ว ระบบจะแสดงหน้าดังรูปที่ 4.5 เพื่อให้ผู้ใช้งานกรอกอีเมลที่ได้ทำการลงทะเบียนไว้ หลังจากนั้นระบบจะทำการส่งลิ้งสำหรับการเปลี่ยนรหัสผ่านแสดงดังรูปที่ 4.6 ไปยังที่ระบุไว้อีเมล เพื่อให้ผู้ใช้งานทำการเปลี่ยนรหัสผ่านโดยการเปลี่ยนรหัสผ่านจะดำเนินการตามรูปที่ 4.7



รูปที่ 4.5 กรณีลืมรหัสผ่าน


Reset your password for project-926139153550

noreply@hearing-test-2e94b.firebaseapp.com <noreply@hearing-test-2e94b.firebaseapp.com>
17:19

To: s.chanoknan@hotmail.com

Hello,

Follow this link to reset your project-926139153550 password for your s.chanoknan@hotmail.com account.

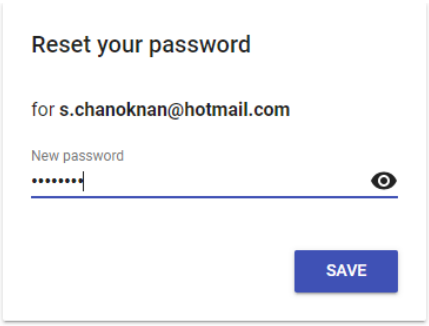
https://hearing-test-2e94b.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=31prSeTRducJSIfwFhyQjdH1aAWv4pYCAANZShvnOeAAAAF8eSoF0w&apiKey=AIzaSyBBfHw_NPXP88bEIU-0FAT1WY_6pE4a3JU&lang=en

If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-926139153550 team

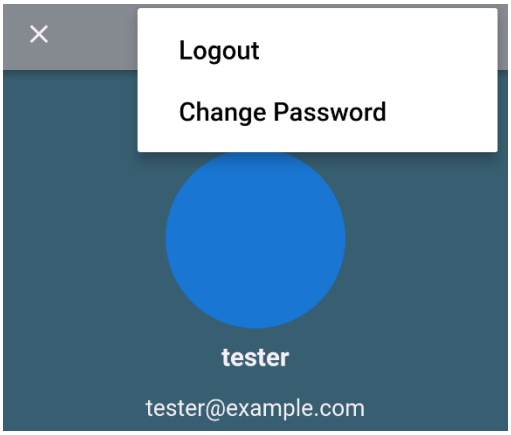
รูปที่ 4.6 ตัวอย่างอีเมลที่ส่งไปยังผู้ใช้งานสำหรับการเปลี่ยนรหัสผ่าน



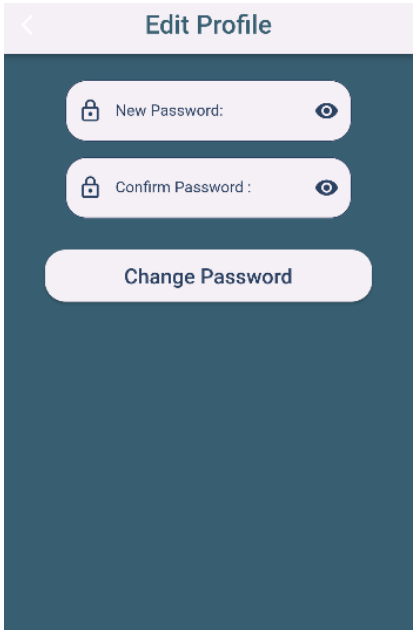
รูปที่ 4.7 หน้าต่างที่ในการเปลี่ยนรหัสผ่านผู้ใช้งาน

4.1.4 กรณีผู้ใช้งานทำการเปลี่ยนรหัสผ่าน

ในกรณีที่ผู้ใช้งานเข้าสู่ระบบแล้วต้องการปลี่ยนพาสเวิร์ด ผู้ใช้งานสามารถเข้าไปที่หน้าแสดงประวัติของผู้ใช้งาน และคลิกที่มุมขวาบนจากนั้นกด Change Password แสดงตามรูปที่ 4.8 จากนั้นระบบจะแสดงหน้าสำหรับการเปลี่ยนพาสเวิร์ดแสดงดังรูปที่ 4.9



รูปที่ 4.8 ปุ่มที่ใช้สำหรับเปลี่ยนพาสเวิร์ด



รูปที่ 4.9 หน้าที่ใช้สำหรับเปลี่ยนรหัสผ่านผู้ใช้งาน

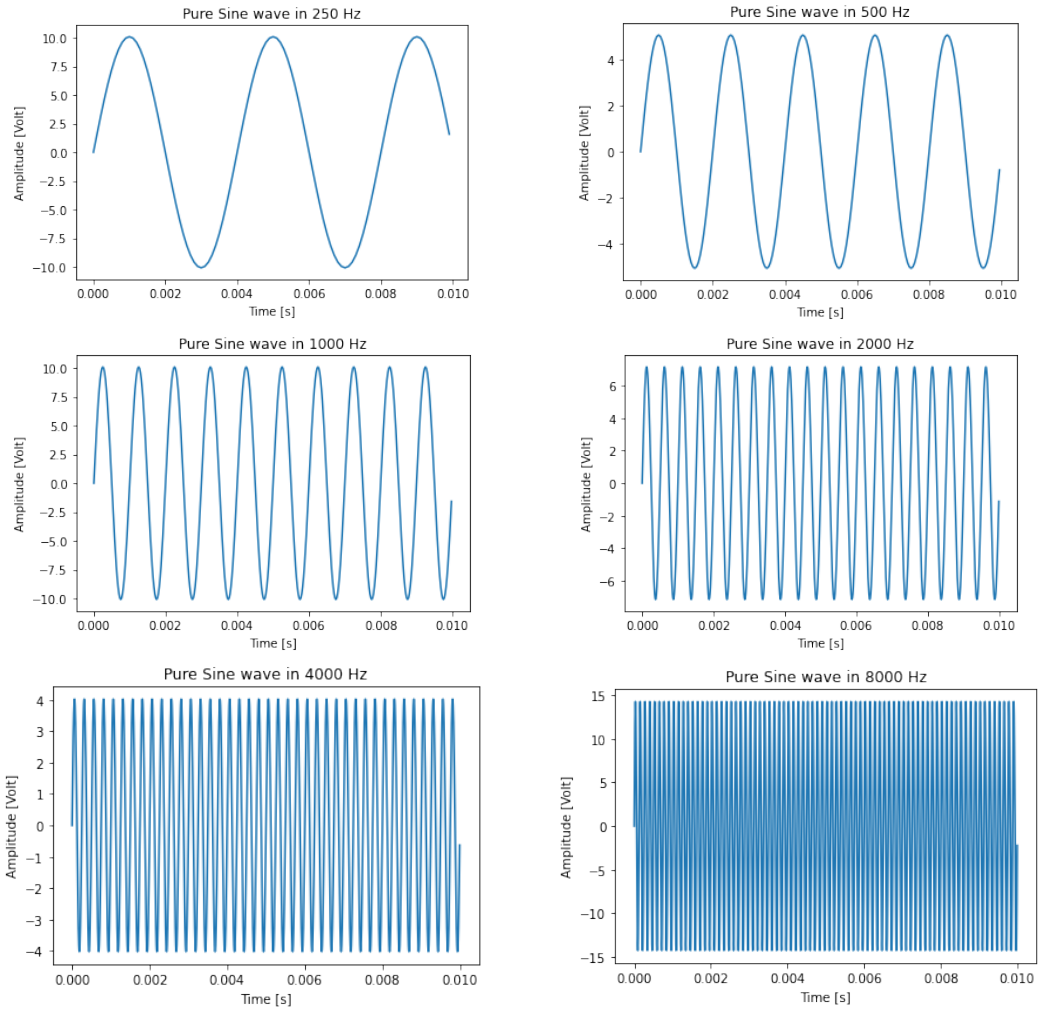
4.2 การทดสอบสมรรถภาพการได้ยินด้วยสัญญาณความถี่เดียว

จากการการออกแบบเสียงที่ใช้ในการทดสอบจะต้องคำนวณระดับความดังของของเสียงที่ใช้ทดสอบในแต่ละความถี่ เพื่อนำไปใช้ในโปรแกรมที่เขียนด้วยภาษา Flutter โดยจะมีระดับความดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 ผลการคำนวณระดับความดังของเสียงที่ออกแบบในแต่ละความถี่

ความถี่	ระดับความดังของเสียง [Volt]
250 Hz	10.09183182
500 Hz	5.057897271
1000 Hz	10.09183182
2000 Hz	7.144469793
4000 Hz	4.017630611
8000 Hz	14.25509134

เมื่อได้คำนวณค่าระดับความดังของเสียงดังตารางที่ 4.1 จะสามารถนำค่าระดับความดังของเสียงไปสร้างสัญญาณความถี่เดียวเพื่อไปใช้ในการทดสอบสมรรถภาพการได้ยิน ซึ่งจะมีลักษณะของสัญญาณดังรูปที่ 4.10



รูปที่ 4.10 สัญญาณความถี่เดียวที่ใช้ทดสอบในแต่ละความถี่

เมื่อได้สัญญาณความถี่เดียวที่ใช้ทดสอบในแต่ละความถี่ ส่วนต่อจะเป็นการบันทึกเสียงด้วยโปรแกรมภาษา Python ด้วยนามสกุล .mp3 ซึ่งจะนำเสียงที่บันทึกได้ไปใช้ในส่วนของแอปพลิเคชันเพื่อใช้ในการทดสอบสมรรถภาพการได้ยิน

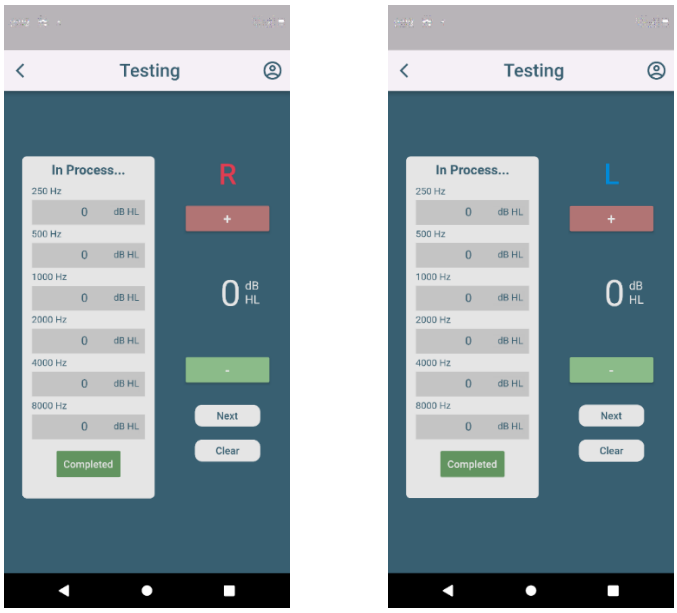
ในการทดสอบสัญญาณความถี่เดียวจะต้องนำเสียงที่สร้างขึ้นเข้าไปในโปรแกรมที่เขียนด้วยภาษา Flutter เพื่อใช้ในแอปพลิเคชัน และในการเล่นเสียงจะสามารถปรับค่าได้ตั้งแต่ 0 ถึง 100% ของเสียงที่สร้างขึ้น จึงจะต้องคำนวณระดับเสียงที่ตั้งแต่ -10 ถึง 100 dB HL เพื่อให้สามารถนำไปปรับระดับความดังของเสียงในส่วนของโปรแกรมได้ ซึ่งผลลัพธ์ที่ได้แสดงตามตารางที่ 4.2

ตารางที่ 4.2 ผลการคำนวณระดับความดังของเสียงของแต่ละระดับเสียง dB HL หน่วยเป็น Volt

ระดับความดังของเสียง (dB HL)	แอมพลิจูดที่ใช้ทดสอบการได้ยินในแต่ละความถี่					
	250 Hz	500 Hz	1000 Hz	2000 Hz	4000 Hz	8000 Hz
-10	0.00003191	0.00001599	0.00003191	0.00002259	0.00001270	0.00004508
-5	0.00005675	0.00002844	0.00005675	0.00004018	0.00002259	0.00008016
0	0.00010092	0.00005058	0.00010092	0.00007144	0.00004018	0.00014255
5	0.00017946	0.00008994	0.00017946	0.00012705	0.00007144	0.00025350
10	0.00031913	0.00015994	0.00031913	0.00022593	0.00012705	0.00045079
15	0.00056751	0.00028443	0.00056751	0.00040176	0.00022593	0.00080162
20	0.00100918	0.00050579	0.00100918	0.00071445	0.00040176	0.00142551
25	0.00179461	0.00089944	0.00179461	0.00127049	0.00071445	0.00253495
30	0.00319132	0.00159945	0.00319132	0.00225928	0.00127049	0.00450786
35	0.00567505	0.00284426	0.00567505	0.00401763	0.00225928	0.00801623
40	0.01009183	0.00505790	0.01009183	0.00714447	0.00401763	0.01425509
45	0.01794610	0.00899435	0.01794610	0.01270486	0.00714447	0.02534954
50	0.03191317	0.01599448	0.03191317	0.02259280	0.01270486	0.04507856
55	0.05675054	0.02844265	0.05675054	0.04017631	0.02259280	0.08016227
60	0.10091832	0.05057897	0.10091832	0.07144470	0.04017631	0.14255091
65	0.17946097	0.08994355	0.17946097	0.12704864	0.07144470	0.25349535
70	0.31913174	0.15994476	0.31913174	0.22592797	0.12704864	0.45078557
75	0.56750541	0.28442647	0.56750541	0.40176306	0.22592797	0.80162270
80	1.00918318	0.50578973	1.00918318	0.71444698	0.40176306	1.42550913
85	1.79460967	0.89943546	1.79460967	1.27048635	0.71444698	2.53495354
90	3.19131743	1.59944755	3.19131743	2.25927972	1.27048635	4.50785569
95	5.67505408	2.84426465	5.67505408	4.01763061	2.25927972	8.01622695
100	10.0918318	5.05789727	10.0918318	7.14446979	4.01763061	14.25509134

จากผลการคำนวณตามตารางที่ 4.2 จะนำถูกไปใช้กำหนดระดับความดังในการทดสอบการได้ยินในส่วนของโปรแกรม Flutter เพื่อควบคุมเสียงให้เป็นไปตามกลไกของแอปพลิเคชันที่ได้ออกแบบไว้ โดยจะมีหน้าทดสอบในแอปพลิเคชันเพื่อทดสอบเสียงในการทดสอบสมรรถภาพการได้ยิน

โดยจะแยกทั้งหูซ้ายและขวาเพื่อสะดวกในการทดสอบเสียงและทดสอบสมรรถภาพการได้ยินให้สามารถใช้งานได้



(ก) (ข)

รูปที่ 4.11 หน้าการทดสอบเสียง

(ก) ฝั่งขวา (ข) ฝั่งซ้าย

จากการทดสอบบนแอปพลิเคชันสำหรับการทดสอบสมรรถภาพการได้ยิน สามารถแสดงเสียงที่ถูกสร้างขึ้นได้ทั้ง 2 ข้าง ตามที่ปริญญานิพนธ์ต้องการ ซึ่งจะสามารถปรับระดับความดังของเสียงในปุ่มบวก (+) และลบ (-) และสามารถเปลี่ยนเสียงหรือความถี่โดยกดปุ่ม complete ตามที่ได้ออกแบบไว้ แสดงตามรูปที่ 4.11

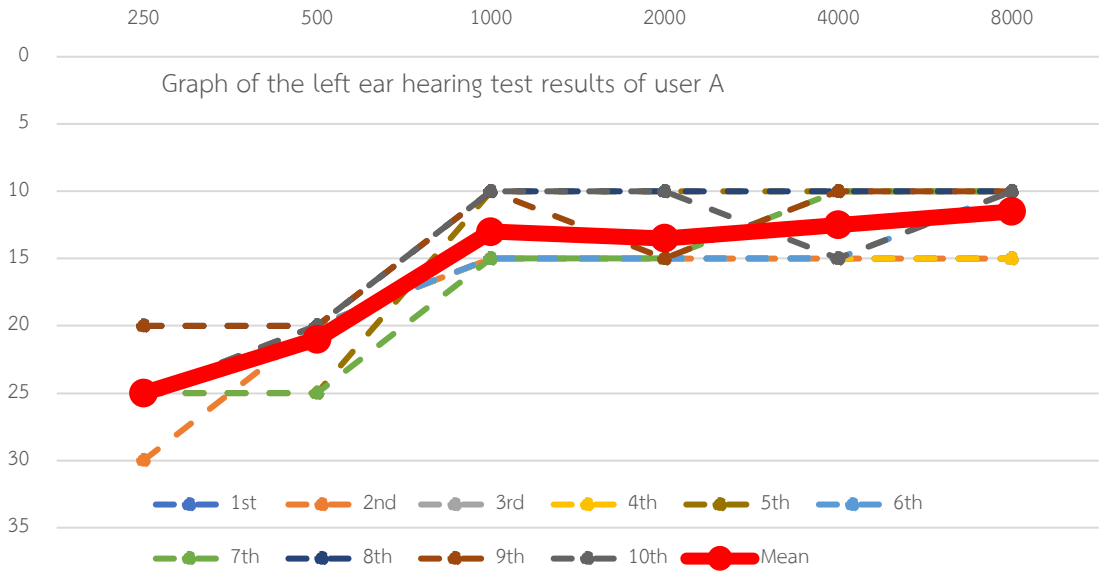
4.3 การวัดประสิทธิภาพของแอปพลิเคชันสำหรับการทดสอบการได้ยิน

4.3.1 การทดสอบความเสถียรภาพของแอปพลิเคชัน

จากแอปพลิเคชันสำหรับทดสอบการได้ยิน ได้มีการวัดความเสถียรภาพของแอปพลิเคชัน โดยมีการทดสอบสมรรถภาพการได้ยินจากผู้ใช้งาน 3 คน โดยมีผลการทดสอบคนละ 10 ครั้ง ในสถานที่ที่มีสภาวะแวดล้อมเดียวกันและเป็นไปตามข้อแนะนำการทดสอบตามแอปพลิเคชันได้ระบุไว้ ซึ่งผลที่ได้จากการทดสอบทั้งหมดมีดังนี้

ตารางที่ 4.3 ผลการทดสอบสมรรถภาพการได้ยินหูซ้ายของผู้ใช้งานคนที่ 1

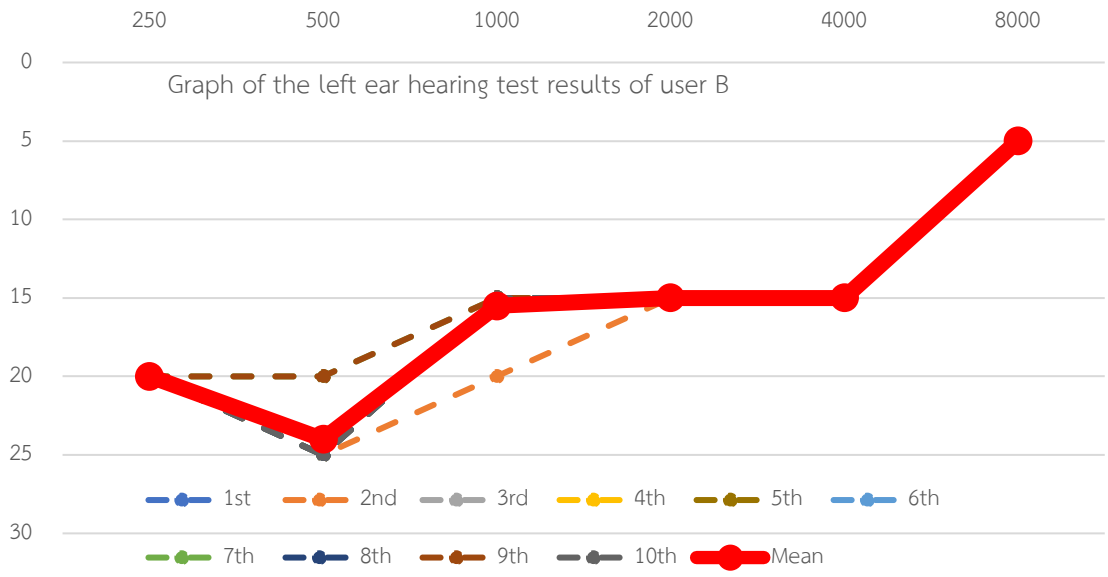
ครั้งที่	250 Hz	500 Hz	1000 Hz	2000 Hz	4000 Hz	8000 Hz
1	25	20	15	15	15	15
2	30	20	15	15	15	15
3	30	20	15	15	10	10
4	25	20	15	15	15	15
5	25	25	10	10	10	10
6	25	20	15	15	15	10
7	25	25	15	15	10	10
8	20	20	10	10	10	10
9	20	20	10	15	10	10
10	25	20	10	10	15	10
ค่าเฉลี่ย	**25**	**21**	**13**	**13.5**	**12.5**	**11.5**
ค่าเบี่ยงเบนมาตรฐาน	**3.162278**	**2**	**2.44949**	**2.291288**	**2.5**	**2.291288**
ค่าความแปรปรวน	**10**	**4**	**6**	**5.25**	**6.25**	**5.25**



รูปที่ 4.12 กราฟผลการทดสอบสมรรถภาพการได้ยินหูซ้ายของผู้ใช้งานคนที่ 1

ตารางที่ 4.4 ผลการทดสอบสมรรถภาพการได้ยินหูซ้ายของผู้ใช้งานคนที่ 2

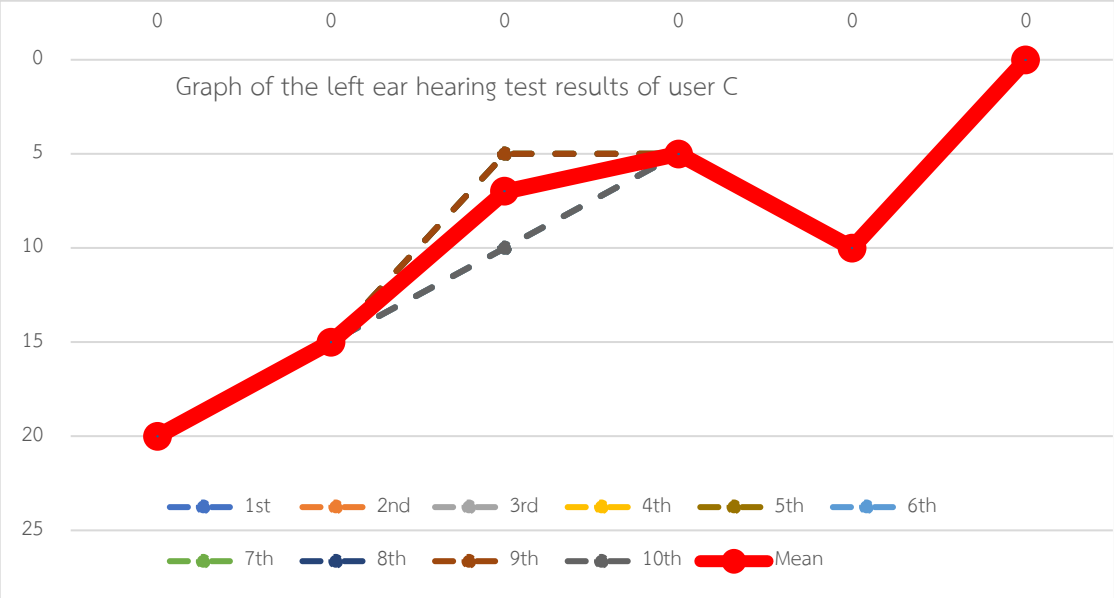
ครั้งที่	250 Hz	500 Hz	1000 Hz	2000 Hz	4000 Hz	8000 Hz
1	20	25	15	15	15	5
2	20	25	20	15	15	5
3	20	25	15	15	15	5
4	20	25	15	15	15	5
5	20	25	15	15	15	5
6	20	25	15	15	15	5
7	20	20	15	15	15	5
8	20	25	15	15	15	5
9	20	20	15	15	15	5
10	20	25	15	15	15	5
ค่าเฉลี่ย	**20**	**24**	**15.5**	**15**	**15**	**5**
ค่าเบี่ยงเบนมาตรฐาน	**0**	**2**	**1.5**	**0**	**0**	**0**
ค่าความแปรปรวน	**0**	**4**	**2.25**	**0**	**0**	**0**



รูปที่ 4.13 กราฟผลการทดสอบสมรรถภาพการได้ยินหูซ้ายของผู้ใช้งานคนที่ 2

ตารางที่ 4.5 ผลการทดสอบสมรรถภาพการได้ยินหูซ้ายของผู้ใช้งานคนที่ 3

ครั้งที่	250 Hz	500 Hz	1000 Hz	2000 Hz	4000 Hz	8000 Hz
1	20	15	10	5	10	0
2	20	15	5	5	10	0
3	20	15	5	5	10	0
4	20	15	5	5	10	0
5	20	15	10	5	10	0
6	20	15	10	5	10	0
7	20	15	5	5	10	0
8	20	15	5	5	10	0
9	20	15	5	5	10	0
10	20	15	10	5	10	0
ค่าเฉลี่ย	**20**	**15**	**7**	**5**	**10**	**0**
ค่าเบี่ยงเบนมาตรฐาน	**0**	**0**	**2.44949**	**0**	**0**	**0**
ค่าความแปรปรวน	**0**	**0**	**6**	**0**	**0**	**0**

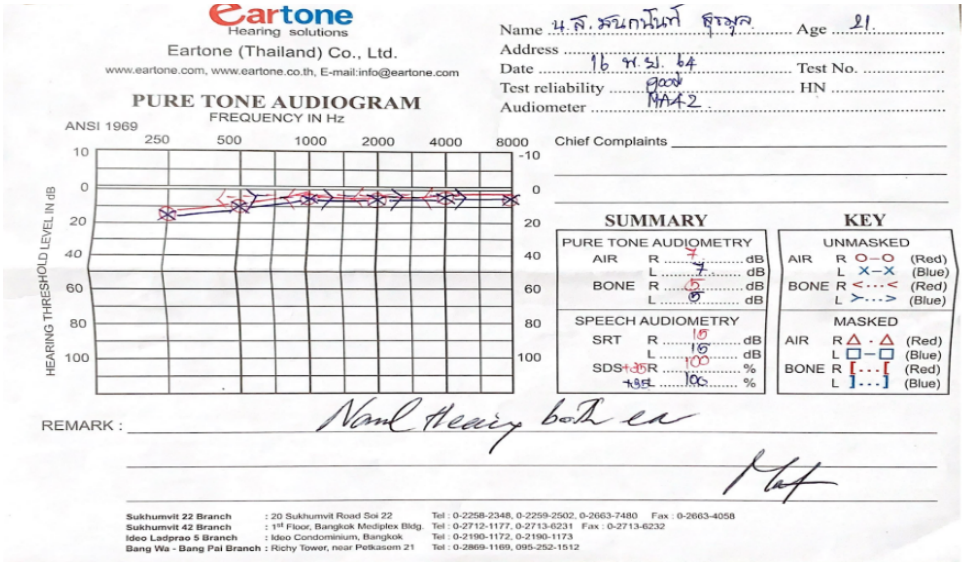


รูปที่ 4.14 กราฟผลการทดสอบสมรรถภาพการได้ยินหูซ้ายของผู้ใช้งานคนที่ 3

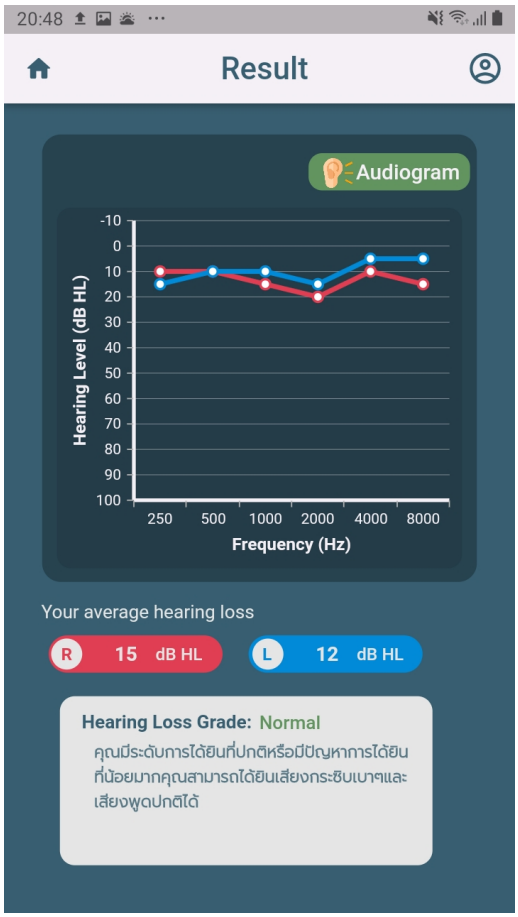
จากตารางที่ 4.3, 4.4 และ 4.5 จะเห็นว่าผลการทดสอบสมรรถภาพการได้ยินมีความใกล้เคียงกันในแต่ละความถี่ของผู้ใช้งานนั้น โดยมีค่าเฉลี่ย ค่าส่วนเบี่ยงเบนมาตรฐานและค่าความแปรปรวนดังในตาราง โดยมีค่าส่วนเบี่ยงเบนมาตรฐานไม่เกิน 5 dB HL ซึ่งเป็นระยะห่างที่ใช้ในการปรับระดับเสียงในการทดสอบสมรรถภาพการได้ยิน ดังนั้นแอปพลิเคชันสำหรับทดสอบการได้ยินมีความเสถียรภาพ

4.3.2 การวัดแนวโน้มผลทดสอบสมรรถภาพการได้ยินระหว่างแอปพลิเคชันกับศูนย์การทดสอบสมรรถภาพการได้ยิน

จากผู้ใช้งานแอปพลิเคชันที่ได้ทำการทดสอบสมรรถภาพการได้ยิน ได้ทำการทดสอบสมรรถภาพการได้ยินที่ศูนย์การทดสอบ และนำผลการทดสอบมาเปรียบเทียบ เพื่อวัดแนวโน้มและลักษณะผลการทดสอบสมรรถภาพการได้ยินของแอปพลิเคชันสำหรับทดสอบการได้ยินให้ใกล้เคียงกับศูนย์การทดสอบ ซึ่งจะสามารถเปรียบเทียบได้ดังนี้



รูปที่ 4.15 ผลการทดสอบสมรรถภาพการได้ยินจากศูนย์การทดสอบ

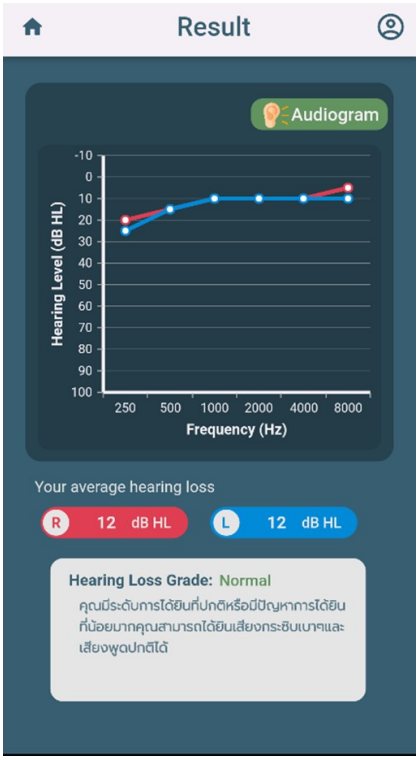


รูปที่ 4.16 ผลการทดสอบสมรรถภาพการได้ยินจากแอปพลิเคชัน

จากผลการทดสอบสมรรถภาพการได้ยินข้างต้นจะเห็นได้ว่าผลการทดสอบจากทั้งสองส่วนคือแอปพลิเคชันและศูนย์การทดสอบมีความใกล้เคียงกัน ซึ่งมีแนวโน้มที่ไปในทิศทางเดียวกัน

4.4 ผลลัพธ์ที่ได้จากการใช้งานแอปพลิเคชันการทดสอบสมรรถภาพการได้ยิน

หลังจากทำการทดสอบการได้ยินทั้งหูขวาและหูซ้ายเสร็จเป็นที่เรียบร้อยจากรูปที่ 4.12 ระบบจะแสดงผลไปยังหน้ากราฟการได้ยิน (Audiogram) ค่าเฉลี่ยของการสูญเสียการได้ยิน และการวิเคราะห์ผลที่ได้จากการทดสอบ แสดงดังรูปที่ 4.17



รูปที่ 4.17 ตัวอย่างผลการทดสอบและกราฟการได้ยินในระดับปกติ

จากรูปที่ 4.17 สีแดงจะแสดงถึงผลลัพธ์ที่ได้จากหูขวา และสีน้ำเงินจะแสดงผลลัพธ์ที่ได้จากหูซ้าย โดยค่าเฉลี่ยของการสูญเสียการได้ยินสามารถคำนวณได้จาก การนำความถี่ 500, 1000 และ 2000 เฮิร์ตซ์ มาบวกกันและหารเฉลี่ยตามสมการที่ 4.1 สาเหตุที่นำความถี่ทั้งสามมาคำนวณค่าเฉลี่ยเนื่องจากความถี่ตั้งแต่ช่วง 500 – 2000 เฮิร์ตซ์ เป็นช่วงความถี่ที่มนุษย์สามารถเข้าใจเสียงพยัญชนะหรือสระที่ใช้ในการสื่อสาร [24] นอกจากนี้ยังมีเงื่อนไขในการคำนวณค่าเฉลี่ยของการสูญเสียการได้ยิน ถ้าหากผลลัพธ์ที่ได้มีค่าเป็นลบจะถูกตั้งค่าเป็นศูนย์ และค่าที่สูงกว่าหนึ่งร้อยจะถูกปรับเป็น 100 [25]

$$\text{ค่าเฉลี่ยการสูญเสียการได้ยิน} = \frac{\text{ระดับการได้ยินที่ความถี่ } \mathbf{500} + \text{ระดับการได้ยินที่ความถี่ } \mathbf{1000} + \text{ระดับการได้ยินที่ความถี่ } \mathbf{2000}}{3} \quad (4.1)$$

ในส่วนของการวิเคราะห์ผลที่ได้จากการทดสอบการได้ยิน การประเมินช่วงของระดับความบกพร่องทางการได้ยินนั้น ปริญญานิพนธ์นี้ได้อ้างอิงผลมาจาก WHO ตามรูปที่ 4.18

Grade of impairment[16]	Corresponding audiometric ISO value[17]	Performance	Recommendations
0 - No impairment	25 dB or better (better ear)	No or very slight hearing problems. Able to hear whispers.	
1 - Slight impairment	26-40 dB (better ear)	Able to hear and repeat words spoken in normal voice at 1 metre.	Counselling. Hearing aids may be needed.
2 - Moderate impairment	41-60 dB (better ear)	Able to hear and repeat words spoken in raised voice at 1 metre.	Hearing aids usually recommended.
3 - Severe impairment	61-80 dB (better ear)	Able to hear some words when shouted into better ear.	Hearing aids needed. If no hearing aids available, lip-reading and signing should be taught.
4 - Profound impairment including deafness	81 dB or greater (better ear)	Unable to hear and understand even a shouted voice.	Hearing aids may help understanding words. Additional rehabilitation needed. Lip-reading and sometimes signing essential.

รูปที่ 4.18 World Health Organization Grades of Hearing Impairment (WHO, 2008) [26]

รูปที่ 4.19 ตัวอย่างผลการทดสอบและกราฟการได้ยิน

(ก) ระดับหูตึงเล็กน้อย (ข) ระดับหูตึงปานกลาง

4.5 ผลลัพธ์ที่ได้จากการออกแบบวงจรกรองความถี่แบบดิจิทัลสำหรับเครื่องช่วยฟัง

4.5.1 กราฟการได้ยินที่นำมาใช้ทดสอบการออกแบบวงจรกรองความถี่แบบดิจิทัลสำหรับเครื่องช่วยฟัง

การทดสอบในหัวข้อนี้จะนำระบบที่ได้ออกแบบในหัวข้อที่ 3.1.6 ใช้กับตัวต้นแบบกราฟการได้ยินแทน โดยจะใช้การประมาณค่าในช่วง (Interpolation) มาปรับใช้กับกราฟการได้ยินตามที่กล่าวไว้ในหัวข้อที่ 3.1.6 ซึ่งประเภทของ Interpolation ที่ใช้คือ การประมาณค่าในช่วงเชิงเส้น (Linear Interpolation) โดยในโปรแกรม MATLAB สามารถเรียกใช้คำสั่งได้ดังนี้

$$\mathrm{yy} = \mathrm{interp}(\mathrm{x}, \mathrm{y}, \mathrm{xx}, 'OPTION')$$

โดยค่า x, y แทนเวกเตอร์ข้อมูลของ x และ y ในกรณี linear interpolation เวกเตอร์ x และ y จะต้องมีข้อมูลอย่างน้อย 2 ค่า เช่น x = [1 3] และ y = [2 6]

xx แทนจุดที่ต้องการทราบค่าประมาณ เป็นข้อมูลตัวเลขที่อยู่ในช่วงเวกเตอร์ x

yy แทนตัวแปรที่ใช้เก็บค่าที่ได้จากการประมาณค่าในช่วงตำแหน่ง xx

option คือตัวเลือกในการ interpolation ในกรณี linear interpolation ใช้‘linear’ หรือไม่มี OPTION ก็ได้

คำสั่ง interpl() ใน MATLAB คือ การประมาณค่าในช่วง โดยรับ input เพียงค่าเดียว

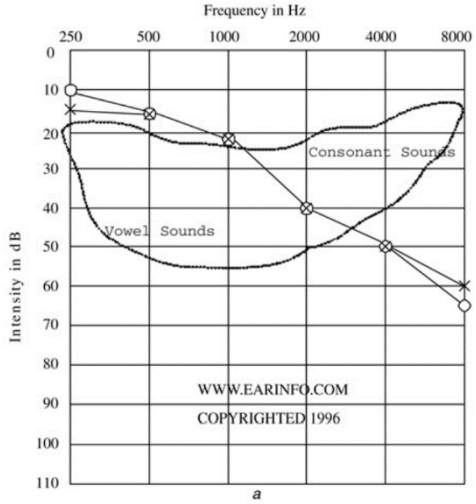
กรณีศึกษาที่นำมาใช้ในการทดลองการปรับใช้กับทฤษฎีการประมาณค่าในช่วง (Interpolation) กับ กราฟการได้ยิน (Audiogram) ใช้สัญลักษณ์ ‘X’ แทนหูซ้าย และ ‘O’ แทนหูขวา และใช้ช่วงความถี่ (Octave Frequency) ในการทดสอบ โดยความถี่ดังกล่าวอ้างอิงจาก [28] และรูปที่ 2.6 ซึ่งเป็นบทความที่ใช้ในการศึกษาการออกแบบวงจรกรองความถี่แบบดิจิทัล และบทความทางการแพทย์เกี่ยวกับการทดสอบการได้ยิน [8]

[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

โดยในการทดสอบนี้เลือกกราฟการได้ยินต้นแบบมาใช้ทดสอบนั้นมีทั้งหมด 9 ตัวอย่างดังนี้

4.5.1.1 ตัวอย่างกราฟการได้ยินที่ 1

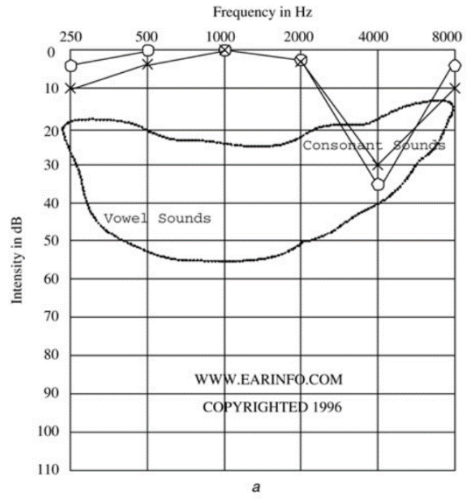
กราฟการได้ยินที่ 1 เป็นการสูญเสียทางการได้ยินที่เกิดขึ้นจากอายุที่เพิ่มขึ้น มีการสูญเสียการได้ยินเมื่ออยู่ในย่านความถี่ที่สูงขึ้นแสดงตามรูปที่ 4.20



รูปที่ 4.20 ตัวอย่างกราฟการได้ยินที่ 1 [28]

4.5.1.2 ตัวอย่างกราฟการได้ยินที่ 2

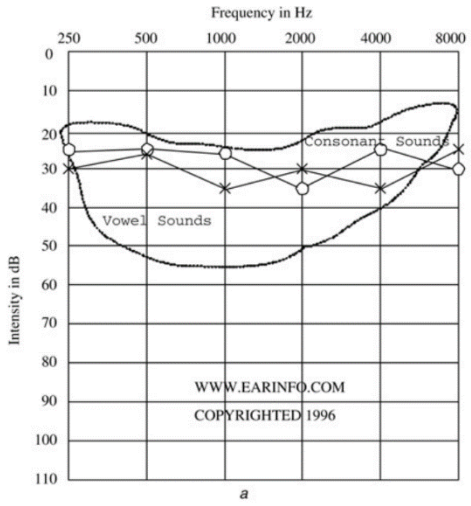
กราฟการได้ยินในตัวอย่างที่2 แสดงการสูญเสียทางการได้ยินอยู่ที่บริเวณ 4000 เฮิร์ตซ์ ซึ่งผู้ที่มีการสูญเสียการได้ยินประเภทนี้มักจะไม่สามารได้ยินเสียงพยัญชนะนี้ชัดเจน เนื่องจากผมกระทบจากสภาวะแวดล้อม



รูปที่ 4.21 ตัวอย่างกราฟการได้ยินที่ 2 [28]

4.5.1.3 ตัวอย่างกราฟการได้ยินที่ 3

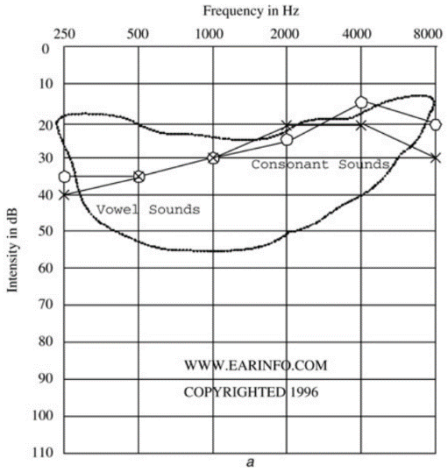
กราฟการได้ยินในตัวอย่างที่ 3 แสดงการสูญเสียทางการได้ยินในสภาวะหูตึงเล็กน้อย (Mild hearing loss) ตลอดช่วงแถบความถี่ ผู้ที่มีการสูญเสียทางการได้ยินในลักษณะนี้จะสามารถได้ยินเสียสระและพยัญชนะได้ไม่ชัดเจน



รูปที่ 4.22 ตัวอย่างกราฟการได้ยินที่ 3 [28]

4.5.1.4 ตัวอย่างกราฟการได้ยินที่ 4

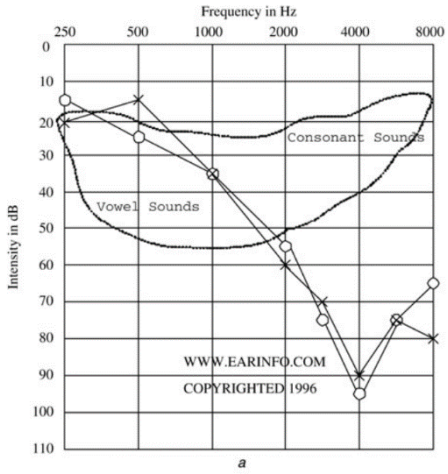
กราฟการได้ยินในตัวอย่างที่ 4 แสดงการสูญเสียทางการสูญเสียทางการได้ยินในสภาวะหูตึงเล็กน้อยไปจนถึงหูตึงปานกลางที่ช่วงความถี่ต่ำ และสภาวะหูตึงเล็กน้อยในช่วงความถี่สูง จึงทำให้ไม่สามารถได้ยินเสียงสระเนื่องจากเสียงที่ดัง ส่งผลให้การสนทนาในระยะใกล้นั้นเป็นเรื่องจำเป็น



รูปที่ 4.23 ตัวอย่างกราฟการได้ยินที่ 4 [28]

4.5.1.5 ตัวอย่างกราฟการได้ยินที่ 5

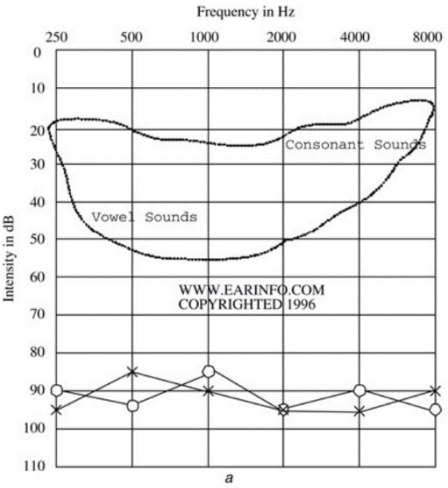
กราฟการได้ยินในตัวอย่างที่ 5 แสดงการสูญเสียทางการได้ยินในสภาวะหูตึงปานกลางไปจนถึงหูหนวกในช่วงความถี่ปานกลางไปจนถึงช่วงความถี่สูง ซึ่งเป็นรูปแบบการสูญเสียการได้ยินที่พบได้มากคนงานสูงอายุในโรงงานอุตสาหกรรม



รูปที่ 4.24 ตัวอย่างกราฟการได้ยินที่ 5 [28]

4.5.1.6 ตัวอย่างกราฟการได้ยินที่ 6

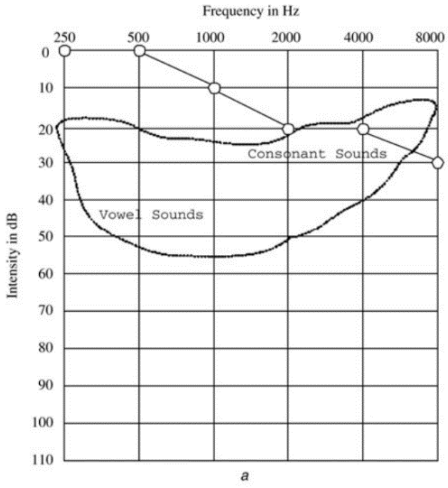
กราฟการได้ยินในตัวอย่างที่ 6 แสดงการสูญเสียทางการได้ยินในสภาวะหูตึงอย่างรุนแรงและหูหนวกตลอดช่วงความถี่ซึ่ง hearing thresholds ส่วนมากจะอยู่ที่ 90 เดซิเบล



รูปที่ 4.25 ตัวอย่างกราฟการได้ยินที่ 6 [28]

4.5.1.7 ตัวอย่างกราฟการได้ยินที่ 7

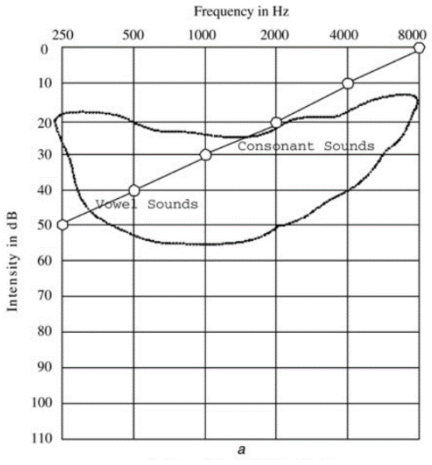
กราฟการได้ยินในตัวอย่างที่ 7 แสดงการสูญเสียทางการได้ยินในสภาวะหูตึงเล็กน้อยในช่วงความถี่สูงซึ่งจะเรียกว่า การสูญเสียแบบ ski slope



รูปที่ 4.26 ตัวอย่างกราฟการได้ยินที่ 7 [28]

4.5.1.8 ตัวอย่างกราฟการได้ยินที่ 8

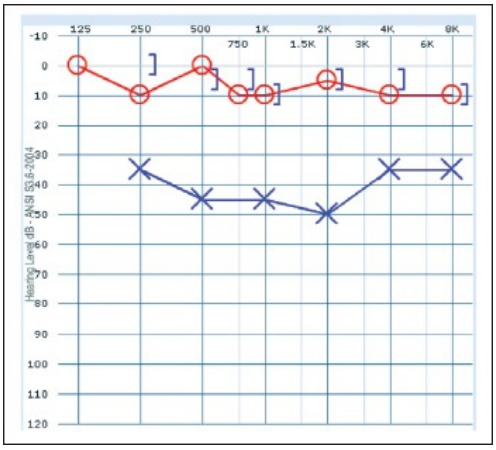
กราฟการได้ยินในตัวอย่างที่ 8 แสดงการสูญเสียงทางการได้ยินที่ลดลงตามช่วงความถี่



รูปที่ 4.27 ตัวอย่างกราฟการได้ยินที่ 8 [28]

4.5.1.9 ตัวอย่างกราฟการได้ยินที่ 9

กราฟการได้ยินในตัวอย่างที่ 9 แสดงการสูญเสียทางการได้ยินของเด็กอายุ 6 ปี ซึ่งสูญเสียทางการได้ยินตั้งแต่กำเนิด (Congenital hearing loss) ที่บริเวณของหูซ้าย เนื่องจากอยู่ในสภาพแวดล้อมที่เป็นมลพิษทางเสียง



รูปที่ 4.28 ตัวอย่างกราฟการได้ยินที่ 9 [38]

4.5.2 ผลลัพธ์ที่ได้จากกราฟการได้ยินที่นำมาใช้ทดสอบการออกแบบวงจรกรองความถี่แบบดิจิทัลสำหรับเครื่องช่วยฟัง

สำหรับผลการทดสอบการออกแบบวงจรกรองความถี่แบบดิจิทัลกับกราฟการได้ยินต้นแบบจะแสดงค่าสัมประสิทธิ์เทอมเศษ (Numerator), ค่าสัมประสิทธิ์เทอมส่วน (Denominator) และค่าเกน (Gain) ที่ให้ค่าความผิดพลาดที่น้อยที่สุดจากการคำนวณด้วย Mean Absolute Error โดยจะทำการทดสอบวงจรกรองความถี่กับกราฟการได้ยินต้นแบบที่ได้กล่าวไว้ข้างต้นทั้งหมด 5 ครั้ง ซึ่งผลการทดสอบที่ได้เป็นดังนี้

4.5.2.1 ผลการทดสอบจากตัวอย่างกราฟการได้ยินที่ 1

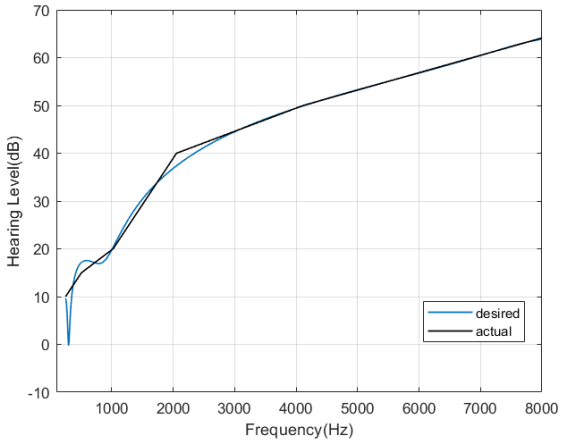
[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

Hearing Threshold = [10 15 20 40 50 65]

ตารางที่ 4.6 ผลการทดสอบจากกราฟการได้ยินที่ 1 ด้วย Mean Absolute Error

f_{LPF} (Hz)	Q_{LPF}	G_{LPF}	f_{BPF} (Hz)	Q_{BPF}	G_{BPF}	f_{HPF} (Hz)	Q_{HPF}	G_{HPF}	Error
250	0.1	5	4000	0.4	200	8000	0.4	800	0.4818
250	0.1	10	4000	0.8	200	8000	0.8	800	0.5431
250	0.1	1	6000	0.1	400	8000	0.2	700	0.6181
4000	0.1	1	6000	0.2	100	8000	0.3	70	0.7305
4000	0.5	1	6000	0.1	40	8000	0.2	70	1.0464

ผลการการทดสอบจากกราฟการได้ยินที่ 1 ที่ให้ค่าความผิดพลาดน้อยที่สุดเป็นไปตามรูปที่ 4.29



รูปที่ 4.29 ผลการการทดสอบจากกราฟการได้ยินที่ 1 ที่มี Error = 0.4818 dB

โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมตัวส่วน และค่าเกน (G) ของรูปที่ 4.29 มีค่าดังนี้

ตารางที่ 4.7 ค่าพารามิเตอร์ที่ได้จากผลการการทดสอบจากกราฟการได้ยินที่ 1

ชนิดของ filter	H(z)	ค่าเกน (G)
Lowpass Filter	$H(z) = \frac{0.0021 + 0.0042z^{-1} + 0.0021z^{-2}}{1 - 1.456z^{-1} + 0.4737z^{-2}}$	3.0287
Bandpass Filter	$H(z) = \frac{0.5479 + 0 - 0.5479z^{-2}}{1 + 0.0979z^{-1} - 0.096z^{-2}}$	111.3103
Highpass Filter	$H(z) = \frac{-0.256 + 05124z^{-1} - 0.256z^{-2}}{1 + 0.3891z^{-1} - 1.636z^{-2}}$	1.5996e+03

4.5.2.2 ผลการทดสอบจากตัวอย่างกราฟการได้ยินที่ 2

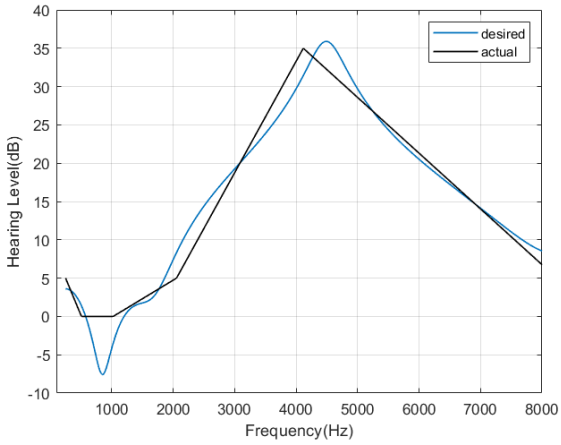
[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

Hearing Threshold = [5 0 0 5 35 5]

ตารางที่ 4.8 ผลการทดสอบจากกราฟการได้ยินที่ 2 ด้วย Mean Absolute Error

f_{LPF} (Hz)	Q_{LPF}	G_{LPF}	f_{BPF} (Hz)	Q_{BPF}	G_{BPF}	f_{HPF} (Hz)	Q_{HPF}	G_{HPF}	Error
2000	0.1	10	4000	0.1	10	8000	0.1	10	1.3966
2000	0.1	1	2000	0.3	50	2000	0.1	1	1.5048
2000	0.5	5	4000	0.6	10	8000	0.7	1	1.7322
2000	0.1	10	5000	0.5	10	8000	0.5	15	1.8244
3000	0.2	10	6000	0.3	10	8000	0.4	10	1.9360

ผลการการทดสอบจากกราฟการได้ยินที่ 2 ที่ให้ค่าความผิดพลาดน้อยที่สุดเป็นไปตามรูปที่ 4.30



รูปที่ 4.30 ผลการการทดสอบจากกราฟการได้ยินที 2 ทีมี Error = 1.3966 dB

โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมตัวส่วน และค่าเกน (G) ของรูปที่ 4.30 มีค่าดังนี้

ตารางที่ 4.9 ค่าพารามิเตอร์ที่ได้จากผลการการทดสอบจากกราฟการได้ยินที่ 2

ชนิดของ filter	$H(z)$	ค่าเกน (G)
Lowpass Filter	$H(z) = \frac{0.0296 + 0.0593z^{-1} + 0.0296z^{-2}}{1 - 1.274z^{-1} + 0.3924z^{-2}}$	1.5148
Bandpass Filter	$H(z) = \frac{0.1001 + 0 - 0.1001z^{-2}}{1 + 0.1673z^{-1} - 0.798z^{-2}}$	59.9160
Highpass Filter	$H(z) = \frac{1.3655 - 2.731z^{-1} + 1.3655z^{-2}}{1 - 2.468z^{-1} + 1.9943z^{-2}}$	2.5517

4.5.2.3 ผลการทดสอบจากตัวอย่างกราฟการได้ยินที่ 3

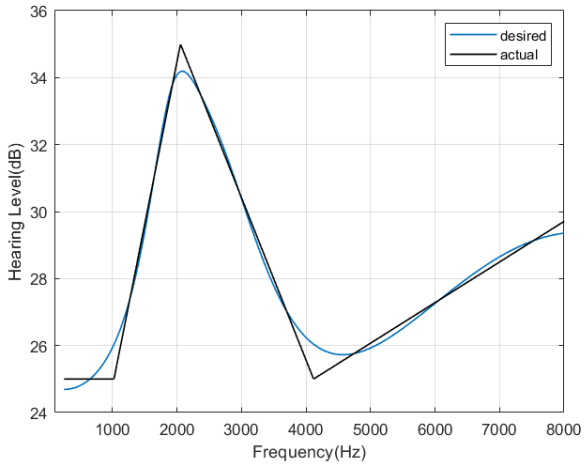
[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

Hearing Threshold = [25 25 25 35 25 30]

ตารางที่ 4.10 ผลการทดสอบจากกราฟการได้ยินที่ 3 ด้วย Mean Absolute Error

f_{LPF} (Hz)	Q_{LPF}	G_{LPF}	f_{BPF} (Hz)	Q_{BPF}	G_{BPF}	f_{HPF} (Hz)	Q_{HPF}	G_{HPF}	Error
2000	0.1	10	4000	0.7	30	8000	0.7	20	0.2168
2000	0.1	10	4000	0.8	30	8000	0.7	20	0.2276
2000	0.1	10	4000	0.6	30	8000	0.7	20	0.4024
1750	0.6	20	4000	0.7	20	8000	0.7	10	0.6641
2000	0.8	20	4000	0.8	10	8000	0.8	10	0.6643

ผลการการทดสอบจากกราฟการได้ยินที่ 3 ที่ให้ค่าความผิดพลาดน้อยที่สุดเป็นไปตามรูปที่ 4.31



รูปที่ 4.31 ผลการการทดสอบจากกราฟการได้ยินที่ 3 ที่มี Error = 0.2168 dB

โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมตัวส่วน และค่าเกน (G) ของรูปที่ 4.31 มีค่าดังนี้

ตารางที่ 4.11 ค่าพารามิเตอร์ที่ได้จากผลการการทดสอบจากกราฟการได้ยินที่ 3

ชนิดของ filter	$H(z)$	ค่าเกน (G)
Lowpass Filter	$H(z) = \frac{0.164 + 0.328z^{-1} + 0.164z^{-2}}{1 - 0.865z^{-1} + 0.5212z^{-2}}$	17.1587
Bandpass Filter	$H(z) = \frac{0.1387 + 0 - 0.1387z^{-2}}{1 - 1.343z^{-1} + 0.7225z^{-2}}$	32.1654
Highpass Filter	$H(z) = \frac{4.5978 - 9.196z^{-1} + 4.5977z^{-2}}{1 + 9.465z^{-1} + 26.856z^{-2}}$	29.4565

4.5.2.4 ผลการทดสอบจากตัวอย่างกราฟการได้ยินที่ 4

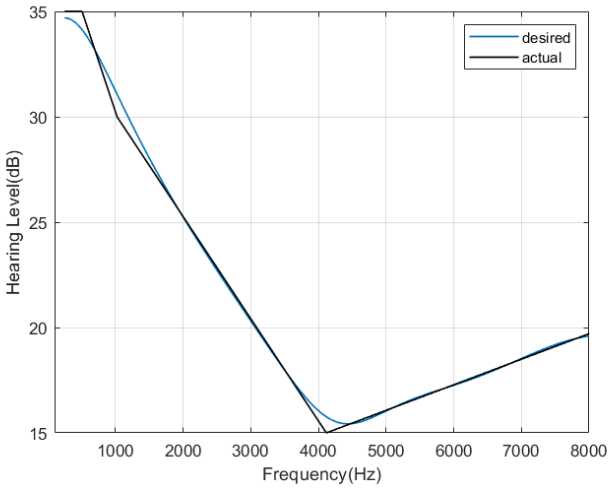
[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

Hearing Threshold = [35 35 30 25 15 20]

ตารางที่ 4.12 ผลการทดสอบจากกราฟการได้ยินที่ 4 ด้วย Mean Absolute Error

f_{LPF} (Hz)	Q_{LPF}	G_{LPF}	f_{BPF} (Hz)	Q_{BPF}	G_{BPF}	f_{HPF} (Hz)	Q_{HPF}	G_{HPF}	Error
2000	0.7	30	4000	0.6	10	8000	0.5	10	0.1760
1750	0.6	30	4000	0.7	20	8000	0.7	10	0.1947
2000	0.7	20	4000	0.7	10	8000	0.7	10	0.2102
2000	0.5	20	4000	0.8	10	8000	0.8	10	0.2440
2000	0.7	400	40000	0.6	10	8000	0.5	10	0.2486

ผลการการทดสอบจากกราฟการได้ยินที่ 4 ที่ให้ค่าความผิดพลาดน้อยที่สุดเป็นไปตามรูปที่ 4.32



รูปที่ 4.32 ผลการการทดสอบจากกราฟการได้ยินที่ 4 ที่มี Error = 0.1760 dB

โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมตัวส่วน และค่าเกน (G) ของรูปที่ 4.32 มีค่าดังนี้

ตารางที่ 4.13 ค่าพารามิเตอร์ที่ได้จากผลการการทดสอบจากกราฟการได้ยินที่ 4

ชนิดของ filter	H(z)	ค่าเกน (G)
Lowpass Filter	$H(z) = \frac{0.0519 + 0.1039z^{-1} + 0.0519z^{-2}}{1 - 0.911z^{-1} + 0.1192z^{-2}}$	54.2469
Bandpass Filter	$H(z) = \frac{0.3475 + 0 - 0.3475z^{-2}}{1 + 0.2035z^{-1} + 0.305z^{-2}}$	2.049
Highpass Filter	$H(z) = \frac{-0.813 + 1.626z^{-1} - 0.813z^{-2}}{1 + 0.8265z^{-1} - 3.426z^{-2}}$	9.6029

4.5.2.5 ผลการทดสอบจากตัวอย่างกราฟการได้ยินที่ 5

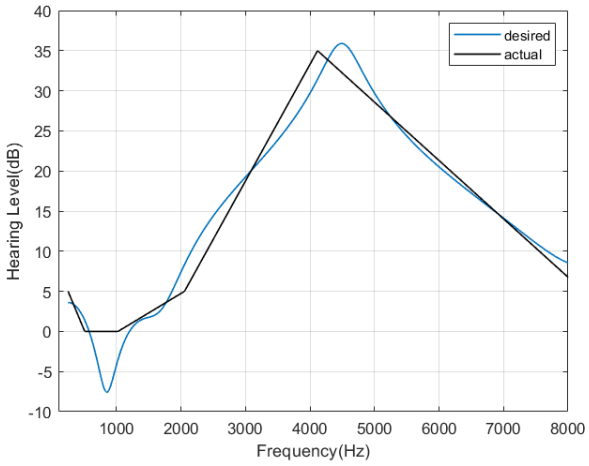
[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 3000 4000 6000 8000]

Hearing Threshold = [15 25 35 55 75 95 75 65]

ตารางที่ 4.14 ผลการทดสอบจากกราฟการได้ยินที่ 5 ด้วย Mean Absolute Error

f_{LPF} (Hz)	Q_{LPF}	G_{LPF}	f_{BPF} (Hz)	Q_{BPF}	G_{BPF}	f_{HPF} (Hz)	Q_{HPF}	G_{HPF}	Error
550	0.5	500	3750	0.9	7500	8000	0.7	5500	3.9234
550	0.5	10	3750	0.9	750	8000	0.7	150	3.9322
200	0.5	800	3250	3	8200	4100	10	5010	3.9455
250	0.7	80	3750	0.9	70	4000	0.3	70	4.0061
200	0.5	800	3250	2	8200	4100	10	5010	4.0283

ผลการการทดสอบจากกราฟการได้ยินที่ 5 ที่ให้ค่าความผิดพลาดน้อยที่สุดเป็นไปตามรูปที่ 4.33



รูปที่ 4.33 ผลการการทดสอบจากกราฟการได้ยินที่ 5 ที่มี Error = 3.9234 dB

โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมตัวส่วน และค่าเกน (G) ของรูปที่ 4.33 มีค่าดังนี้

ตารางที่ 4.15 ค่าพารามิเตอร์ที่ได้จากผลการการทดสอบจากกราฟการได้ยินที่ 5

ชนิดของ filter	$H(z)$	ค่าเกน (G)
Lowpass Filter	$H(z) = \frac{0.0393 + 0.0786z^{-1} + 0.0393z^{-2}}{1 - 1.572z^{-1} + 0.7297z^{-2}}$	79.1612
Bandpass Filter	$H(z) = \frac{0.059 + 0 - 0.059z^{-2}}{1 + 0.0938z^{-1} + 0.882z^{-2}}$	3.8051e+04
Highpass Filter	$H(z) = \frac{10.887 - 21.78z^{-1} - 10.887z^{-2}}{1 - 21.14z^{-1} + 21.409z^{-2}}$	3.8654e+03

4.5.2.6 ผลการทดสอบจากตัวอย่างกราฟการได้ยินที่ 6

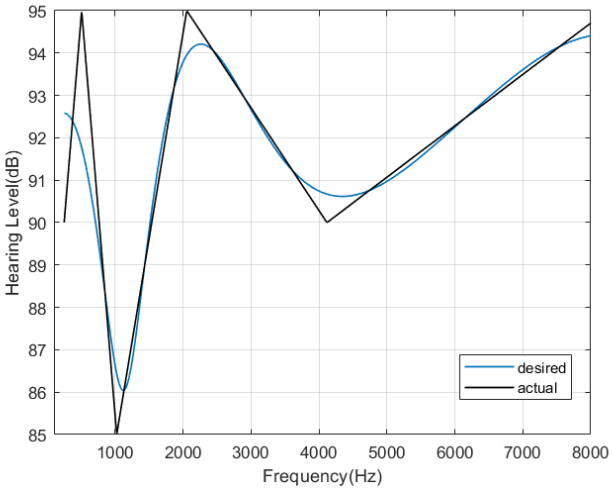
[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

Hearing Threshold = [90 95 85 95 90 95]

ตารางที่ 4.16 ผลการทดสอบจากกราฟการได้ยินที่ 6 ด้วย Mean Absolute Error

f_{LPF} (Hz)	Q_{LPF}	G_{LPF}	f_{BPF} (Hz)	Q_{BPF}	G_{BPF}	f_{HPF} (Hz)	Q_{HPF}	G_{HPF}	Error
1000	2.2	37000	3950	2	41500	8000	0.675	50000	0.2846
900	2.2	38000	3950	2	41500	8000	0.675	50000	0.3561
1000	0.3	100	5000	0.1	100	5000	0.1	100	1.1891
1000	0.2	100	5000	0.1	100	5000	0.1	100	1.4095
250	0.2	40	2000	0.1	40	2000	0.1	40	1.4402

ผลการการทดสอบจากกราฟการได้ยินที่ 6 ที่ให้ค่าความผิดพลาดน้อยที่สุดเป็นไปตามรูปที่ 4.34



รูปที่ 4.34 ผลการการทดสอบจากกราฟการได้ยินที่ 6 ที่มี Error = 0.2846 dB

โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมตัวส่วน และค่าเกน (G) ของรูปที่ 4.34 มีค่าดังนี้

ตารางที่ 4.17 ค่าพารามิเตอร์ที่ได้จากผลการการทดสอบจากกราฟการได้ยินที่ 6

ชนิดของ filter	H(z)	ค่าเกน (G)
Lowpass Filter	$H(z) = \frac{0.0314 + 0.0627z^{-1} + 0.0314z^{-2}}{1 - 1.305z^{-1} + 0.4307z^{-2}}$	4.2555e+04
Bandpass Filter	$H(z) = \frac{0.2623 + 0 - 0.2623z^{-2}}{1 - 1.09z^{-1} + 0.4755z^{-2}}$	5.7544e+04
Highpass Filter	$H(z) = \frac{2.2201 - 4.44z^{-1} + 2.2201z^{-2}}{1 + 9.5746z^{-1} + 17.455z^{-2}}$	5.2749e+04

4.5.2.7 ผลการทดสอบจากตัวอย่างกราฟการได้ยินที่ 7

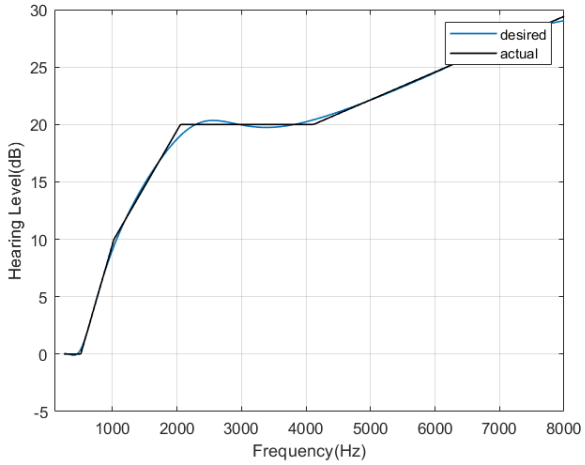
[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

Hearing Threshold = [0 0 10 20 20 30]

ตารางที่ 4.18 ผลการทดสอบจากกราฟการได้ยินที่ 7 ด้วย Mean Absolute Error

f_{LPF} (Hz)	Q_{LPF}	G_{LPF}	f_{BPF} (Hz)	Q_{BPF}	G_{BPF}	f_{HPF} (Hz)	Q_{HPF}	G_{HPF}	Error
2000	0.3	10	4000	0.9	10	8000	0.5	10	0.1689
2000	0.1	10	4000	0.7	15	8000	0.5	15	0.1701
2000	0.4	10	4000	0.5	15	8000	0.6	20	0.2632
2000	0.3	10	4000	0.9	10	8000	0.5	15	0.2889
1750	0.6	20	4000	0.7	20	8000	0.7	10	0.2919

ผลการการทดสอบจากกราฟการได้ยินที่ 7 ที่ให้ค่าความผิดพลาดน้อยที่สุดเป็นไปตามรูปที่ 4.35



รูปที่ 4.35 ผลการการทดสอบจากกราฟการได้ยินที่ 7 ที่มี Error = 0.1689 dB

โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมตัวส่วน และค่าเกน (G) ของรูปที่ 4.35 มีค่าดังนี้

ตารางที่ 4.19 ค่าพารามิเตอร์ที่ได้จากผลการการทดสอบจากกราฟการได้ยินที่ 7

ชนิดของ filter	H(z)	ค่าเกน (G)
Lowpass Filter	$H(z) = \frac{0.0710 + 0.1419z^{-1} + 0.0710z^{-2}}{1 - 0.157z^{-1} + 0.5596z^{-2}}$	1.0064
Bandpass Filter	$H(z) = \frac{0.2371 + 0 - 0.2371z^{-2}}{1 - 0.99z^{-1} + 0.5258z^{-2}}$	11.8053
Highpass Filter	$H(z) = \frac{-2.518 + 5.036z^{-1} - 2.518z^{-2}}{1 - 8.318z^{-1} - 19.39z^{-2}}$	28.593

4.5.2.8 ผลการทดสอบจากตัวอย่างกราฟการได้ยินที่ 8

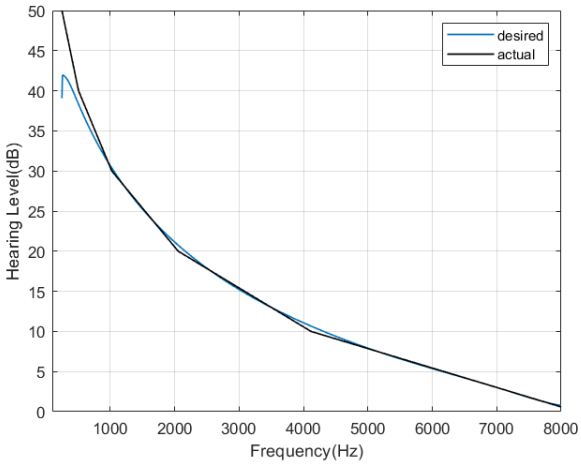
[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

Hearing Threshold = [50 40 30 20 10 0]

ตารางที่ 4.20 ผลการทดสอบจากกราฟการได้ยินที่ 8 ด้วย Mean Absolute Error

f_{LPF} (Hz)	Q_{LPF}	G_{LPF}	f_{BPF} (Hz)	Q_{BPF}	G_{BPF}	f_{HPF} (Hz)	Q_{HPF}	G_{HPF}	Error
250	0.4	40	4000	0.7	30	7000	0.7	5	0.3550
2000	0.7	10	4000	0.5	15	8000	0.7	10	0.6978
2000	0.7	10	4000	0.6	10	8000	0.5	10	0.8432
2000	0.7	20	4000	0.6	10	8000	0.5	10	0.9038
2000	0.3	10	4000	0.9	10	8000	0.5	10	1.0880

ผลการการทดสอบจากกราฟการได้ยินที่ 8 ที่ให้ค่าความผิดพลาดน้อยที่สุดเป็นไปตามรูปที่ 4.36



รูปที่ 4.36 ผลการการทดสอบจากกราฟการได้ยินที่ 8 ที่มี Error = 0.3550 dB

โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมตัวส่วน และค่าเกน (G) ของรูปที่ 4.36 มีค่าดังนี้

ตารางที่ 4.21 ค่าพารามิเตอร์ที่ได้จากผลการการทดสอบจากกราฟการได้ยินที่ 8

ชนิดของ filter	H(z)	ค่าเกน (G)
Lowpass Filter	$H(z) = \frac{0.0077 + 0.0153z^{-1} + 0.0077z^{-2}}{1 - 1.532z^{-1} + 0.563z^{-2}}$	89.8826
Bandpass Filter	$H(z) = \frac{0.0715 + 0 - 0.0715z^{-2}}{1 - 1.857z^{-1} + 0.857z^{-2}}$	35.5233
Highpass Filter	$H(z) = \frac{9.2306 - 18.46z^{-1} + 9.2306z^{-2}}{1 + 41.808z^{-1} + 77.73z^{-2}}$	1.0733

4.5.2.9 ผลการทดสอบจากตัวอย่างกราฟการได้ยินที่ 9

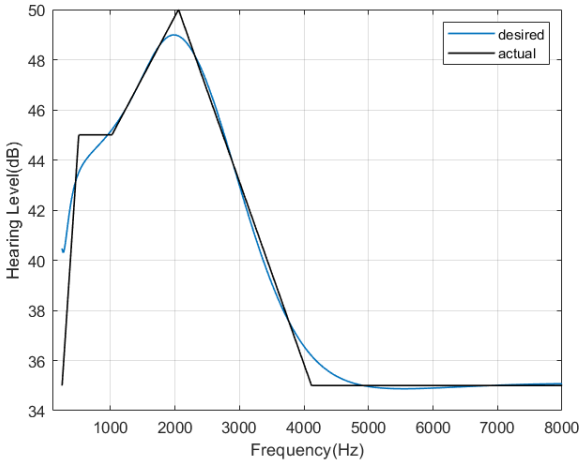
[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

Hearing Threshold = [35 45 45 50 35 35]

ตารางที่ 4.22 ผลการทดสอบจากกราฟการได้ยินที่ 9 ด้วย Mean Absolute Error

$\mathbf{f_{LPF}\ (Hz)}$	$\mathbf{Q_{LPF}}$	$\mathbf{G_{LPF}}$	$\mathbf{f_{BPF}\ (Hz)}$	$\mathbf{Q_{BPF}}$	$\mathbf{G_{BPF}}$	$\mathbf{f_{HPF}\ (Hz)}$	$\mathbf{Q_{HPF}}$	$\mathbf{G_{HPF}}$	**Error**
1010.12	0.71	150	3952	0.5	310	8000	0.3	200	0.2893
1010.12	0.7101	150	3952	0.5	310	8000	0.3	200	0.2995
1010.1	0.71	150	3952	0.5	310	8000	0.3	200	0.3039
1010	0.71	150	3952	0.5	310	8000	0.3	200	0.3055
1010	0.7	150	3952	0.5	310	8000	0.3	200	0.3108

ผลการการทดสอบจากกราฟการได้ยินที่ 9 ที่ให้ค่าความผิดพลาดน้อยที่สุดเป็นไปตามรูปที่ 4.37



รูปที่ 4.37 ผลการการทดสอบจากกราฟการได้ยินที่ 9 ที่มี Error = 0.2893 dB

โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมตัวส่วน และค่าเกน (G) ของรูปที่ 4.37 มีค่าดังนี้

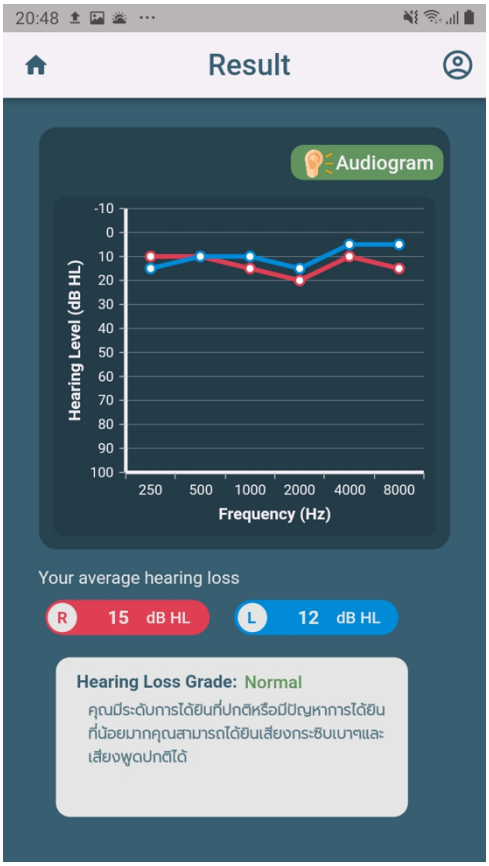
ตารางที่ 4.23 ค่าพารามิเตอร์ที่ได้จากผลการการทดสอบจากกราฟการได้ยินที่ 9

ชนิดของ filter	$\mathrm{H(z)}$	**ค่าเกน (G)**
Lowpass Filter	$\mathrm{H(z)} = \frac{0.14 + 0.28z^{-1} + 0.14z^{-2}}{1 - 0.944z^{-1} + 0.5037z^{-2}}$	105.5608
Bandpass Filter	$\mathrm{H(z)} = \frac{0.1933 + 0 - 0.1933z^{-2}}{1 - 1.269z^{-1} + 0.6135z^{-2}}$	122.4536
Highpass Filter	$\mathrm{H(z)} = \frac{0.9658 - 1.932z^{-1} + 0.9658z^{-2}}{1 - 1.932z^{-1} + 0.9318z^{-2}}$	56.7276

4.5.3 การนำค่าการทดสอบสรรถภาพการได้ยินจากฐานข้อมูลมาใช้ในการออกแบบวงจรกรองความถี่แบบดิจิทัล

[f1 f2 f3 f4 f5 f6] = [250 500 1000 2000 4000 8000]

Hearing Threshold = [15 10 10 15 5 5]



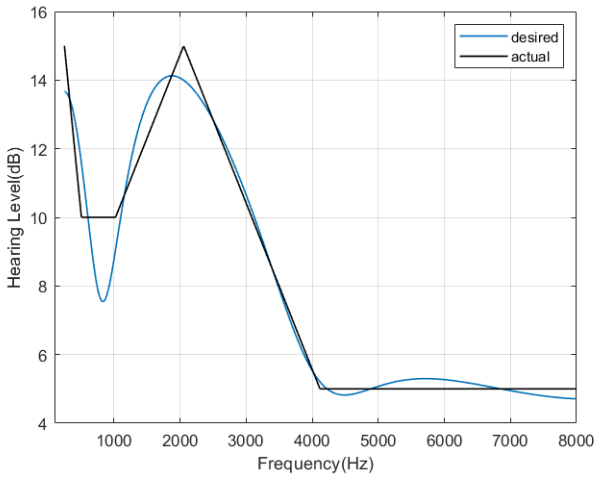
รูปที่ 4.38 ผลการทดสอบสมรรถภาพการได้ยินจากแอปพลิเคชันทดสอบการได้ยิน

จากการนำผลการทดสอบสมรรถภาพการได้ยินในฐานข้อมูลแสดงดังรูปที่ 4.38 เพื่อไปคำนวณหาค่าสัมประสิทธิ์เทอมเศษ สัมประสิทธิ์เทอมส่วน และค่าเกนของแต่ละวงจรกรองความถี่ในส่วนของเซิร์ฟเวอร์ โดยในที่นี้แสดงตัวอย่างการออกแบบวงจรกรองความถี่ของผู้ซ้ายของผู้ใช้งานซึ่งผลลัพธ์จากการคำนวณด้วยกระบวนการ Nelder-Mead algorithm จะมีค่าดังตารางที่ 4.24

ตารางที่ 4.24 ค่าสัมประสิทธิ์และค่าอัตราขยายของแต่ละวงจรกรองความถี่แบบดิจิทัล

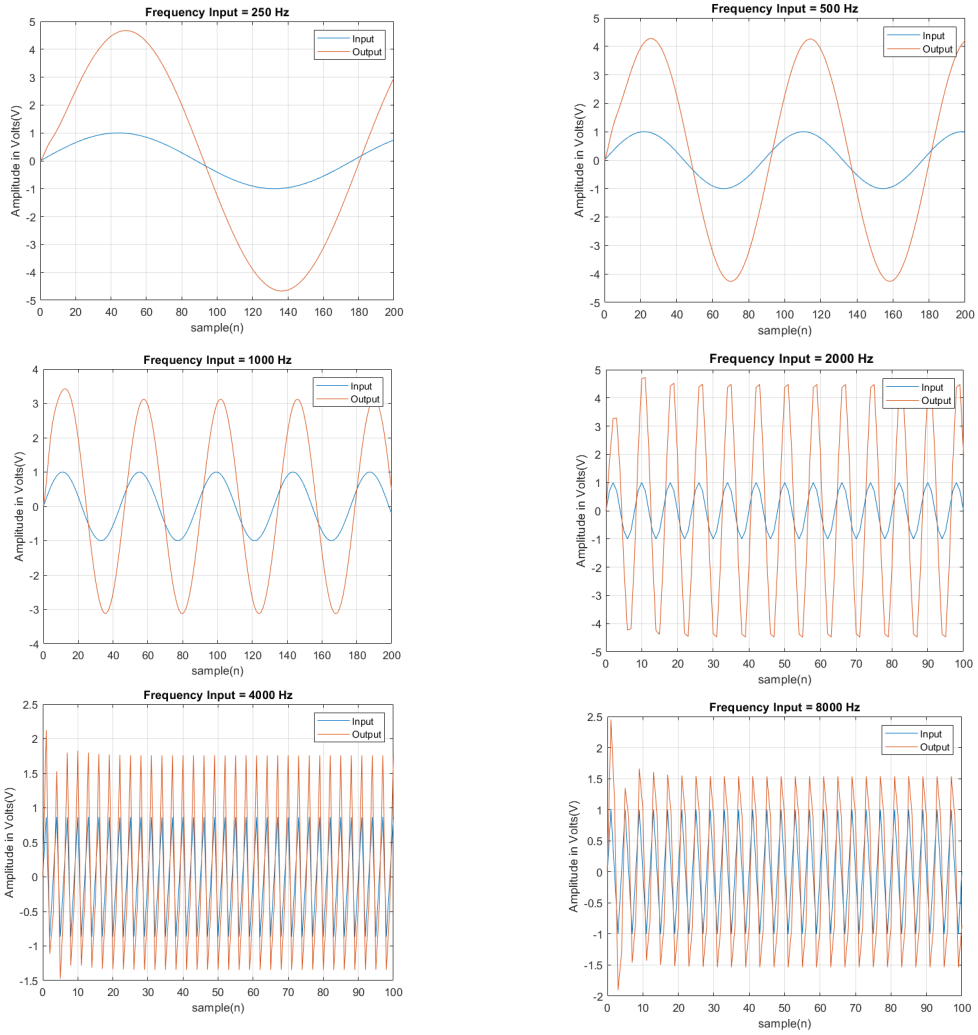
ชนิดของ filter	$H(z)$	**ค่าเกน (G)**
Lowpass Filter	$H(z) = \frac{0.01768 + 0.03537z^{-1} + 0.01768z^{-2}}{1 - 1.41905z^{-1} + 0.48981z^{-2}}$	4.82232
Bandpass Filter	$H(z) = \frac{0.30613 + 0 - 0.30613z^{-2}}{1 - 1.13436z^{-1} + 0.38772z^{-2}}$	5.73525
Highpass Filter	$H(z) = \frac{0.35512 - 0.71025z^{-1} + 0.35512z^{-2}}{1 - 0.16770z^{-1} + 0.25279z^{-2}}$	1.71755

จากตารางที่ 4.24 จะสามารถสร้างผลการตอบสนองทางความถี่ได้ เมื่อเทียบผลการตอบสนองทางความถี่ระหว่างวงจรกรองความถี่แบบดิจิทัลที่ออกแบบกับค่าการทดสอบสมรรถภาพการได้ยิน ซึ่งจะแสดงได้ดังรูปที่ 4.39



รูปที่ 4.39 ผลการตอบสนองทางความถี่ระหว่างวงจรกรองความถี่ที่ออกแบบกับค่าการทดสอบสมรรถภาพการได้ยิน

จากรูปที่ 4.39 จะเห็นได้ว่าวงจรกรองความถี่แบบดิจิทัลที่ออกแบบมีความใกล้เคียงกับผลการทดสอบสมรรถภาพการได้ยิน โดยมีค่าความแตกต่างโดยวัดจากการคำนวณด้วยหลักการ Mean Absolute Error ซึ่งจะมีค่าความแตกต่างอยู่ที่ 0.350659 ซึ่งเมื่อนำสัญญาณตัวอย่างในความถี่ต่าง ๆ มาผ่านวงจรกรองความถี่แบบดิจิทัลที่ได้ออกแบบ โดยสัญญาณตัวอย่างคือสัญญาณ Sine wave ขนาด 1 โวลต์ ที่ความถี่ Octave จะมีลักษณะเปลี่ยนแปลงตามระดับ Amplitude ของวงจรกรองความถี่แบบดิจิทัล และสามารถแสดงผลการทดสอบได้ดังรูปที่ 4.40

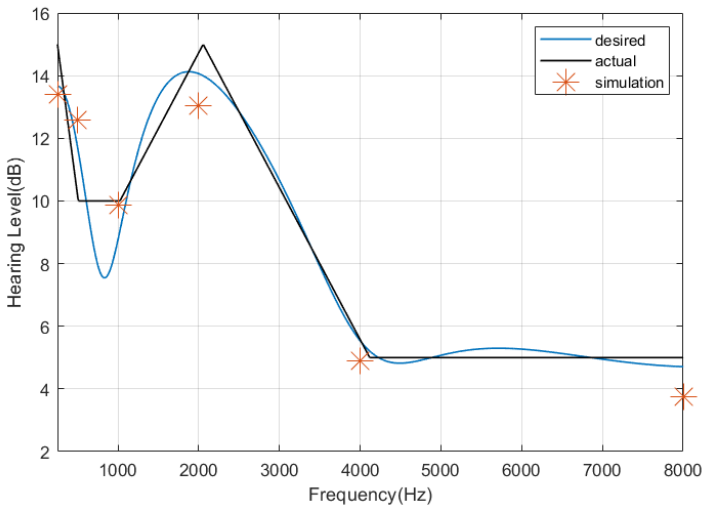


รูปที่ 4.40 ผลการทดสอบวงจรกรองความถี่แบบดิจิทัลที่ออกแบบกับสัญญาณตัวอย่าง

จากรูปที่ 4.40 จะเห็นได้ว่าวงจรกรองความถี่แบบดิจิทัลสามารถทำให้สัญญาณในความถี่ Octave มี Amplitude เปลี่ยนไปตามลักษณะของวงจรกรองความถี่นั้น ๆ ซึ่งจะทำให้สามารถชดเชยการสูญเสียสมรรถภาพการได้ยินตามผลการทดสอบสมรรถภาพการได้ยินของผู้ใช้งาน

ตารางที่ 4.25 ค่าอัตราขยายที่แต่ละความถี่ของการชดเชยการสูญเสียการได้ยินของรูปที่ 4.39

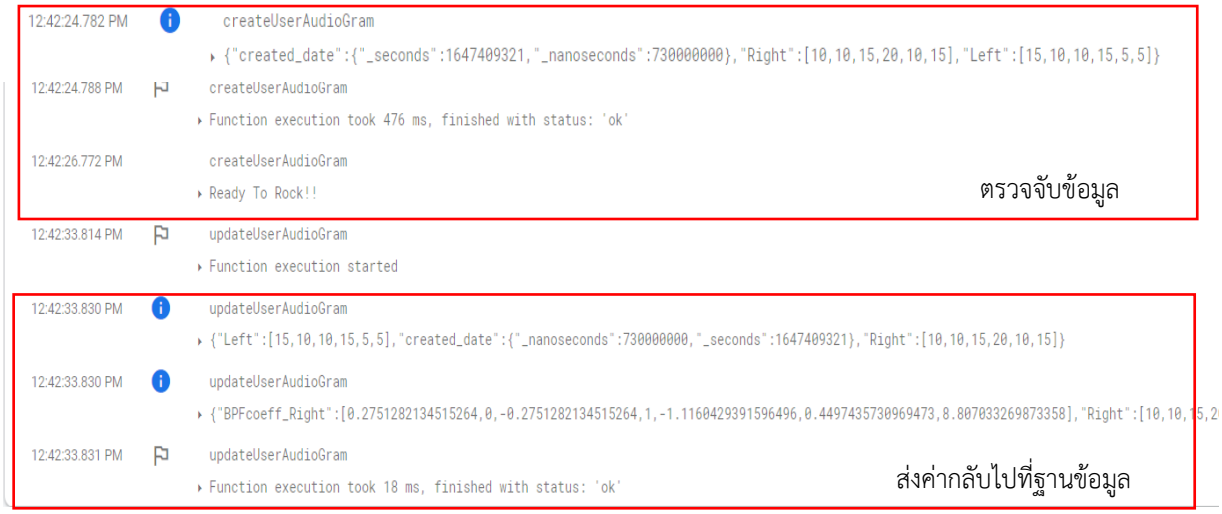
ความถี่ (Hz)	250	500	1000	2000	4000	8000
Input (volt)	1	1	1	1	1	1
Output (volt)	4.669	4.260	3.120	4.479	1.759	1.538
อัตราขยาย (เท่า)	4.669	4.260	3.120	4.479	1.759	1.538
อัตราขยาย (dB)	13.3845	12.5882	9.88309	13.0236	4.90532	3.73913



รูปที่ 4.41 ผลการจำลองที่ได้จากการออกแบบวงจรกรองสัญญาณวัดที่ความถี่ Octave การชดเชยการสูญเสียการได้ยินของรูปที่ 4.39 เปรียบเทียบกับวงจรกรองที่ต้องการ

4.6 การทดสอบคลาวด์เซิร์ฟเวอร์ที่ใช้สำหรับการออกแบบวงจรกรองความถี่

ในกระบวนการถัดไปเป็นกระบวนการออกแบบวงกรองความถี่แบบดิจิทัลที่อยู่ในฝั่งของเซิร์ฟเวอร์ โดยใช้ผลที่ได้จากการทดสอบสมรรถภาพการได้ยินจากแอปพลิเคชันสำหรับการทดสอบการได้ยิน ซึ่งผลดังกล่าวจะถูกนำไปเก็บไว้ที่ฐานข้อมูลอย่าง Firebase เพื่อนำไปประมวลผลต่อในส่วนของเซิร์ฟเวอร์ตามที่ได้ออกแบบไว้ในหัวข้อที่ 3.1.7 โดยระบบจะตรวจจับผลที่ได้จากการทดสอบสมรรถภาพการได้ยินแบบตามเวลาจริง และนำค่าที่ได้ไปประมวลผลต่อที่เซิร์ฟเวอร์ เพื่อนำไปใช้งานต่อในฝั่งของแอปพลิเคชันเครื่องช่วยฟัง โดยผลลัพธ์ที่ได้เป็นไปตามรูปที่ 4.42

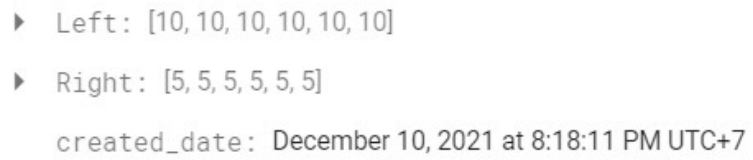


รูปที่ 4.42 Logs การทำงานของ Cloud Function

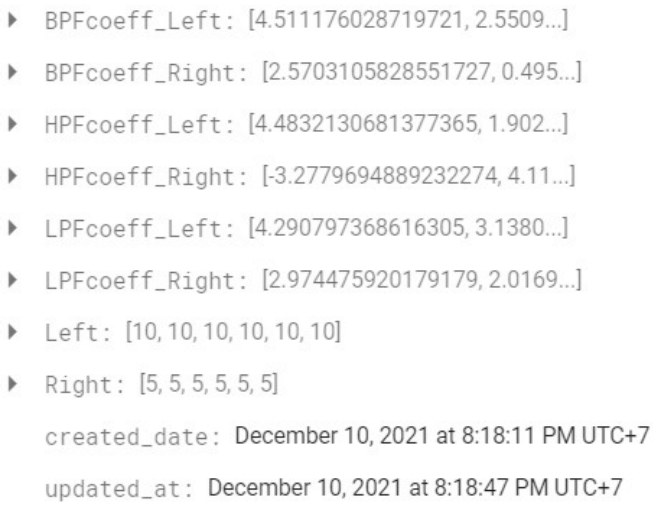
```
Warning: Maximum number of function evaluations has been exceeded.
Current function value:  0.3506585077573114
Iterations:  1247
Function evaluations:  1800
Coefficients:  [2.17184422e+03 4.44619161e-01 4.82232165e+00 4.30855200e+03
 6.52781091e-01 5.73525635e+00 1.00825939e+04 8.30782183e-01
 1.71755967e+00]
Warning: Maximum number of function evaluations has been exceeded.
Current function value:  0.24239944336828298
Iterations:  1259
Function evaluations:  1801
Coefficients:  [8.39507090e+03 3.51143985e-01 3.15379121e+00 4.85861817e+03
 8.40801844e-01 8.80703327e+00 7.55243736e+03 4.90912184e-01
 5.06507131e+00]
```

รูปที่ 4.43 ผลการทำงานของเซิร์ฟเวอร์

จากรูปที่ 4.42 จะเห็นว่าเซิร์ฟเวอร์ได้รับค่าจากระบบที่ตรวจจับผลการทดสอบสมรรถภาพการได้ยินและนำไปคำนวณหาค่าสัมประสิทธิ์สำหรับวงจรกรองความถี่แบบดิจิทัล และส่งกลับไปที่ฐานข้อมูลหรือ Firebase เพื่อนำไปใช้ในแอปพลิเคชันเครื่องช่วยฟังได้ ซึ่งรูปที่ 4.43 จะแสดงผลการทำงานของคลาวด์เซิร์ฟเวอร์บน Command Line Interface ของ DIgitalOcean



รูปที่ 4.44 ฐานข้อมูลก่อนส่งค่าไปคำนวณที่เซิร์ฟเวอร์



รูปที่ 4.45 ค่าที่เซิร์ฟเวอร์ส่งกลับมาฐานข้อมูลหลังประมวลผล

4.6.1 การวัดเวลาที่ใช้ทั้งระบบบนแอปพลิเคชันเครื่องช่วยฟัง

หลังจากทำการติดตั้งโปรแกรมที่ใช้สำหรับคำนวณค่าสัมประสิทธิ์ที่ใช้สำหรับออกแบบวงจรกรองความถี่แบบดิจิทัลเสร็จเป็นที่เรียบร้อย ปริญญานิพนธ์นี้ได้ทำการทดสอบประสิทธิภาพในการประมวลผลของระบบทั้งหมดตั้งแต่ผู้ใช้งานเริ่มส่งค่าการทดสอบสมรรถภาพการได้ยินจากโทรศัพท์มือถือ ค่าการทดสอบสมรรถภาพการได้ยินถูกส่งมายัง Firebase โดยมีการใช้บริการ Cloud Function เพื่อตรวจจับเวลาของข้อมูลที่เข้ามา และ บริการ MQTT สำหรับการป้องกันการชนกันของข้อมูลที่เข้ามา จากนั้นข้อมูลดังกล่าวถูกส่งไปยังคลาวด์เซิร์ฟเวอร์เพื่อใช้สำหรับการประมวลผล และส่งค่าที่ได้จากการคำนวณกลับมายัง Firebase เพื่อให้โทรศัพท์มือถือสามารถนำค่าสัมประสิทธิ์ที่ใช้สำหรับออกแบบวงจรกรองความถี่แบบดิจิทัลมาชดเชยการสูญเสียการได้ยินบนแอปพลิเคชันเครื่องช่วยฟังได้

โดยการวัดเวลาที่ใช้ของระบบการทำงานทั้งหมดบนแอปพลิเคชันเครื่องช่วยฟังจะใช้ timestamp ของข้อมูลค่าการทดสอบสมรรถภาพการได้ยินจากเครื่องโทรศัพท์ที่เข้ามายัง Firebase และ timestamp ของข้อมูลค่าสัมประสิทธิ์ที่ใช้สำหรับออกแบบวงจรกรองความถี่แบบดิจิทัลจากคลาวด์เซิร์ฟเวอร์ที่เข้ามายัง Firebase ในหน่วยวินาที โดยมีตัวอย่างข้อมูลที่ใช้ในการวัดเวลาที่ใช้ทั้งระบบบนแอปพลิเคชันเครื่องช่วยฟังแสดงดังรูปที่ 4.46

created_date: 8 February 2022 at 16:22:45 UTC+7

updated_at: 8 February 2022 at 16:22:56 UTC+7

รูปที่ 4.46 ตัวอย่างเวลาที่ใช้ในการวัดประสิทธิภาพของระบบการทำงานบนแอปพลิเคชันเครื่องช่วยฟัง

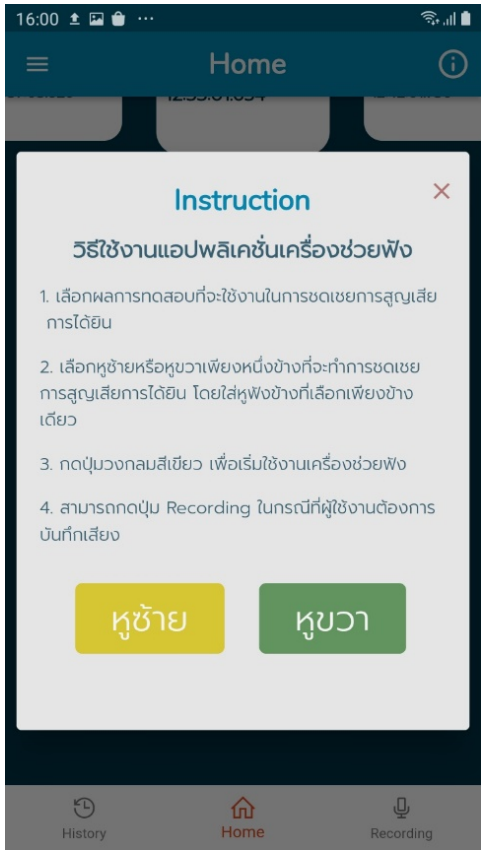
ซึ่งในการวัดเวลาที่ใช้ของระบบการทำงานทั้งหมดบนแอปพลิเคชันเครื่องช่วยฟังจะทำการทดสอบโดยการส่งค่าสมรรถภาพการได้ยินจากเครื่องโทรศัพท์มือถือทั้งหมด 5 ครั้ง และนำเวลาที่ได้มาหาค่าเฉลี่ย ซึ่งจากการทดสอบเวลาที่ได้อยู่ที่ประมาณ 12.6 วินาที ผลการทดสอบที่ได้ทั้ง 5 ครั้งเป็นไปตามตารางที่ 4.26

ตารางที่ 4.26 เวลาที่ใช้ของระบบการทำงานทั้งหมดบนแอปพลิเคชันเครื่องช่วยฟังจำนวน 5 ครั้ง

ครั้งที่	**เวลาที่ใช้ในการประมวลผลทั้งหมด** (วินาที) (จากผู้ใช้งานไปยังคลาวด์เซิร์ฟเวอร์และกลับมายังผู้ใช้งาน)
1	12
2	12
3	13
4	14
5	12
ค่าเฉลี่ย	**12.6**

4.7 ผลลัพธ์ที่ได้จากการใช้งานแอปพลิเคชันเครื่องช่วยฟัง

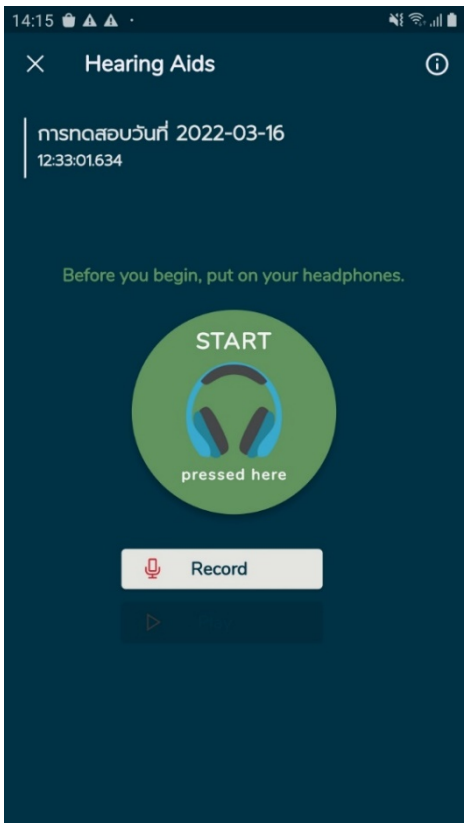
ในการเก็บผลการทดลองจากการใช้งานแอปพลิเคชันเครื่องช่วยฟัง โดยแอปพลิเคชันจะทำการนำข้อมูลของสัมประสิทธิ์ของวงจรกรองความถี่ต่ำ สัมประสิทธิ์ของวงจรกรองความถี่แถบผ่าน และสัมประสิทธิ์ของวงจรกรองความถี่สูง รวมทั้งค่าอัตราขยายของแต่ละวงจรกรองความถี่ ของแต่ละข้างจะถูกนำไปใช้กับสมการผลต่างสืบเนื่อง เพื่อใช้ในการชดเชยการสูญเสียสมรรถภาพการได้ยิน โดยในการเก็บผลการทดลองจะเก็บผลที่หูข้างซ้ายและมีค่าการสูญเสียสมรรถภาพการได้ยินที่ 15, 10, 10, 15, 5 และ 5 ที่ความถี่ 250, 500, 1000, 2000, 4000 และ 8000 ตามลำดับ โดยผู้ใช้งานจะต้องเลือกค่าการสูญเสียและข้างของหูให้ถูกต้องเพื่อทำการชดเชยการสูญเสีย ดังรูปที่ 4.47



รูปที่ 4.47 หน้าแสดงสำหรับเลือกข้างของหูที่ต้องการชดเชยการสูญเสียการได้ยิน

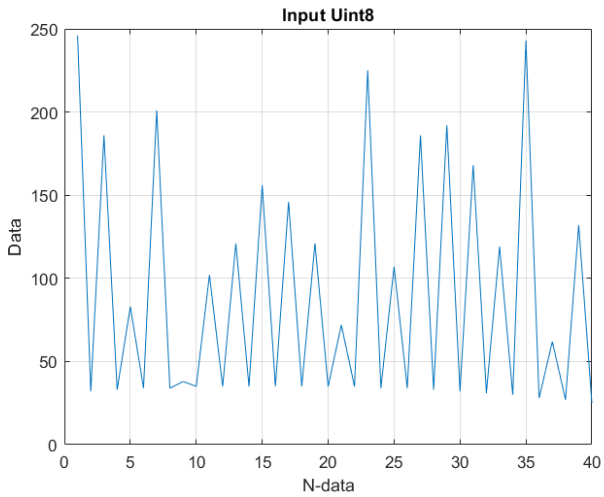
เมื่อเลือกข้างของหูเสร็จ แอปพลิเคชันจะนำพาไปสู่หน้าของระบบการชดเชยการสูญเสียสมรรถภาพการได้ยิน โดยค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมส่วน และค่าอัตราขยาย จะถูก

ส่งมาให้ระบบเรียบร้อยโดยหากกดปุ่มสีเขียวจะสามารถฟังเสียงที่ถูกชดเชยการสูญเสียสมรรถภาพการได้ยินในขณะนั้น ดังรูปที่ 4.48



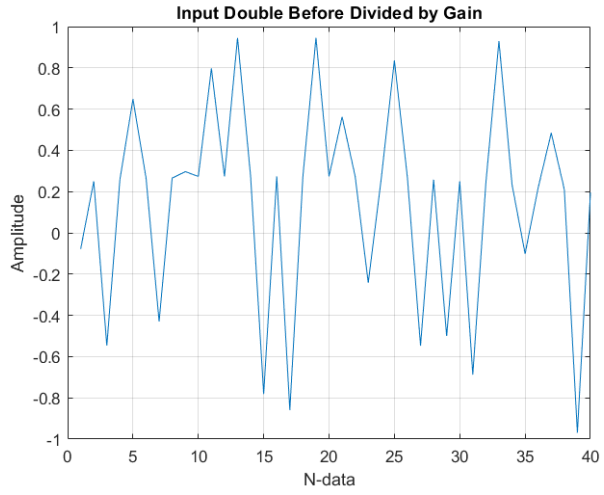
รูปที่ 4.48 หน้าระบบการชดเชยการสูญเสียสมรรถภาพการได้ยิน

เมื่อกดปุ่มสีเขียวจะเป็นการเริ่มกระบวนการชดเชยการสูญเสียสมรรถภาพการได้ยิน โดยจะรับข้อมูลเสียงจากไมโครโฟนและแบ่งข้อมูลออกเป็นชุด เพื่อให้ระบบยังสามารถส่งข้อมูลออกทางหูฟังในขณะที่ยังใช้งานระบบทันที เมื่อนำข้อมูลบางส่วนของชุดข้อมูลเสียงจากไมโครโฟนในรูปแบบ Uint8 ออกมาแสดงดังรูปที่ 4.49



รูปที่ 4.49 สัญญาณข้อมูลเสียงจากไมโครโฟนรูปแบบ Uint8

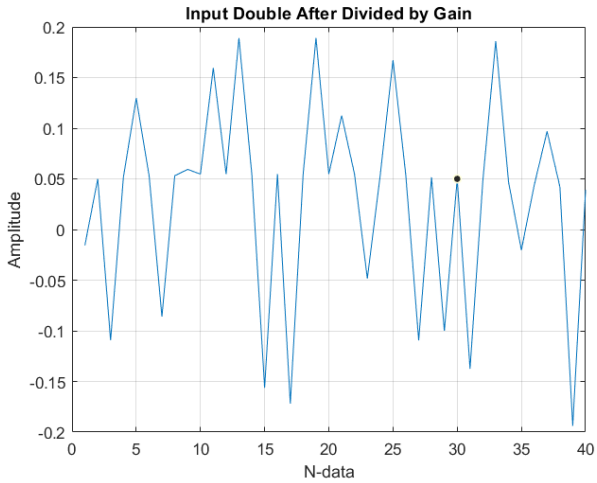
จากรูปที่ จะเห็นว่าข้อมูลเสียงจากไมโครโฟนยังเป็นข้อมูลรูปแบบ Uint8 จึงต้องทำกระบวนการแปลงรูปแบบข้อมูลให้อยู่ในรูปแบบ Double เพื่อให้สามารถนำข้อมูลเสียงไปใช้กับวงจรกรองความถี่แบบดิจิทัล เพื่อทำการชดเชยการสูญเสียสมรรถภาพการได้ยิน โดยมีการแปลงด้วยทฤษฎี 2nd Complement ด้วยหลักการ Q7 และทำการ Normalized เพื่อให้ข้อมูลมีค่าอยู่ในช่วง -1 ถึง 1 ซึ่งจะแสดงได้ดังรูปที่ 4.50



รูปที่ 4.50 สัญญาณข้อมูลเสียงจากไมโครโฟนรูปแบบ Double

แต่เนื่องจากกระบวนการชดเชยการสูญเสียสมรรถภาพการได้ยินมีโอกาสที่จะเกิดการ Overflow ของข้อมูล จึงต้องลดขนาดของข้อมูลเสียงก่อนนำไปใช้กับวงจรกรองความถี่แบบดิจิทัลโดยจะถูกลดขนาดเป็นจำนวนตามอัตราขยายสูงสุดของแต่ละวงจรกรองความถี่แบบดิจิทัล โดยค่าอัตราขยายสูงสุดของค่าการสูญเสียสมรรถภาพการได้ยินนี้มีค่า 5.735256354483635 โดยค่านี้จะถูก

นำไปลดขนาดของข้อมูลเสียงและนำไปใช้ร่วมกับวงจรกรองความถี่แบบดิจิทัล ซึ่งหลังจากลดขนาดของข้อมูลจะสามารถแสดงได้ดังรูปที่ 4.51



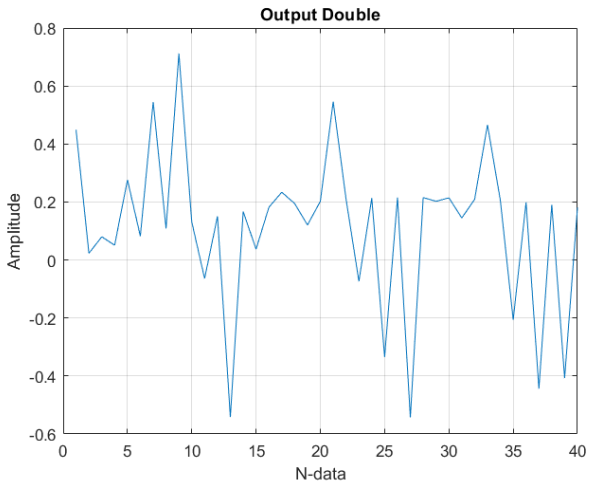
รูปที่ 4.51 สัญญาณข้อมูลเสียงจากไมโครโฟนรูปแบบ Double หลังจากลดขนาดข้อมูล

หลังจากทำการลดขนาดของข้อมูลเสียงจะสามารถนำข้อมูลเสียงมาชดเชยการสูญเสียสมรรถภาพการได้ยินร่วมกับวงจรกรองความถี่แบบดิจิทัล โดยมีค่าสัมประสิทธิ์เทอมเศษ ค่าสัมประสิทธิ์เทอมส่วน และค่าอัตราขยายของแต่ละวงจรกรองความถี่แบบดิจิทัลดังตารางที่ 4.27

ตารางที่ 4.27 ค่าสัมประสิทธิ์และค่าอัตราขยายของแต่ละวงจรกรองความถี่แบบดิจิทัล

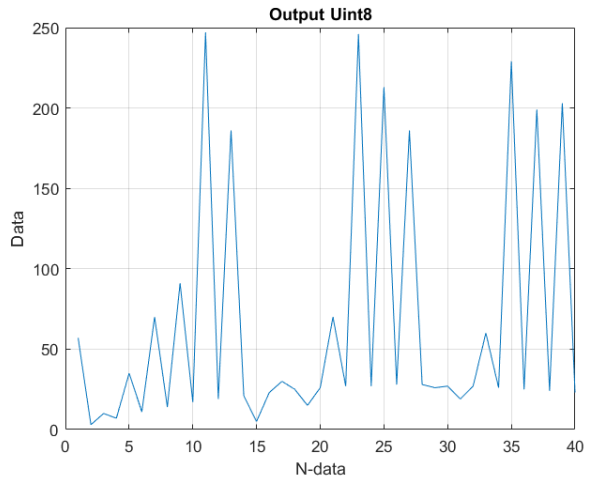
ชนิดของ filter	$H(z)$	**ค่าเกน (G)**
Lowpass Filter	$H(z) = \frac{0.01768 + 0.03537z^{-1} + 0.01768z^{-2}}{1 - 1.41905z^{-1} + 0.48981z^{-2}}$	4.82232
Bandpass Filter	$H(z) = \frac{0.30613 + 0 - 0.30613z^{-2}}{1 - 1.13436z^{-1} + 0.38772z^{-2}}$	5.73525
Highpass Filter	$H(z) = \frac{0.35512 - 0.71025z^{-1} + 0.35512z^{-2}}{1 - 0.16770z^{-1} + 0.25279z^{-2}}$	1.71755

จากค่าสัมประสิทธิ์และค่าอัตราขยายข้างต้น จะสามารถนำค่าเหล่านี้มาใช้ร่วมกับสมการผลต่างสืบเนื่องเพื่อให้ข้อมูลเสียงถูกชดเชยการสูญเสียสมรรถภาพการได้ยิน โดยข้อมูลที่ผ่านสมการผลต่างสืบเนื่องจะมีข้อมูลเป็นรูปแบบ Double เหมือนรูปแบบที่นำมาใช้ในสมการผลต่างสืบเนื่อง ซึ่งสามารถนำมาแสดงเป็นสัญญาณได้ดังรูปที่ 4.52



รูปที่ 4.52 สัญญาณเสียงที่ผ่านวงจรกรองความถี่แบบดิจิทัลในรูปแบบ Double

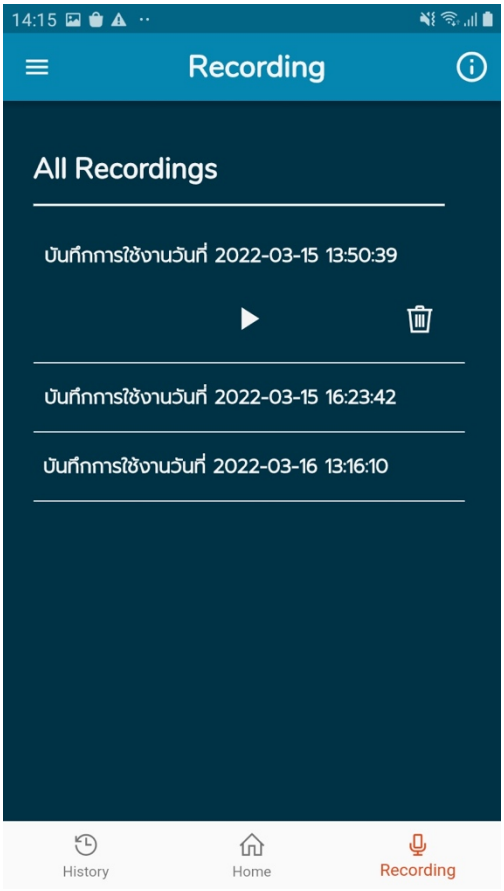
เมื่อได้ข้อมูลที่ถูกชดเชยการสูญเสียสมรรถภาพการได้ยิน จะนำข้อมูลที่ได้เข้าสู่กระบวนการแปลงรูปแบบข้อมูลอีกครั้งเพื่อให้ข้อมูลที่ได้แปลงเป็นเสียงแล้วให้ผู้ใช้งานได้ยินเสียงที่ถูกชดเชยแล้ว โดยจะแปลงข้อมูลจากรูปแบบ Double ให้เป็นรูปแบบ Uint8 โดยใช้ทฤษฎี 2's Complement หรือเป็นกระบวนการย้อนกลับเหมือนกระบวนการแปลงรูปแบบข้อมูลก่อนที่จะนำมาใช้กับสมการผลต่างสืบเนื่อง ซึ่งจะมีลักษณะสัญญาณได้ดังรูปที่ 4.53



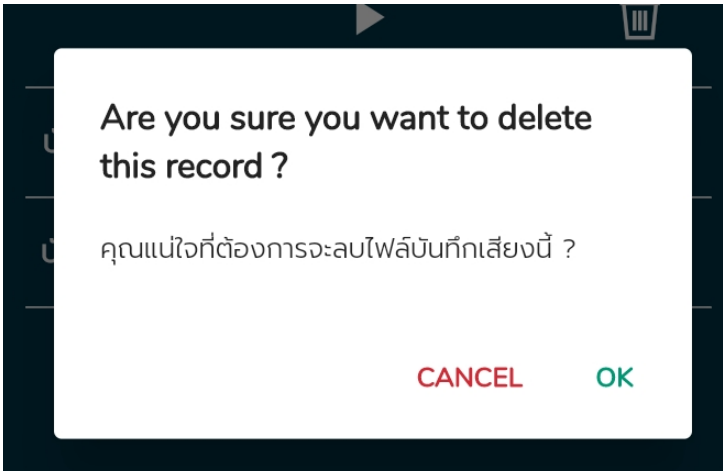
รูปที่ 4.53 สัญญาณข้อมูลเสียงที่ผ่านวงจรกรองความถี่แบบดิจิทัลในรูปแบบ Uint8

หลังจากได้แปลงรูปแบบข้อมูลจาก Double ให้อยู่ในรูปแบบ Uint8 ก็จะสามารถนำข้อมูลที่ได้ส่งกลับไปที่หูฟังเพื่อให้ผู้ใช้งานได้ยินเสียงที่ถูกชดเชยการสูญเสียสมรรถภาพการได้ยินผ่านหูฟังของตนเอง โดยผู้ใช้งานจะต้องใส่หูฟังเฉพาะข้างที่ตนเองได้เลือกไว้ในตอนต้น เพื่อให้การชดเชยการสูญเสียได้มีประสิทธิภาพสูงสุด และระบบทั้งหมดที่กล่าวมายังสามารถนำไปใช้ในฟังก์ชันเสริมที่เป็นการ

บันทึกเสียงเพื่อให้ผู้ใช้งานสามารถบันทึกเสียงที่ตนเองต้องการไว้ ซึ่งเสียงที่ถูกบันทึกไว้จะเป็นเสียงที่ถูกชดเชยการสูญเสียสมรรถภาพการได้ยินเรียบร้อยแล้ว แสดงดังรูปที่ 4.54



รูปที่ 4.54 หน้าแสดงไฟล์บันทึกเสียง



รูปที่ 4.55 กล่องข้อความในกรณีที่ผู้ใช้งานต้องการลบไฟล์บันทึกเสียง

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์นี้ประสบความสำเร็จในการออกแบบและสร้างโมบายแอปพลิเคชันสำหรับทดสอบการได้ยินและแอปพลิเคชันเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน โดยระบบประกอบไปด้วย 4 ส่วน คือ ส่วนที่หนึ่งคือ การออกแบบและสร้างหน้าโมบายแอปพลิเคชันเพื่อผู้ใช้งาน โดยใช้โปรแกรมภาษา Flutter ในการสร้าง ส่วนที่สองคือ การสร้างสัญญาณความถี่เดียว (Pure tone) เพื่อนำมาใช้ทดสอบสมรรถภาพการได้ยินในแอปพลิเคชัน ส่วนที่สามคือ การนำผลการทดสอบสมรรถภาพการได้ยินไปออกแบบวงจรกรองความถี่แบบดิจิทัลในส่วนของเซิร์ฟเวอร์ และส่วนสุดท้ายคือ ระบบเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยินด้วยวงจรกรองความถี่แบบดิจิทัลที่ออกแบบในแอปพลิเคชัน ซึ่งในส่วนแรกได้ทำการออกแบบและสร้างให้ผู้ใช้ได้มีการใช้งานที่หลากหลาย อย่างเช่น ระบบลงทะเบียนเข้าใช้งาน ระบบประวัติผู้ใช้งาน ระบบประวัติการทดสอบสมรรถภาพการได้ยิน ระบบผลการทดสอบสมรรถภาพการได้ยิน ระบบเก็บไฟล์เสียงที่ถูกชดเชยเพื่อฟังภายหลัง และระบบเครื่องช่วยฟังภายในแอปพลิเคชัน ซึ่งผู้ใช้งานสามารถเข้าถึงระบบดังกล่าวเพื่อให้ได้ประสบการณ์การใช้งานที่ต่อเนื่อง ส่วนที่สองจะสร้างสัญญาณความถี่เดียวโดยใช้โปรแกรมภาษาไพธอน โดยเสียงจะมีความถี่ตั้งแต่ 250, 500, 1000, 2000, 4000 และ 8000 เฮิรตซ์ และแยกเสียงสำหรับหูแต่ละข้างเพื่อใช้การทดสอบสมรรถภาพการได้ยิน และนำมาประยุกต์ใช้กับแอปพลิเคชันสำหรับทดสอบการได้ยิน ในส่วนที่สามเมื่อได้ผลการทดสอบจากแอปพลิเคชันสำหรับการทดสอบการได้ยิน จะนำไปใช้ในการออกแบบวงจรกรองความถี่แบบดิจิทัลบนเซิร์ฟเวอร์ เพื่อให้คำนวณหาค่าสัมประสิทธิ์ของวงจรกรองความถี่เพื่อใช้ในแอปพลิเคชันเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน และในส่วนสุดท้ายในการสร้างเครื่องช่วยฟังด้วยค่าสัมประสิทธิ์ของวงจรกรองความถี่แบบดิจิทัล โดยผู้ใช้งานสามารถฟังเสียงที่ถูกชดเชยได้ในเวลาขณะนั้นและมีส่วนเสริมเป็นการอัดเสียงที่ถูกชดเชย เพื่อให้ผู้ใช้งานได้ใช้งานได้ในขณะที่ใช้งานแอปพลิเคชันตลอดเวลา

5.2 ปัญหาและข้อเสนอแนะ

การออกแบบและสร้างแอปพลิเคชันสำหรับทดสอบการได้ยินและแอปพลิเคชันเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยินสามารถนำไปปรับปรุง แก้ไข และพัฒนาได้ในอนาคตให้สามารถมีความเสถียรของระบบมากยิ่งขึ้น และระบบอื่นๆเพื่อให้ตอบสนองความต้องการของผู้ใช้งาน และครอบคลุมการใช้งานให้ครบทุกองค์ประกอบ ซึ่งแอปพลิเคชันยังมีข้อบกพร่องอยู่หลายประการเช่น ความเร็วในการใช้งานของแอปพลิเคชันที่ไม่เหมาะสมกับเทคโนโลยี จึงสามารถแก้ปัญหาโดยการจัดโครงสร้างของโปรแกรมและคัดกรองข้อมูลที่ไม่ได้ใช้งานออกไป เพื่อให้ความเร็วของแอปพลิเคชันเร็วขึ้นและตอบสนองมากยิ่งขึ้น ข้อบกพร่องในเรื่องสภาพแวดล้อมในการทดสอบสมรรถภาพการได้ยินที่ไม่สามารถทราบถึงความเหมาะสมของสภาพแวดล้อม จึงสามารถเพิ่มระบบการวัดเสียงสภาพแวดล้อมเพื่อให้สามารถหาพื้นที่ที่เหมาะสมในการทดสอบได้ และยังมีข้อบกพร่องอื่นในเรื่องความเร็วในการส่งข้อมูลเสียงที่ถูกชดเชยยังมีความช้าเมื่อเทียบกับสถานการณ์ปัจจุบัน จึงสามารถจัดการเรียงข้อมูลในการประมวลผลหรือรวมระบบในการส่งข้อมูลให้มีความกระชับมากยิ่งขึ้น เพื่อให้มีกระบวนการทำงานลดลงจะสามารถเพิ่มความเร็วในการส่งข้อมูลมากยิ่งขึ้น นอกจากนี้เนื่องด้วยสถานการณ์โควิด-19 จึงทำให้แอปพลิเคชันทั้งสองยังไม่สามารถนำไปทดลองใช้กับผู้บกพร่องทางการได้ยิน จึงสามารถนำไปพัฒนาและต่อยอดเพิ่มเติมได้ในอนาคต เพื่อให้การทำงานของแอปพลิเคชันเกิดประโยชน์และประสิทธิภาพสูงสุดต่อผู้ใช้งาน

บรรณานุกรม

[1] เมดิโปรการแพทย์และสุขภาพ. “การตรวจสมรรถภาพการได้ยิน.”
https://www.chanahospital.go.th/content/การตรวจสมรรถภาพการได้ยิน.
[2] MedThai. “การตรวจการได้ยิน (Audiometry หรือ Audiometric Test).”
https://medthai.com/การตรวจการได้ยิน/.
[3] ณัฐดนัย เนียมทอง. “ได้ยินหรือไม่ได้ยิน ทดสอบกันอย่างไร.”
https://www.scimath.org/article-physics/item/7862-2018-02-22-02-48-07.
[4] Mr Mike Pringle FRCS(ORL). “Hearing Tests.”
https://earsandhearinguk.com/ear/deafness/about-deafness/deafness-hearing_tests/.
[5] Chiemi Tanaka PhD. “Speech Audiometry.”
https://www.sciencedirect.com/topics/medicine-and-dentistry/speech-audiometry.
[6] International Organization for Standardization, “Acoustics - Audiometric test methods - Part 2: Sound field audiometry with pure-tone and narrow-band test signals”, 2009.
[7] Warwick Williams, “The calculation of maximum permissible ambient noise levels for audiometric testing to a given threshold level with a specified uncertainty.”, 2010.
[8] JENNIFER JUNNILA WALKER, MD, MPH, U.S. Army Health Clinic, Schofield Barracks, Hawaii
LEANNE M. CLEVELAND, AuD, Fort Richardson Troop Health Clinic, Joint Base Elmendorf-Richardson, Alaska JENNY L. DAVIS, AuD, Landstuhl Regional Medical Center, Landstuhl, Germany JENNIFER S. SEALES, AuD, General Leonard Wood Army Community Hospital, Fort Leonard Wood, Missouri, Am Fam Physician. “Audiometry Screening and Interpretation.”
https://www.aafp.org/afp/2013/0101/p41.html.
[9] Meyer-Bisch. “A-weighting.”
https://en.wikipedia.org/wiki/A-weighting.
[10] ทรงพล แจ่มแจ้ง. Equal Loudness Contour สิ่งที่ Sound Engineer ต้องรู้.
https://www.liveforsound.com/equal-loudness-contour-is/?fbclid=IwAR1kFDPaJw1XU_Cc6jGmZNfGx5JTzmoj6yVWXHjYQI2kTn_e6hSiaUpFv9M.

[11] Sirawit. ทำความรู้จัก Firebase และผลิตภัณฑ์ต่าง ๆ ในช่วงต้นปี 2019 กัน. https://medium.com@sirawit/firebase-คืออะไร-ทำความรู้จัก-firebase-ในช่วงต้นปี-2019-กัน-473a8e8699fb.

[12] Sarayut Nonsiri, PhD. ภาษาโปรแกรม Python คืออะไร ?. https://www.9experttraining.com/articles/python-คืออะไร.

[13] “ขั้นตอนการติดตั้ง Python (ไพทอน) เวอร์ชั่น 3.6.3 ลงบน Windows 10.” https://www.mindphp.com/บทเรียนออนไลน์/83-python/4876-installpython.html.

[14] ROSSARIN PHANITWONG. “Node JS คืออะไร?.” https://www.glurgeek.com/education/ node-js-คืออะไร/.

[15] SUPPORT THAIEASYELEC. “ปฐมบท MQTT.” https://blog.thaieasyelec.com/introduction-to-mqtt/.

[16] A. Oppenheim. “IIR TO FIR SYSTEM CONVERSION.” http://www.controlsystemsacademy.com/0028/0028.html.

[17] Xiaobin@CMU. “Dynamic Optimization.” https://sites.google.com/site/xiaobincmu/DynamicOptimization.

[18] g_abhiroop. “Optimization algorithm in MATLAB.” https://www.fiverr.com/g_abhiroop/implement-any-optimization-algorithm-in-matlab.

[19] ทิวกาล แซ่ตั้ง. “การออกแบบวงจรกรองสัญญาณดิจิตอลชนิดผลตอบสนองอิมพัลส์ไม่จำกัดแบบผสมผสานโดยใช้ไบลิเนียร์ปาสคาลเมตริกซ์และการหาค่าเหสมาะที่สุดด้วยขั้นตอนวิธีเชิงพันธุกรรมแบบปรับปรุงสำหรับเครื่องช่วยฟังแบบดิจิตอล = Hybrid IIR digital filter design using bilinear pascal matrix and optimization with modified genetic algorithm for digital hearing aids.”, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2557.

[20] “ระเบียบวิธีคำนวณเชิงตัวเลขสำหรับงานวิศวกรรม.” http://cheqa.rmuti.ac.th/rmuti_1700/2558/255897/2.2/2.8.pdf.

[21] “Samsung Galaxy J7 Core.” https://ipricethailand.com /ราคา/samsung-galaxy-j7-core/.

[22] “Kramer USB 2.0 Type-A Male to Micro USB 2.0 Type-B Male Cable (3').” https://www.bhphotovideo.com/c/product/1282063-REG/kramer_c_usb_microb_3_usb2_micro_b_5_pin.html.

[23] Carrie Tsai. “แจ็คหูฟัง: มันคืออะไรและมันทำงานอย่างไร.”
https://www.neway.mobi/th/news/what-is-headphone-jack-and-how-does-it-work.html.
[24] James Russell. “The Human Voice and the Frequency Range.”
https://blog.accusonus.com/pro-audio-production/human-voice-frequency-range/.
[25] “Reasons Not to Describe Degree of Hearing Loss as Percentage.”
https://hkincus.com/blogs/blogs/hearing-loss-percentage-vs-db.
[26] Scientific Committee on Emerging and Newly Identified Health Risks SCENIHR. “Potential health risks of exposure to noise from personal music players and mobile phones including a music playing function.”, 2008.
[27] Athiwat. “Data Science on the Google Cloud Platform: ทำความรู้จักกับ Google Cloud Platform (GCP).”
https://medium.com/machines-school/data-science-on-the-google-cloud-platform-ท ำความรู้จักกับ-google-cloud-platform-gcp-7605b4560fb8
[28] T.-B. Deng. “Three-channel variable filter-bank for digital hearing aids.” Department of Information Science, Faculty of Science, Toho University, 2008.
[29] “DigitalOcean คืออะไร สมัครใช้งาน คลาวด์เซิร์ฟเวอร์ วันนี้มีเครดิต 100$ ฟรี.”
https://www.mindphp.com/คู่มือ/73-คืออะไร/6619-digitalocean.html
[30] “สมการผลต่างสืบเนื่อง (Difference Equation).” http://old-book.ru.ac.th/e-book/e/EC371/ec371-7-1.pdf
[31] “Analog vs Digital: What is the Difference Between Analog and Digital?.”
https://www.guru99.com/analog-vs-digital.html
[32] “การส่งผ่านข้อมูลดิจิตอล (Digital Transmission).”
http://personal.sut.ac.th/paramate/files/compcom/compcomm04.pdf
[33] “มาทำความรู้จัก รูปแบบสัญญาณเสียง Digital แบบ PCM กันเถอะ (ฉบับปรับปรุง).”
https://rev.at1987.com/articles/pulse-code-modulation/2/
[34] DigitalOcean. “How to Create a Droplet from the DigitalOcean Control Panel.”
https://docs.digitalocean.com/products/droplets/how-to/create/
[35] Chai Phonbopit. “Digital Ocean คืออะไร ? + สอนวิธีการติดตั้งและสร้าง Droplet.”
https://www.devahoy.com/blog/2014/03/getting-started-with-digital-ocean

[36] DigitalOcean. “How to Create SSH Keys with OpenSSH on MacOS or Linux.” https://docs.digitalocean.com/products/droplets/how-to/add-ssh-keys/create-with-openssh/

[37] Wancheng Zhou. “AVR 16bit Stereo Wave Player Simple, Cost-Effective, CD Audio Quality.”https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2014/wz233/ECE%204760%20Final%20Report%20(HTML)/ECE%204760%20Stereo%20Player.html

[38] The Hearing Journal. “Congenital Conductive Hearing Loss.” https://journals.lww.com/thehearingjournal/Fulltext/2014/03000/Symptom___Congeni

[39] INTIMEX. “เครื่องช่วยฟัง คืออะไร เหมาะกับการได้ยินแบบไหน.” https://www.intimexhearing.com/hearingaid/

ภาคผนวก ก

โค้ดที่ใช้ออกแบบระบบโดยรวมของแอปพลิเคชันสำหรับการทดสอบได้ยิน

1. โค้ดโปรแกรมภาษา Flutter สำหรับการออกแบบระบบการทำงาน

https://drive.google.com/drive/folders/1g2mtWL_6fAVmlbBrYRyWtzqm6Hmkkl9w?usp=sharing

2. โค้ดโปรแกรมภาษา Python สำหรับการสร้างสัญญาณเสียง

```
T0 = 80 #defualt 80
T = 4*T0
N = np.arange(0,T,1)
f = [250, 500, 1000, 2000, 4000, 8000]

SPL = [40,34,40,37,32,43]
dBv = []
Amp = []
for i in range(len(SPL)):
    v = 20*(math.log(0.05*(10**(((SPL[i]+100)-93.9)/20)),10))
    dBv.append(v)
    amp = 10**(v/20)
    Amp.append(amp)

a = []
c = []
Fs = []
j = 0
for i in range(6):
    b = []
    Fs.append(f[i]*40)
    x = 2*np.pi*f[i]*N/Fs[i]
    a = amp*np.sin(x)
    time = 300*(f[i]//250)
    for j in range(time):
        b.extend(a)
    c.append(b)
print('Audio play is done')

for i in range(len(c)):
    x = np.float32(c[i])
    ct = np.transpose(c[i])
    ct0 = np.zeros(len(c[i]),dtype='d')
    ct1 = np.transpose(ct0)
    m = np.reshape(ct,1*len(ct))
    yleft = np.column_stack((m,ct1))
    x = np.float32(yleft)
    write("left{0}.mp3".format(i+1), Fs[i], x)
print('DONE')
```

```
T0 = 80 #defualt 80
T = 4*T0
N = np.arange(0,T,1)
f = [250, 500, 1000, 2000, 4000, 8000]

SPL = [40,34,40,37,32,43]
dBv = []
Amp = []
for i in range(len(SPL)):
    v = 20*(math.log(0.05*(10**(((SPL[i]+100)-93.9)/20)),10))
    dBv.append(v)
    amp = 10**(v/20)
    Amp.append(amp)

a = []
c = []
Fs = []
j = 0
for i in range(6):
    b = []
    Fs.append(f[i]*40)
    x = 2*np.pi*f[i]*N/Fs[i]
    a = amp*np.sin(x)
    time = 300*(f[i]//250)
    for j in range(time):
        b.extend(a)
    c.append(b)
print('Audio play is done')

for i in range(len(c)):
    x = np.float32(c[i])
    ct = np.transpose(c[i])
    ct0 = np.zeros(len(c[i]),dtype='d')
    ct1 = np.transpose(ct0)
    m = np.reshape(ct,1*len(ct))
    yleft = np.column_stack((ct1,m))
    x = np.float32(yleft)
    write("right{0}.mp3".format(i+1), Fs[i], x)
print('DONE')
```

ภาคผนวก ข

โค้ดสำหรับการสร้างเซิร์ฟเวอร์เพื่อใช้ออกแบบวงจรกรองความถี่แบบดิจิทัล

1. โค้ดโปรแกรมภาษา Node.js และคำสั่งที่ใช้สำหรับโปรโตคอล MQTT เพื่อประมวลผลบน Cloud Function

```
const functions = require("firebase-functions");
const admin = require("firebase-admin");
const mqtt = require("mqtt");

admin.initializeApp();

const client = mqtt.connect('mqtt://*url mqtt*/ /');

client.on("connect", function () {
  functions.logger.info("Ready To Rock!!");
});

client.on("reconnect", () => {
  functions.logger.info("MQTT client is reconnecting...");
});

client.on("disconnect", () => {
  functions.logger.info("MQTT client is disconnecting...");
});

client.on("error", function () {
  functions.logger.error("Can't connect");
  client.reconnect();
});

exports.createUserAudioGram = functions
  .region("asia-southeast1")
```

```
  .firestore.document("users/{userId}/audiogram_history/{audiogram_historyId}")
  .onCreate((snapshot, __) => {
    functions.logger.info(snapshot.data());
    let trigg_message = {
      userID: __.params.userId,
      audiogramID: __.params.audiogram_historyId,
      lValue: snapshot.data().Left,
      rValue: snapshot.data().Right,
    };
    client.publish("/audiogram", JSON.stringify(trigg_message), function (err) {
      if (err) {
        functions.logger.error("Error:" + err);
        response.send("Error:" + err);
        reject();
      }
    });
    return Promise.resolve();
  });

exports.updateUserAudioGram = functions
  .region("asia-southeast1")
  .firestore.document("users/{userId}/audiogram_history/{audiogram_historyId}")
  .onUpdate((snapshot, __) => {
    functions.logger.info(snapshot.before.data());
    functions.logger.info(snapshot.after.data());
    return Promise.resolve();
  });
```

1.1 แพ็คเก็ตที่ติดสั่งสำหรับการรันข้อมูลในส่วนภาษาโปรแกรม Node.js

```
{
  "name": "functions",
  "description": "Cloud Functions for Firebase",
  "scripts": {
    "serve": "firebase emulators:start --only functions",
    "shell": "firebase functions:shell",
    "start": "npm run shell",
    "deploy": "firebase deploy --only functions",
    "logs": "firebase functions:log"
  },
  "engines": {
    "node": "10"
  },
  "main": "index.js",
  "dependencies": {
    "firebase-admin": "^9.8.0",
    "firebase-functions": "^3.14.1",
    "mqtt": "^4.2.6"
  },
  "devDependencies": {
    "firebase-functions-test": "^0.2.0"
  },
  "private": true
}
```

2. โค้ดโปรแกรมภาษา Python สำหรับประมวลผลค่าสัมประสิทธิ์เพื่อใช้ออกแบบวงจรกรองความถี่แบบดิจิทัล

```
import numpy as np
import math
import logging
import os
import json
import matplotlib.pyplot as plt
from numpy.core.fromnumeric import var
import pandas as pd
import scipy.optimize
from scipy import signal
from scipy import interpolate
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from urllib.parse import urlparse

#google cloud session
from datetime import datetime, timezone
import time
import pytz
import paho.mqtt.client as mqtt
import firebase_admin
from dotenv import load_dotenv
from firebase_admin import credentials, firestore
from pydash import get

load_dotenv()

# recived incoming data ??
```

```
cred = credentials.Certificate("hearing-test-2e94b-firebase-adminsdk-6qp9b-
5a2383d7ad.json")
firebase_admin.initialize_app(cred)
db = firestore.client()

host = os.getenv('CLOUDMQTT_URL')
url = urlparse(host)
topic = url.path[1:]

def on_connect(self, client, userdata, rc):
    logging.info('connected')
    self.subscribe("/"+topic, 0)

def on_message(client, userdata, msg):
    logging.debug(msg.payload.decode("utf-8", "strict"))
    m_in = json.loads(msg.payload.decode("utf-8", "strict"))
    user_id = get(m_in, 'userID')
    a_id = get(m_in, 'audiogramID')
    leftValue = get(m_in, 'lValue')
    rightValue = get(m_in, 'rValue')

    Data = [leftValue,rightValue]
    Lcoeff=[]
    Bcoeff=[]
    Hcoeff=[]

    # processing data
    #fminserch
    for i in range(2):

        banana = lambda x: ErrorFunction(x[0],x[1],x[2],x[3],x[4],x[5],x[6],x[7],x[8], Data[i])
```

```
        opt = scipy.optimize.fmin(func=banana,
x0=[1599,0.299,9.5,4000,0.899,10,8000,0.5,10],xtol=1e-4,ftol=1e-4,full_output=True)
#,full_output=True to see all outputs [1500,0.3,10,4000,0.9,10,8000,0.5,10]
        print('Current function value: ',opt[1])
        print('Iterations: ',opt[2])
        print('Function evaluations: ',opt[3])

        xopt = opt[0]
        print('Coefficients: ',xopt)
        flpf = abs(xopt[0])
        Qlpf = abs(xopt[1])
        Glpf = abs(xopt[2])
        fbpf = abs(xopt[3])
        Qbpf = abs(xopt[4])
        Gbpf = abs(xopt[5])
        fhpf = abs(xopt[6])
        Qhpf = abs(xopt[7])
        Ghpf = abs(xopt[8])
        q,h_lpf,num_lpf,den_lpf = BiquadLPF(flpf,Qlpf)

        q,h_bpf,num_bpf,den_bpf = BiquadBPF(fbpf,Qbpf)

        q,h_hpf,num_hpf,den_hpf  = BiquadHPF(fhpf,Qhpf)

        LL=np.concatenate((num_lpf,den_lpf,Glpf),axis = None)
        BB=np.concatenate((num_bpf,den_bpf,Gbpf),axis = None)
        HH=np.concatenate((num_hpf,den_hpf,Ghpf),axis = None)

        #sending patient's data back to firebase
        UTC = pytz.utc
        Lcoeff.append((LL).tolist())
```

```
        Bcoeff.append((BB).tolist())
        Hcoeff.append((HH).tolist())

    # update back to DB
    doc_result =
db.collection(u'users').document(user_id).collection(u'audiogram_history').document(a_i
d)
    doc_result.update({
        u'LPFcoeff_Left': Lcoeff[0], # array
        u'BPFcoeff_Left': Bcoeff[0],
        u'HPFcoeff_Left':  Hcoeff[0],
        u'LPFcoeff_Right': Lcoeff[1],
        u'BPFcoeff_Right': Bcoeff[1],
        u'HPFcoeff_Right':  Hcoeff[1],
        u'updated_at': datetime.now(UTC),
    })


def BiquadLPF(flpf,Qlpf):
    f_samp = 44100
    ohm0 = np.tan(np.pi*flpf/f_samp)
    A1 = [(ohm0)**2, 0, 0]
    B1 = [(ohm0)**2, ohm0/Qlpf, 1]
    P = [[1,1,1],
        [2,0,-2],
        [1,-1,1]]
    a1 = np.dot(P,np.transpose(A1))
    b1 = np.dot(P,np.transpose(B1))
    num= a1/b1[0]
    den = b1/b1[0]
    q1,h1 = signal.freqz(num,den,worN=1000,fs=44100)#fs=16000
```

```
    return q1,h1,num,den

def BiquadBPF(fbpf,Qbpf):
    f_samp = 44100
    ohm0 = np.tan(np.pi*fbpf/f_samp)
    A1 = [0, ohm0/Qbpf, 0]
    B1 = [(ohm0)**2, ohm0/Qbpf, 1]
    P = [[1,1,1],
        [2,0,-2],
        [1,-1,1]]
    a1 = np.dot(P,np.transpose(A1))
    b1 = np.dot(P,np.transpose(B1))
    num= a1/b1[0]
    den = b1/b1[0]
    q1,h1 = signal.freqz(num,den,worN=1000,fs=44100)
    return q1,h1,num,den

def BiquadHPF(fhpf,Qhpf):
    f_samp = 44100
    ohm0 = np.tan(np.pi*fhpf/f_samp)
    A1 = [0, 0, 1]
    B1 = [(ohm0)**2, ohm0/Qhpf, 1]
    P = [[1,1,1],
        [2,0,-2],
        [1,-1,1]]
    a1 = np.dot(P,np.transpose(A1))
    b1 = np.dot(P,np.transpose(B1))
    num= a1/b1[0]
    den = b1/b1[0]
    q1,h1 = signal.freqz(num,den,worN=1000,fs=44100)
    return q1,h1,num,den
```

```
def ErrorFunction(fL,QL,GL,fB,QB,GB,fH,QH,GH, newval2):
    q,h_lpf,num_lpf,den_lpf = BiquadLPF(fL,QL)
    HL = h_lpf*GL
    q,h_bpf,num_bpf,den_bpf = BiquadBPF(fB,QB)
    HB = h_bpf*GB
    q,h_hpf,num_hpf,den_hpf = BiquadHPF(fH,QH)
    HH = h_hpf*GH
    desired_flt = abs(HL+HB+HH)
    h_desired = 20*np.log10(desired_flt)
    octave_freq = [250,500,1000,2000,4000,8000]
    threshold_dB = newval2
    spc = np.linspace(250,8000,1000)
    h_ideal = np.interp(spc,octave_freq,threshold_dB)
    err = mean_absolute_error(h_ideal,h_desired)
    return err

def on_subscribe(client, obj, mid, granted_qos):
    logging.debug(f'Subscribed: {str(mid)} {str(granted_qos)}')

def on_disconnect(client, userdata, rc):
    logging.debug(f'disconnecting reason {str(rc)}')
    client.connected_flag = False
    client.disconnect_flag = True

client = mqtt.Client()
client.on_connect = on_connect
client.on_disconnect = on_disconnect
client.on_message = on_message
client.on_subscribe = on_subscribe
client.username_pw_set(url.username, url.password)
```

```
client.connect(url.hostname, url.port)
client.loop_forever()
```

2.1 แพ็คเก็ตที่ต้องติดตั้งเพื่อโปรแกรม Python

```
numpy==1.21.4

# math==3.1.0

matplotlib==3.4.3

pandas==1.3.4

scipy==1.7.2

sklearn==0.0


google==3.0.0

datetime==4.3

time-tools==1.0.0

pytz==2021.3

firebase_admin==5.1.0

paho-mqtt

python-dotenv

pydash
```

2.2 โค้ดภาษา JavaScript จาก Firebase ที่ใช้สำหรับเชื่อมต่อระบบ

{

"type": "service_account",

"project_id": "hearing-test-2e94b",

"private_key_id": "private key",

"private_key": "-----BEGIN PRIVATE KEY-----\n private key /n-----END PRIVATE KEY-----\n",

"client_email": "firebase-adminsdk-6qp9b@hearing-test-2e94b.iam.gserviceaccount.com",

"client_id": "118041894423857945049",

"auth_uri": "https://accounts.google.com/o/oauth2/auth",

"token_uri": "https://oauth2.googleapis.com/token",

"auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",

"client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-6qp9b%40hearing-test-2e94b.iam.gserviceaccount.com"

}

ภาคผนวก ค

โค้ดที่ใช้ออกแบบระบบโดยรวมของแอปพลิเคชันเครื่องช่วยฟังสำหรับผู้บกพร่องทางการได้ยิน

1. โค้ดโปรแกรมภาษา Flutter สำหรับการออกแบบระบบการทำงาน

https://drive.google.com/drive/folders/12uqH_YJyMjJaBZ-g7tGU-dNvwaPVP5lA?usp=sharing