MOBILE APPLICATION FOR SIGN LANGUAGE LEARNING WITH REAL TIME FEEDBACK

By

Lee Xiao Xu Alexis

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JANUARY 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Mobile Application for Sign Language Learning with Real-time Feedback

Academic Session: January 2024

I ____________________LEE XIAO XU ALEXIS__________________

(CAPITAL LETTER)

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

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

Verified by,

_________________________ (Author's signature)

_________________________ (Supervisor's signature)

Address:
81, Persiaran Mahsuri 2/4
11900 Bayan Lepas
Pulau Pinang

Ng Hui Fuang
Supervisor's name

Date: 23 April 2024

Date: 25/4/2024

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 23 April 2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that LEE XIAO XU ALEXIS (ID No: 20ACB02090) has completed this final year project entitled "Mobile Application for Sign Language Learning with Real-time Feedback" under the supervision of Dr Ng Hui Fuang (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

DECLARATION OF ORIGINALITY

I declare that this report entitled "**MOBILE APPLICATION FOR SIGN LANGUAGE LEARNING WITH REAL TIME FEEDBACK**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ________________

Name : Lee Xiao Xu Alexis

Date : 23 APRIL 2024

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr Ng Hui Fuang, who has given me this bright opportunity to engage in a mobile application development project. It is my first step to establish a career in mobile application development field. A million thanks to you.

To a very special person in my life, Ng Zheng, for his patience, unconditional support, and love, and for standing by my side during hard times. Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

With over 70 million deaf people worldwide, sign languages serve as means of communication and connection within Deaf communities. However, limited accessibility of sign language education poses barriers to social inclusion and awareness. This project proposes developing an innovative mobile application for interactive sign language learning to benefit both Deaf individuals and hearing loss individuals globally. The app aims to deliver courses methodically from basic vocabulary to advanced grammar, diverse learning materials like video demonstrations, quizzes and exercises. A major innovation of this project involves integrating computer vision and machine learning for real-time sign recognition and feedback during signing exercises. Machine learning algorithms using MediaPipe and deep learning will analyse users' hand motions to provide corrections for improving technique. Overall, this project strives to transform sign language learning through assistive technologies. This mobile application aspires to deliver innovative tools empowering deaf and hearing loss individuals globally to connect across social barriers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

WHO	World Health Organization
ASL	American Sign Language
BSL	British Sign Language
LSF	French Sign Language
AI	Artificial Intelligence
SVM	Support Vector Machine
CSV	Comma-separated values
CNN	Convolutional Neural Network
RNN	Recurrent Neural Networks
LSTM	Long Short-Term Memory

CHAPTER 1 INTRODUCTION

1.1 Project Background

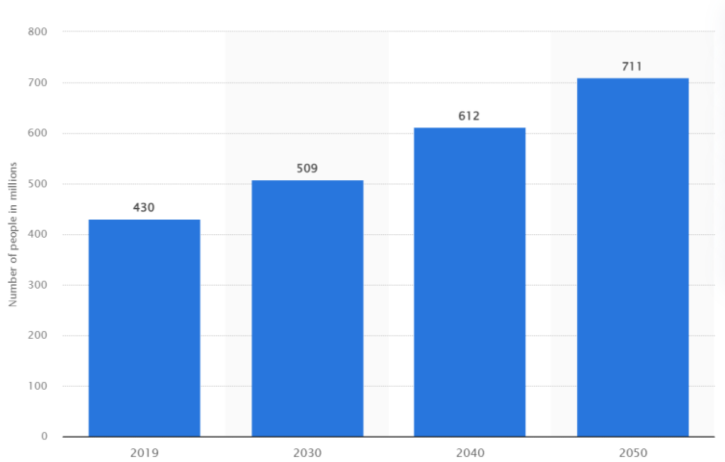


Figure 1. 1 Projected number of disabling hearing loss (in millions) [1]

According to the World Health Organization (WHO), about 5% of the global population experiences some form of hearing loss [2]. According to the [1], in 2019, around 430 million people suffered from disabling hearing loss. It is predicted that by 2050, this number would have risen to almost 711 million. Delays in language acquisition, difficulties learning to read and write, and feelings of isolation are only some of the negative consequences of untreated hearing loss.

Using a wide variety of hand shapes, orientations, movements, facial emotions, and other body movements, sign languages constitute a highly advanced visual-spatial language. They emerge naturally wherever communities of deaf people exist. There are over 300 distinct sign languages used globally by Deaf communities [3]. These include American Sign Language (ASL) which is primarily used in the United States and parts of Canada as shown in Figure 1.2, British Sign Language (BSL) in the UK, Australian Sign Language (Auslan), French Sign Language (LSF), among many others. Sign languages are fully developed natural languages with their own distinctive grammars, syntax rules, and vocabularies independent of the spoken or written language of their country or region [3]. Sign languages allow people who are deaf or hard of hearing to fully engage in social interactions and access education and employment. The ability to communicate effectively in sign language is a crucial link to the hearing world.

The recent breakthroughs in computer vision and machine learning enable sign language learning apps to incorporate real-time gesture detection into the learning process. These apps can recognize the signs a user is making and give them immediate, automated feedback on how well they're doing. Static learning materials cannot provide this level of engagement and individual feedback. Because of its portability, connectivity, video recording, and embedded sensors, mobile technology opens up new avenues for studying sign language, improving the educational experience by automatically recognizing signs and providing feedback using machine learning and computer vision techniques. Google's Mediapipe is an example of an open-source framework for making multimodal applied machine learning pipelines [4]. It offers machine learning solutions that work across platforms and can be modified for live and streaming video. Mediapipe was used in studies for sign language detection and recognition such as hand pose estimation. Integrating Mediapipe into a sign language learning app could allow accurate hand tracking and sign classification to detect errors and provide real-time feedback to users.

1.1.1 Sign Language Learning Mobile Application

The Sign Language Learning Mobile Application aims to improve sign language learning by filling the current gap in sign language education. This is a solution that focuses on individuals, educators, and communities, in order to help them in various ways.

This sign language learning application will provide universal access to quality sign language education, eliminating geographical and socioeconomic barriers [5]. Besides, it will offer a comprehensive curriculum, covering a wide range of sign language topics and proficiency levels, ensuring that users can progress at their own pace. Furthermore, the application will employ interactive lessons, quizzes, and exercises to make learning sign language engaging and effective. Most importantly, it will integrate real-time sign language recognition and feedback, providing users with precise guidance as they learn.

Our main goal for this project is to ensure that learning sign language is not only accessible, but also enjoyable and beneficial for people of all abilities, ages, and backgrounds. This commitment shows a dedication to supporting different languages, improving communication, and creating a society that is more inclusive and understanding.

1.1.2 Sign Language Detection and Recognition

The goal of sign language detection and recognition systems is to automatically analyse video input containing sign language and interpret the signed communication through various stages of processing.

The first stage of sign language detection is hand and body detection. This involves locating the person signing in the video frame and isolating the hand and arm regions from the background using computer vision algorithms like background subtraction, skin color detection and segmentation [6]. This establishes the regions of interest for subsequent analysis. This is then followed by the second stage, feature extraction. low level visual and motion features are extracted from the segmented hand and arm regions in each video frame. Common features include pixel values, edge orientations, optical flow vectors representing handshape and movement, coordinate positions [6].

After feature extractions, classification is then carried out. Machine learning classifiers leverage the extracted spatial and temporal features to recognize individual signs and sign components. Techniques such as convolutional neural networks, hidden Markov models, dynamic time warping, and recurrent neural networks are often employed [7]. After this, language modelling is needed. At the whole sentence or phrase level, context-dependent language models are applied to recognized words, meanings and produce the most probable translation based on grammatical rules and constraints of the sign language. The final output of a sign language recognition system is the transcription of the signed sentence or phrase into the written text of the corresponding spoken or written language.

1.2 Problem Statement and Motivation

Traditional methods of sign language instruction such as in-person classes or printed materials have several limitations that reduce their effectiveness. Classes require students to adhere to fixed schedules, which can be **inflexible** for learners with busy routines. Traveling to and from physical locations for each lesson also increases time commitment and costs. This passive format decreases learner engagement and motivation over time. Therefore, the motivation of this project is to enable more individuals to learn sign languages, by developing a mobile application that overcomes the limitations of traditional methods.

Second, the existing sign language learning apps aim to solve some issues of traditional sign language learning formats such as in class teaching, however, **real-time recognition with feedback are not provided.** Without intelligent assessment through computer vision and machine learning, apps cannot analyze a learner's signing and provide targeted feedback by highlighting the exact keypoints, showing using the correct and incorrect keypoints through visual cues. This lack of personalized instruction and corrections reduces the effectiveness of practice sessions within apps. Thus, this problem motivates the development of system to integrate AI and computer vision techniques for real-time sign language detection and recognition, the app can dynamically evaluate users' signing techniques and automatically generate tailored feedback.

Besides, the existing system are mostly **one-size-fits-all, which fails to tailor learning to different user profiles** [11, 12]. Providing the same core curriculum for all learners, whether children or adults, beginners or advanced students, may not suit each learner's unique needs, preexisting knowledge, and goals. Without personalized profiles, apps cannot deliver effective instruction customized for individual profiles. Thus, this situation motivates the development of a system that offers personalized learning paths based on individual user profiles.

Moreover, the existing **sign language apps and translators are often separated**, making it inconvenient for users. Users often face the inconvenience of having to download a separate translator app to access translation functionalities, resulting in a fragmented user experience. Furthermore, most existing systems lack consideration for combining learning and translating for user to experience a more realistic conversation. This makes it harder for users to learn and communicate naturally. This problem motivates the development of a unified system that integrates sign language learning and translation functionalities, providing users with a natural communication experience.

1.3 Objectives

(a) To develop an interactive sign language learning mobile application

One objective is creating curriculum-based courses covering core language competencies needed for conversational fluency and cultural awareness for levels ranging from beginner to advanced. The learning materials and features support users in building an extensive sign vocabulary, mastering grammar principles, and gaining signing speed and accuracy.

An important objective is incorporating video demonstrations covering a wide range of vocabulary signs, sentences, dialogs, and deaf cultural elements. Filming native signers performing accurate signing provides an immersive learning aid. The video library needs significant scale and diversity representing different signers. Incorporating interactive modules like quizzes, challenges, and conversational dialogue simulations engages users with active signing practice while providing assessment data to the adaptive learning algorithms. Gamification dynamics can further spur participation. Social sharing integration enables community motivation.

Enabling self-directed practice at the learner's own pace and schedule without restrictions imposed by a fixed curriculum timeline or classroom availability is an objective fulfilled by the mobile platform. The portability of the system also enables usage during brief periods of free time, while implementing content chunking techniques supports and accelerates learning progress.

Speech to sign translation is also included to enable users to learn a more realistic conversation, enhancing their overall communication abilities.

(b) To develop a sign language learning mobile application that provides real time feedbacks to users

An advanced objective is developing real-time sign recognition and assessment powered by state-of-the-art deep learning algorithms to analyse signing accuracy. This provides personalized feedback on hand shapes, motions, positioning and timing to improve performance. Significant research and development is needed to technically implement reliable on-device sign language modelling and classification.

1.4 Project Scope and Direction

The first component of the project scope is to develop an **educational mobile application** for sign language learning. This entails designing and building a smartphone app on Android platforms to teach users sign language skills. It should have a polished user interface following platform guidelines and best practices. The development process must ensure the app is performant, secure, accessible, and stable.

A core element of the project scope is including **learning modules** within the app covering vocabulary, grammar, sentence structures, conversational skills, and deaf culture for novice through advanced signers. These modules should be organized into coherent courses and lesson plans to methodically build skills. The learning materials will utilize video demonstrations of signs, quizzes to reinforce memorization, exercises to practice signing, and explanatory text, images, and graphics. The modules must accommodate different learning styles and allow self-directed learning at the student's own pace.

A major portion of the learning content will comprise **video demonstrations** showing fluent signers performing signs, phrases, and conversations. The videos should demonstrate proper technique, speed, and expressions from multiple camera angles for clarity. High-quality video production principles maximize the educational value. A large diverse library of video content will allow learners to immersely watch and absorb sign languages in practice. The videos can also facilitate assessments by having learners replicate the signing.

Interactive learning modules such as quizzes and exercises are another development priority. These activities reinforce memorization, validate comprehension, build physical signing capability, and apply skills. Quizzes help learners self-assess the retention of vocabulary. Physical exercises focus on correctly shaping and moving the hands through sign repetitions and imitation challenges.

One core advantage of a mobile app is enabling self-paced study at any time or place free of commitments interfering with an independent pace. The app experience gives control over when and how long to engage in lessons. **Breaking content into bite-sized chunks** also facilitates a flexible learning pace. As a supplemental or primary learning tool, the app promotes mastery through self-directed, repetitive practice to develop muscle memory.

Another key scope of the project is to implement **personalized user profiles**, accommodating learners at different proficiency levels, including beginners, intermediates, and advanced users. Each profile will offer tailored learning paths, content recommendations, and progress tracking based on the user's demonstrated skills and learning goals.

Another significant component of the project scope is the integration of a **speech-to-sign translator** within the mobile application. This feature enables users to input spoken language and receive real-time translations into sign language. The translator should accurately interpret spoken words and phrases into appropriate signs.

The final major scope component is **sign language recognition and real-time feedback** powered by computer vision and machine learning techniques. Recording video on the mobile device during signing practice allows AI algorithms to analyse the user's hand shapes, motions, and body positions to evaluate accuracy and provide corrections in real-time. Deep learning models can classify signs and break down nuances like improper finger positions. This creates an intuitive practice environment for learners to refine their skills. The computation should happen on-device to avoid privacy concerns of uploading video. State-of-the-art algorithms are necessary for responsive assessments rather than just binary right or wrong feedback.

1.5 Contribution

The mobile app developed for sign language learning contributes to both society and Android users. Firstly, by making sign language learning more accessible to a larger audience, it promotes inclusivity in society. This encourages more polite and meaningful relationships by improving communication and comprehension between individuals who use sign language and those who do not. The app's user-friendly design, interactive learning modules, and personalized paths cater to Android users of all skill levels, making sign language learning engaging and achievable.

Using technologies like computer vision and machine learning, the app's real-time sign language recognition capability is one of its main features. With the use of this tool, users can get instant feedback on the correctness of their signing, which aids in skill improvement. This feature allows users to receive immediate feedback on their signing accuracy, helping them refine their skills effectively. Android users can practice sign language with confidence, knowing they're getting precise guidance, improving education, and making the app valuable for both individuals and society as a whole.

1.6 Report Organization

This report is organised into 7 chapters:

Chapter 1 is the introduction to sign language learning mobile application. Sections included are project background, problem statement and motivation, objectives, project scope and motivation and contribution. Chapter 2 is regarding the literature reviews; review was made on both existing Sign language learning Application and previous work on sign language recognition. Chapter 3 includes the System Methodology and approach such as the general workflow procedure, system design diagram and project timeline. Chapter 4 is the System Design, including system flowchart, database design, system block diagram and the component specifications, as well as the system function. System implementation details such as software and hardware setup, system operations and system function implementation details were recorded in chapter 5. System Evaluation and discussion were made in Chapter 6. Lastly, chapter 7 includes the conclusion and recommendation of the project.

CHAPTER 2 LITERATURE REVIEW

2.1 Review of Existing Sign Language Learning Application

2.1.1 Sign School

SignSchool is a mobile application designed to help users learn American Sign Language (ASL). The core functions of SignSchool are lessons, vocabulary, and signing sessions [10].

The lessons' function is where users can progress through structured ASL lessons. Within the lessons, there are videos showing an instructor demonstrating the signs accompanied by written explanations and example sentences as in Figure 2.1. The lessons start with basic vocabulary like introductions and greetings, and progress to more complex grammar topics. Users can take lessons at their own pace, repeating material as needed. Once a lesson has been completed, users can review a summary of the key signs and concepts. This provides a comprehensive way for users to learn ASL from the fundamentals up in an organized, step-by-step manner.

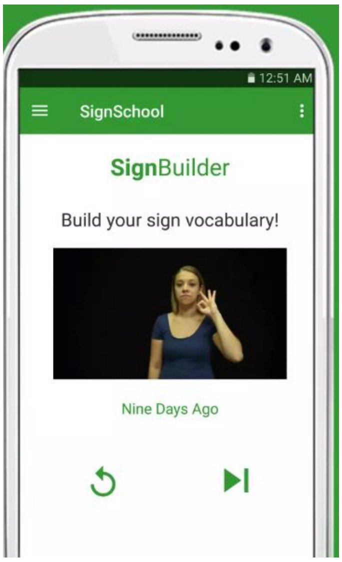


Figure 2. 1 Video Demonstration with Written Explanations of SignSchool App [10]

The topics function allows users to browse and study ASL signs based on desired topics as shown in Figure 2.2. Users can spend time learning new vocabulary related to any topic that interests them, such as Jobs, hobbies, Music and more.

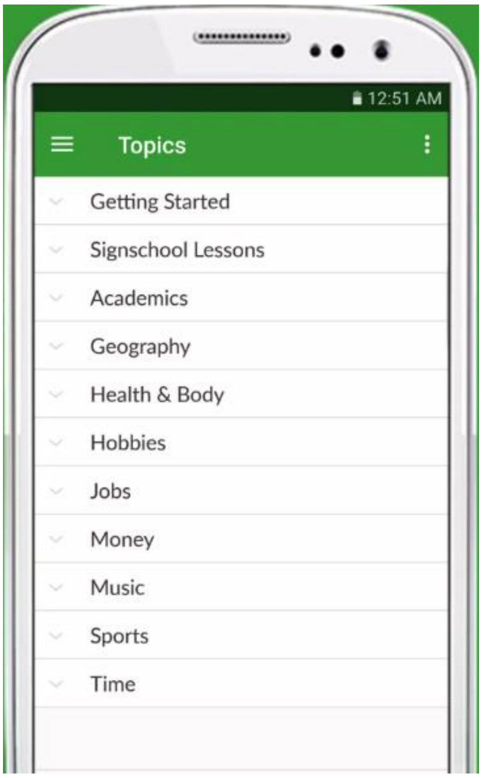


Figure 2. 2 Desired Topics section of SignSchool App [10]

Strengths

SignSchool also has many strengths as an educational tool for people seeking to begin their ASL journey through self-guided means. One major advantage is the consistent reliance on high-quality **video lessons** throughout the entire learning process. For a primarily visual language like ASL, seeing fluent signers model handshapes, motions, facial expressions and placement is absolutely essential since those kinetic elements cannot be conveyed through still images or text alone.

Based on [9], one of the greatest strengths of SignSchool is that it offers materials across a **wide range of difficulty levels to suit learners of all experience levels** and learning styles, this is shown in Figure 2.3. Users can start with basic foundational lessons introduced at a beginning ASL level and progress sequentially through intermediate and advanced content. Within each level, concepts are broken down into easy-to-understand mini lessons that gradually increase in complexity. This scaffolding approach allows learners to build confidence with low-risk early successes before advancing. But it also respects more seasoned students by providing robust material above introductory basics. With full courses from absolute novice to fluent signer, the breadth of difficulties ensures long-term engagement as skills improve over time.

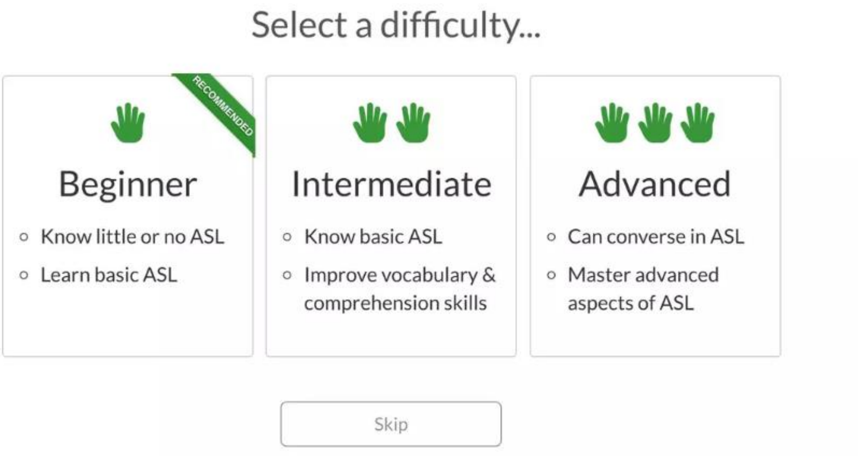


Figure 2. 3 Difficulty level selection of SignSchool App [10]

As in Figure 2.4, SignSchool amplifies engagement and long-term retention through a collection of online **games and quizzes** [9]**.** Formative assessments utilize interactive flashcards, pictured vocabulary drills, and sequenced story reconstructing exercises. Learners achieve flow states merging focus with enjoyment through puzzle solving, matching, interpreting errors to correct and cooperative challenges. The assessments draw on various learning styles beyond isolated cramming to maximize real comprehension.

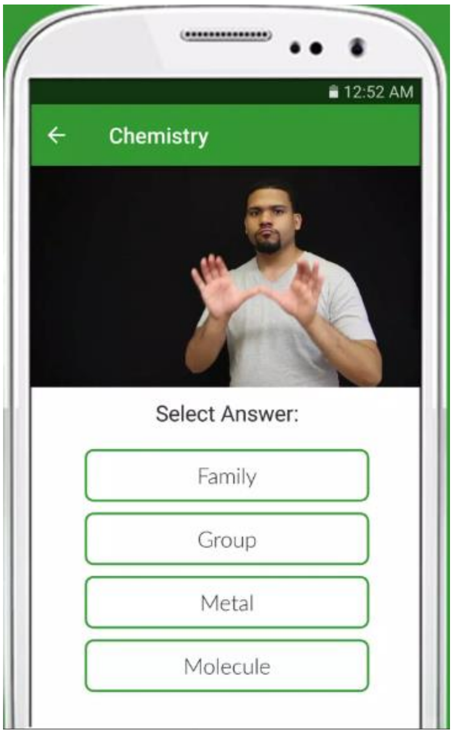


Figure 2. 4 Interactive Quiz of SignSchool App [10]

Limitations

The first major limitation of SignSchool is that it focuses solely on video-based independent learning rather than offering any **in-person instructional opportunities**. While the video lessons and demonstrations are very high quality, there are certain language acquisition benefits that cannot be replicated through an app alone. For one, users do not get the opportunity for immediate feedback if they are signing something incorrectly. In a classroom setting, a teacher can observe a student's motions and facial expressions up close and provide guidance on the spot.

According to [8], the second key limitation is that SignSchool is designed **exclusively for learning American Sign Language** rather than expanding its scope to include other signed languages from around the world. With deaf communities and signed languages existing all over the globe, limiting instruction to only ASL means SignSchool has a relatively narrow target audience and language learning goals. While exposure to Deaf culture aspects through ASL lessons is valuable, true language learners may be looking to immerse themselves in the cultural practices and perspectives associated with other signed tongues. In today's interconnected world where international travel and cultural exchange are increasingly common, having a more international approach could boost SignSchool's appeal and give it more broad utility. Expanding the platform to gradually incorporate other major signed languages over time, such as British Sign Language, French Sign Language, or Asian signed forms would significantly enhance its value for multifaceted language enthusiasts.

2.1.2 Lingvano

The core function of Lingvano is to teach American Sign Language (ASL) through interactive video lessons [11]. The lessons are organized into few vocabulary modules that each focus on a different theme such as greetings, family, feelings, actions, and more. For each sign, the user sees the word in English text, and then a video of a native ASL signer demonstrating the sign. Figure 2.5 shows this function.

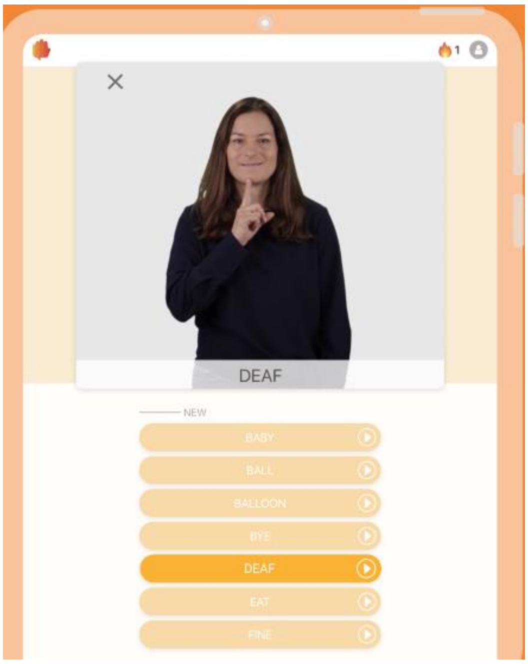


Figure 2. 5 Video Demonstration Function of Lingvano App [11]

Strengths

One of the major strengths of Lingvano is that it supports learning **multiple sign languag**es [8], not just American Sign Language. This allows users to choose the sign language that is most relevant for their needs and locale. Lessons, vocabulary, and communication tools are available for languages such as British Sign Language. The ability to learn the local sign language opens up opportunities to interact with the deaf community anywhere in the world. It also has the benefit of teaching signs that may align more closely with those used in other neighboring countries for an even broader audience.

According to [12], Lingvano takes an engaging, **bite-sized approach** to lessons through the use of short instructional **video clips and included learning games** (Figure 2.6). The videos are only a couple minutes long, making lessons easy to fit into a quick study session. This prevents sign language acquisition from feeling like a large tedious task. Games like matching signs to their written words or fingerspelled equivalents reinforce learning in a low-pressure, fun way. They help keep users coming back to practice more. The concise, game-based format maintains interest and encourages regular practice for faster skill building over time.

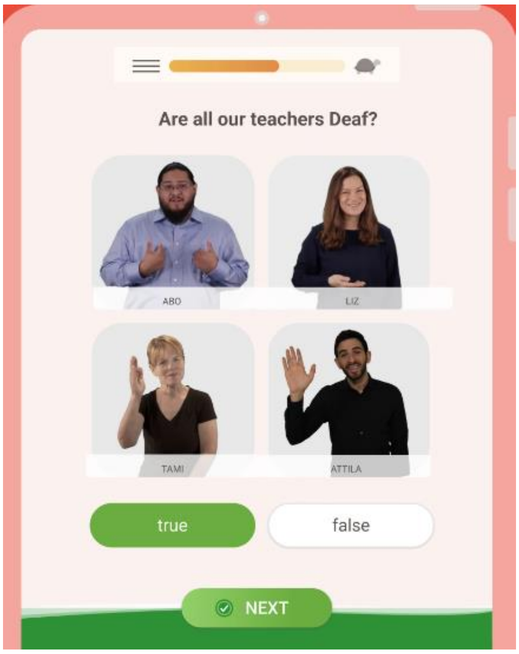


Figure 2. 6 Learning Games of Lingvano App [11]

One feature that supports ongoing learning is the **dictionary of vocabulary** as in Figure 2.7 that can be instantly accessed [8]. Rather than limiting users only to memorizing predefined lesson lists, the sign language dictionary opens up independent discovery. It includes a database of searched terms that can be looked up using English keywords. This provides students and advanced learners a valuable resource for understanding signs encountered in real conversations.

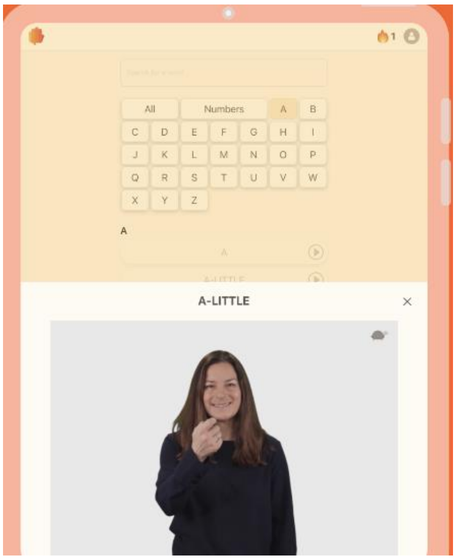


Figure 2. 7 Dictionary of Vocabulary in Lingvano App [11]

A new strength is that Lingvano provides **realistic sign language dialogue** as shown in Figure 2.8. Rather than isolated words, users can engage in natural conversations. This immerses users in authentic language immersion, more closely mimicking real-world interactions. It allows assessment and improvement of comprehensive conversational competence rather than just memorization. The simulated dialogues strengthen preparation for communicating with Deaf communities.

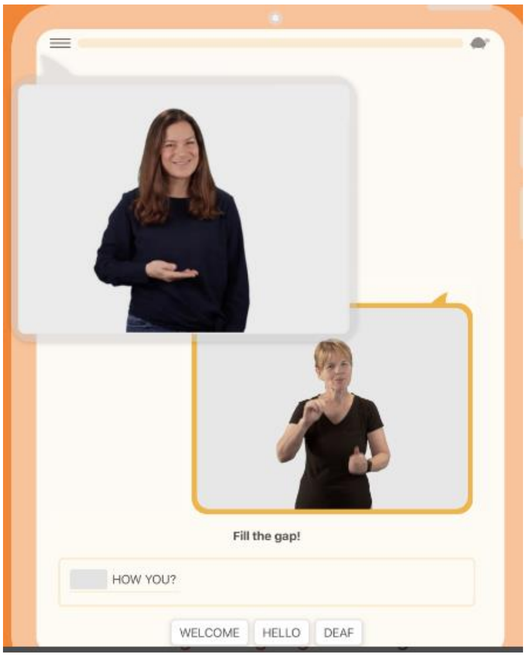


Figure 2. 8 Realistic Sign Dialogue in Lingvano App [11]

Limitations

One limitation of Lingvano compared to in-person instruction is the **lack of real-time feedback** during practice. When recording and comparing signing exercises, users do not receive immediate guidance, correction or reinforcement from a native signing teacher. They must self-assess their abilities, which is not always reliable. Without real time recognition and feedback, mistakes may solidify before receiving clarification later on. This could potentially slow the learning process.

While basic functionality is free to access, advanced features that elevate the learning experience like grammar lessons and unlimited conversations **require a monthly subscription**. However, the costs may not be financially feasible for all users. For individuals and families operating on tight budgets, this could present a limitation. This creates a barrier preventing some motivated learners from achieving higher competency.

2.1.3 Summary of existing system

Table 2.1 summarises all the existing Sign Language Learning Mobile Application mentioned in the previous subsection in terms of strengths and limitations.

Table 2. 1 Comparison table of Existing Application

Application	**Strength**	**Limitations**
SignSchool	- Use of video lessons - wide range of difficulty levels to suit learners of all experience levels - consists of interactive games and quizzes	- lack of real-time feedback - exclusively for learning American Sign Language only
Lingvano	- multiple sign languages - bite-sized approach, reinforce learning in a low-pressure - use of video clips	- Lack of real-time feedback - require monthly subscription

	- include interactive learning games - consist of dictionary of vocabulary - provides realistic sign language dialogue	

2.2 Review of Previous work on Sign language Detection and Recognition

2.2.1 Sign Language Detection and Recognition using MediaPipe and SVM

The paper [13] proposes a methodology for real-time sign language recognition using MediaPipe, for hand tracking and machine learning algorithms, specifically SVM (Support Vector Machine), for classification. Generally, the architecture proposed in this paper is divided into 3 main stages as shown in Figure 2.9*:*

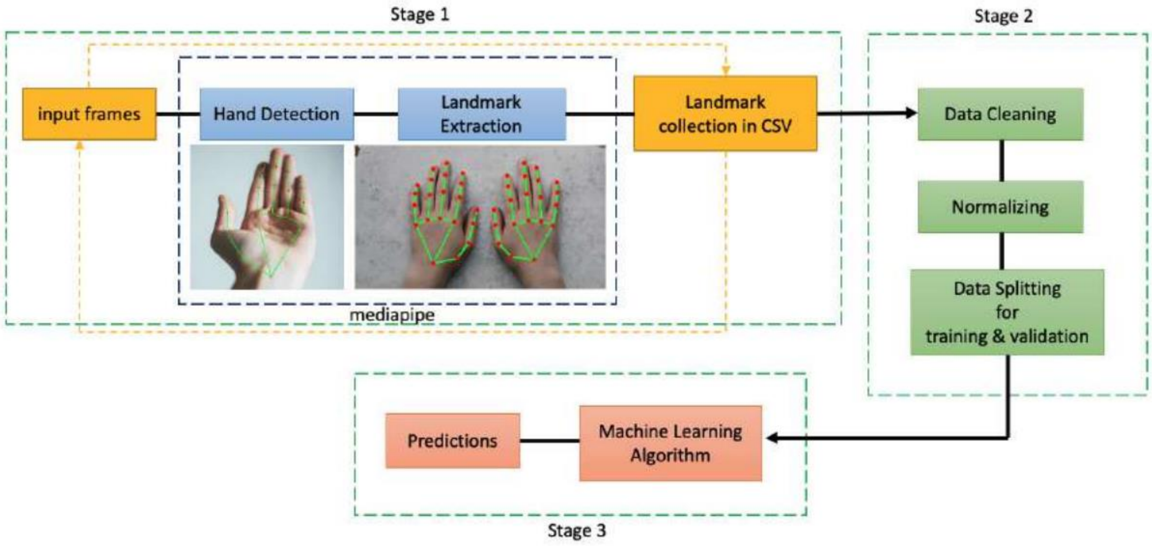


Figure 2. 9 Proposed architecture to detect handgestures and predict sign language finger-spellings

I) Pre-processing of Images using MediaPipe

MediaPipe provides pre-trained machine learning models for detecting and tracking human body parts including hands. It uses two models: Palm Detection Model and Landmark Model. This model accurately localizes 21 3D hand-knuckle coordinates for all fingers, palm and wrist points even when the hand is partially visible [4]. The palm and hand detection approach simplifies hand tracking compared to directly detecting hands from the full image. By first estimating palm regions which are less complex, it provides better context and reduces the need for extensive data augmentation like rotations and flipping. This allows the model to dedicate more power for accurate landmark localization.

The pre-trained palm and hand detection models are applied to the different sign language image datasets. The 21 landmark points output for each image is stored in CSV format indexed by the alphabet or number folder. According to [13], only the x,y coordinates are retained for training the machine learning model later.

II) Data Cleaning and Normalization

The x,y coordinates extracted from all the dataset images are collected into one CSV file. This file is processed using pandas library in python to remove any null entries. Null entries occur when the hand is not clearly detected in some blurry images. Removing these null values prevents bias during model training [14].

The entries are then normalized to rescale the values to fit the system. The cleaned data file is split into 80% for training and 20% for validation of the machine learning model.

III) Prediction using Machine Learning (SVM)

Predictive analysis is performed by training and evaluating different machine learning algorithms, the results are shown in Figure 2.10, SVM gave the highest average accuracy of 99% for most sign language datasets, outperforming the other algorithms. Based on [13], some reasons why SVM is effective are that it performs well in high dimensional spaces and large training data scenarios. The kernel used in this paper is Radial Basis Function (RBF) which maps inputs to higher dimensions for better separability. SVM works well when there is a clear margin of separation between classes, which is applicable for the sign language gestures. Hence, SVM is used to classify the different sign language classes.

Dataset	SVM	KNN	Random Forest	Decision Tree	Naive Bayes	ANN	MLP
ASL(alphabet)	**99.15%**	98.21%	98.57%	98.57%	53.74%	97.12%	94.69%
Indian(alphabet)	**99.29%**	98.87%	98.59%	98.59%	86.77%	94.79%	96.48%
Italian(alphabet)	**98.19%**	96.75%	97.83%	97.83%	77.19%	78.63%	72.14%
ASL(numbers)	**99.18%**	99.18%	97.56%	97.56%	96.74%	95.12%	97.56%
Turkey (numbers)	**96.22%**	93.08%	94.33%	94.33%	83.64%	93.71%	83.64%

Figure 2. 10 Average accuracy obtained using machine learning and deep learning algorithms. [13]

Quantitative analysis of the model is done by evaluating performance metrics like accuracy, precision, recall, F1 score and confusion matrix. According to Figure 2.11, the maximum accuracy of 99.29% is obtained for the Indian sign language alphabet dataset while the minimum accuracy is 96.22% on the Turkey sign language numbers dataset.

Dataset name	Training Accuracy	Testing Accuracy	Precision	Recall	F1-Score
ASL(alphabet)	99.50%	99.15%	99.15%	99.15%	99.15%
Indian(alphabet)	99.92%	99.29%	99.29%	99.29%	99.29%
Italian(alphabet)	99.72%	98.19%	98.19%	98.19%	98.19%
Turkey (numbers)	99.37%	96.22%	96.22%	96.22%	96.22%
American (numbers)	98.77%	99.18%	99.18%	99.18%	99.18%

Figure 2. 11 Performance analysis using SVM algorithm on different datasets [13]

Strengths

The proposed method has several benefits and strengths. It is able to achieve remarkably **high average accuracy** of 99% across multiple sign language datasets, demonstrating the robustness of this method.

Besides, the **lightweight** nature of the model enables easy deployment on smartphones, making the sign language recognition solution highly accessible.

Additionally, the method **does not require any extra hardware** like sensors or data gloves, making the solution very convenient and low-cost. It has potential to be expanded to recognize dynamic sign language videos containing word and sentence sequences, beyond just static alphabet and number images.

Limitations

The current approach has some limitations that could be addressed. It is restricted to identifying static individual gestures in images rather than continuous sign language in video format.

Performance may deteriorate when used in complex backgrounds and varying lighting conditions compared to a controlled setup.

Furthermore, any addition of new gestures to the model requires a retraining process.

2.2.2 Sign Language Detection and Recognition using CNN and RNN(LSTM)

The paper [15] proposes a model using convolutional neural network (CNN) and recurrent neural network (RNN) for American Sign Language (ASL) recognition from video sequences. The proposed model takes video sequences as input and extracts both temporal and spatial features. A CNN is used to extract spatial features from each frame, recognizing the gestures spatially. Then an RNN called LSTM is used to extract temporal features by learning the sequence of gestures over time from the CNN outputs. The combined model is trained and evaluated on a custom ASL dataset. The overall architecture of this method is shown in Figure 2.12 below.

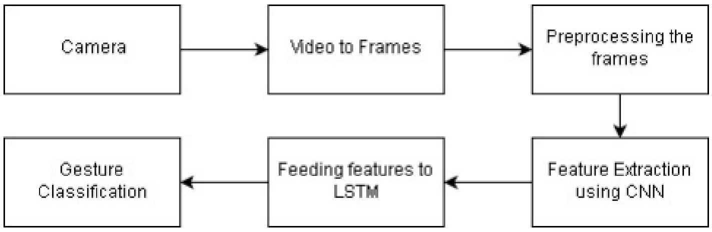


Figure 2. 12 High Level System Architecture

Video frames are fed into an Inception CNN pre-trained on ImageNet to extract spatial features. Inception was chosen as it is widely regarded as one of the best image classifiers. For gesture detection, the paper [15] proposes two approaches using the CNN outputs - either using the predictions from the SoftMax layer, which produces a normalized probability distribution over predicted classes, or using the output of the global pooling layer, which aggregates feature maps and produces a fixed-length output vector per image. The global pooling layer allows for more detailed features to be extracted and analysed by the subsequent RNN compared to the SoftMax predictions. The feature sequences extracted from either the SoftMax or pooling layer are then fed to an LSTM RNN to model temporal dependencies and classify the gesture sequences.

For gesture classification, the paper discusses using an LSTM RNN architecture to classify the gestures based on the sequence of frames. Since the input sequences need to be of fixed length, all frame sequences are trimmed to the same length. Based on [18, 19], a LSTM network is used due to its ability to capture long-term dependencies in sequence data compared to simpler RNNs. The LSTM is trained on the CNN outputs from either the softmax layer or global pooling layer. The model is evaluated with varying numbers of gestures in the dataset, showing the SoftMax layer approach achieving higher accuracy compared to using the global pooling layer outputs (Figure 2.13*)*.

# of Signs	Accuracy with Softmax Layer	Accuracy with Pool Layer
10	90%	55%
50	92%	58%
100	93%	58%
150	91%	55%

Figure 2. 13 Performance With Varying Sample Sizes. [15]

The test set is doubled via augmentation to gather more predictions. Accuracy is reported over two variations of the same gesture. With 10-150 gestures, the softmax approach achieves 90-93% accuracy while the pooling approach shows lower 55-58% accuracy.

Strengths

The RNN captures temporal relationships well for video-based gesture recognition. Besides, applying transfer learning removes the need for very large training sets. The modular pipeline in this approach also allows tuning each component's architecture separately.

Limitations

This proposed approach may suffer loss of accuracy due to the varying facial features and skin tones. The accuracy has dropped when tested with different skin tone that has not been trained on. Besides, the variation of clothing also affects the accuracy.

2.2.3 Sign Language Detection and Recognition using MediaPipe and RNN(LSTM)

The overall architecture of the proposed method in paper [17] is as shown in Figure 2.14 involves using a webcam as input to capture video of a person signing. The video frames are fed into the MediaPipe framework to detect human pose landmarks and hand landmarks on each frame. These landmarks are pre-processed and formatted as sequences of frames. The sequences are then input to a recurrent neural network model with LSTM layers for classifying the sign language gestures.

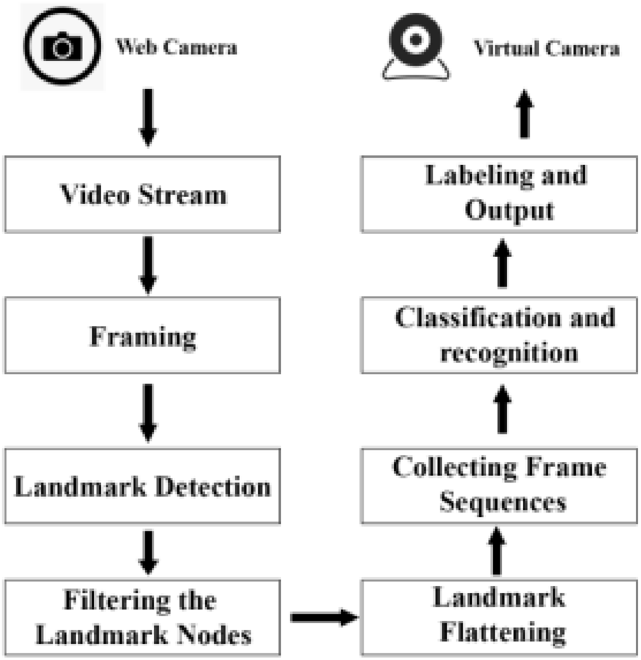


Figure 2. 14 Architecture of proposed Sign Language using MediaPipe and LSTM

The video frames from the webcam are fed into Google's MediaPipe framework for analysis. MediaPipe provides pre-trained machine learning models for detecting face, hand, and human pose landmarks in images or video streams [4]. The raw outputs from MediaPipe consist of large arrays of 3D (x,y,z) coordinates for each landmark point on the detected face, hands, and body. A series of preprocessing steps are applied to format these landmarks into sequential data for input into the

The pre-processed landmark sequences are fed into a Recurrent Neural Network (RNN) to classify the sign language gestures. RNNs are a family of neural networks well-suited for modelling sequential data [17]. Specifically, a type of RNN called Long Short-Term Memory (LSTM) is used, which avoids some of the vanishing gradient problems of basic RNNs [16]. The authors stack multiple LSTM layers to create a deep network architecture. neural network model. The authors in [17] implement a 6-layer neural network comprised of 3 LSTM layers and 3 Dense layers. This combines the sequence modelling of the LSTM layers with the classification outputs of the Dense layers. The first 5 layers use the ReLU activation function. This introduces non-linearities without being as computationally intensive as deeper LSTM layers. The final Dense layer uses SoftMax activation to squashes the outputs into probability scores for each class to predict the sign label. The Adam optimization algorithm is used during training to iteratively tune the weights and biases of the network to minimize classification errors.

Strengths

Achieves high accuracy for small vocabulary continuous sign recognition. Besieds, this method handles multiple signers with some robustness to variations. Furthermore, MediaPipe provides efficient landmark tracking as input. Lastly, this method conducts comprehensive evaluation with multiple rounds of training and testing.

Limitations

When the number of signers increases, the detection accuracy will be affected. Besides, different light conditions were also tested, low-light condition will cause a decrease in accuracy.

2.2.4 Sign Language Detection and Recognition using Mediapipe holistic and RNN(LSTM)

The paper [18] proposes a vision-based real time Bangla sign language recognition system using MediaPipe Holistic and LSTM. The goal is to develop a system that can convert Bangla sign language gestures into text to facilitate communication between deaf-mute individuals and ordinary people. The proposed methodology is as shown in Figure 2.15 below.

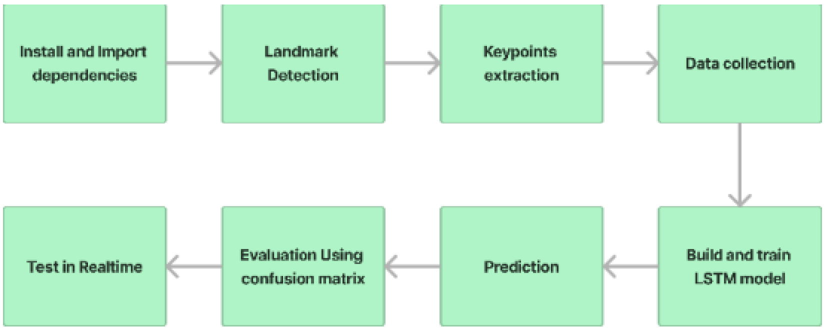


Figure 2. 15 Architecture of proposed Sign Language using MediaPipe Holistic and LSTM

MediaPipe Holistic is used to detect pose landmarks and hand landmarks from image frames [18]. The output provides X, Y, Z coordinates for 33 pose keypoints and 21 hand keypoints per frame. This enables accurate tracking of body and hand motions in real-time and is used to extract frame-wise keypoint data for model training and real-time testing.

LSTM (Long Short-Term Memory) is a type of recurrent neural network well-suited for sequence data like video [19, 20, 21]. It maintains an internal memory state that gets updated based on new inputs. Besides, it also has mechanisms to remember long-term dependencies in the data. Thus, it is suitable to be used to train a model to recognize sequential hand gesture patterns. It will first classify each frame's keypoints into one of the 15 trained gesture classes. The results of this methodology has achieved a high accuracy, which is 99.97% for training data and 98.88% for test data.

Strengths

This approach has achieved a high accuracy over 98% training and test accuracy with the LSTM model. Besides, it maintains performance even on unseen test data. Furthermore, the real-time performance of this method helps to recognize gestures from live video feed with low latency.

Limitations

Small Dataset is used in this paper, a small dataset of only 22500 frames across 15 classes is used. Thus, it is prone to overfitting on the limited data samples. More varied training data needed for generalizable performance. This model trained only on 15 common words and phrases. It cannot recognize the large vocabulary of a complete sign language.

2.2.5 Summary of the Previous work on Sign language Detection and Recognition

Table 2.2 summarises all the Previous work on Sign language Detection and Recognition mentioned in the previous subsection in terms of strengths and limitations.

Table 2. 2 Comparison table of previous work

Methods	Strengths	Limitations
MediaPipe + SVM	- high average accuracy - lightweight - does not require any extra hardware	- restricted to identifying static individual gestures - performance affected by complex background and varying light conditions - new gestures require retraining

CNN + RNN(LSTM)	- well suited for video-based gesture recognition - large training sets not needed - allows tuning each component's architecture separately	- may suffer loss of accuracy due to varying facial features and skin tones.
MediaPipe + RNN(LSTM)	- high accuracy for small continuous sign recognition - handles multiple signers with some robustness to variations. - MediaPipe provides efficient landmark tracking	- when number of signers increases, detection accuracy will be affected. - do not perform well on low-light condition
MediaPipe Holistic + RNN(LSTM)	- High accuracy - maintains performance even on unseen test data - real-time performance to recognize gesture from live video with low latency	- Small dataset used, prone to overfitting on the limited dataset - trained only on 15 common words and phrases, cannot recognize large vocabulary

CHAPTER 3 SYSTEM METHODOLODY/ APPROACH

3.1 Methodology and General Workflow Procedures

This project will be divided into six phases, which include project pre-development, model training for sign language detection and recognition, app development, content development, testing and refinement, and lastly the completed system. Figure 3.1 shows the overall development phase of the project.

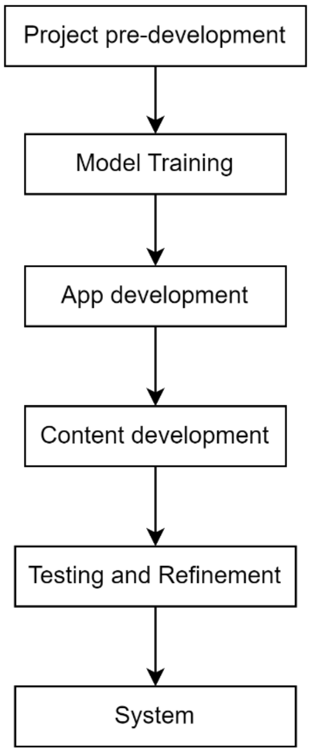


Figure 3. 1 Overall Development Phase

3.1.1 Project pre-development

Extensive research will be done to study the current state of sign language detection technologies. This will include reviewing academic papers on techniques such as hand key point detection using mediaPipe and sign classification using deep learning models. Existing publicly available datasets like ASL, BSL will be analysed for their suitability and coverage. Competitive analysis of leading apps in the space will provide insights into gaps to address. Finally, project objectives will be defined along with technical requirements based on available solutions and user needs. A project methodology is defined in this phase, which is the prototype methodology that is shown in Figure 3.2.

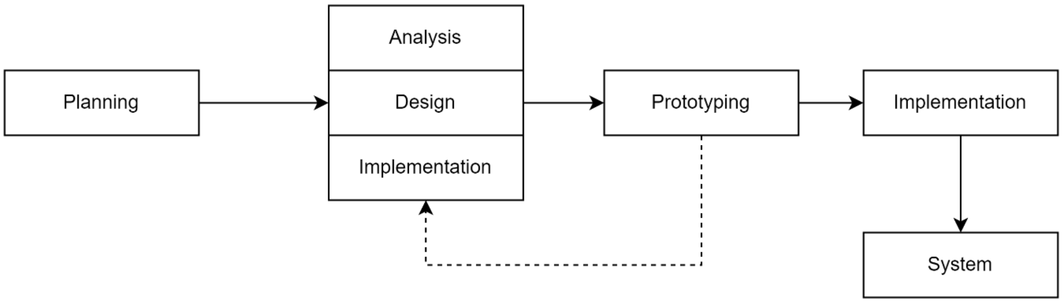


Figure 3. 2 Prototype Methodology

In this methodology, the problem and opportunity through research on sign language are defined in the planning phase. Besides, the project goals, scope and requirements are also defined. In the analysis phase, solutions brainstorming, and competitive analysis of existing applications are conducted to define the features and priorities of the system. As for the design phase, a low-fidelity paper prototype for key screen and flow will be produced to get feedback from target user, followed by the implementation phase to continuously test the prototypes with user and iterate. Next, in the system prototype phase, we will build vertical slices of the app to include key features such as quizzes and recognition demos. This will help to gather feedback on look, feel and core functionality. After being satisfied with the prototype, an implementation phase is conducted to develop a full-fledged module by connecting the modules and integrating MediaPipe for computer vision. This will be refined through continuous integration and testing. Lastly, the system will gradually develop completed features for users and continue to improve based on data and usability issues.

3.1.2 Model Training

MediaPipe provides pre-trained models that will be leveraged for hand and keypoint detection. MediaPipe is an open-source framework developed by Google that contains powerful computer vision and machine learning models to solve various problems in real-time. For this project, MediaPipe's hand detection and multi-hand tracking models will be used. These models have been trained on large datasets to accurately detect hands present in images or video frames and track 21 3D key points per hand including wrist, palm center and finger joints as shown in Figure 3.3. Being pre-trained, these MediaPipe models can run very efficiently on device without much overhead of customized training.

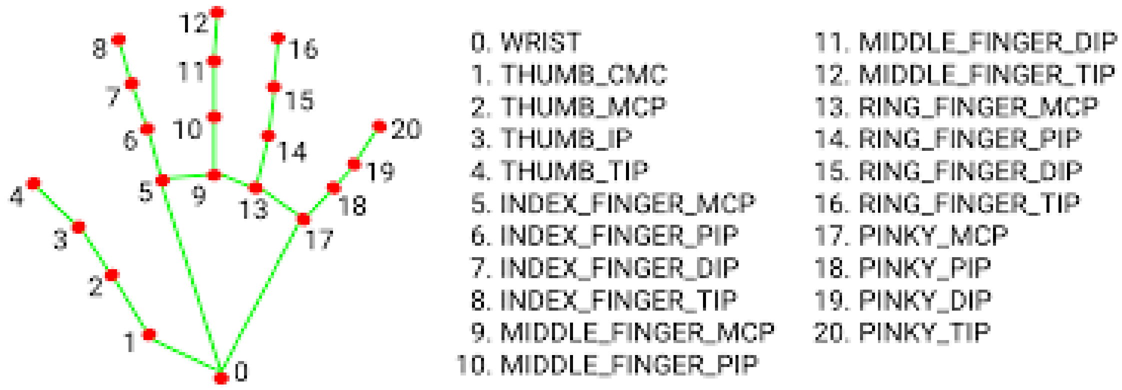


Figure 3. 3 Detected hands and associated 21 keypoints [19]

The detected hands and associated 21 keypoints will form the foundational input for the sign language recognition models. These keypoints capture the critical positional information about hand shape and configuration which is essential to distinguish different signs as shown in Figure 3.4. For example, the bend of specific finger joints can completely change the meaning of a sign. Therefore, instead of just using raw image pixels, focusing the model on these abstracted keypoint coordinates allows it to learn more relevant patterns. This project's custom sign language datasets will contain thousands of images of individual signs annotated with the corresponding hand keypoint data. These datasets will be used to train robust machine learning models for mapping specific hand poses to distinct signs.

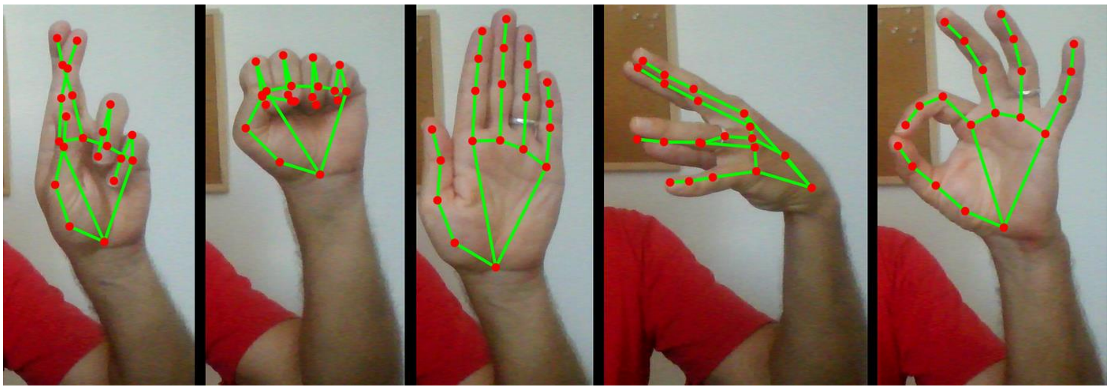


Figure 3. 4 Example of Keypoints capture using MediaPipe [20]

After collecting the list of keypoints, the data will be normalized. Then a deep learning model is trained. After the model has been trained, we assess its performance using validation set to make sure it is capable of accurately recognise the sign languages. We **evaluate its**

precision, recall, accuracy and F1 score to evaluate how well the model can recognise the sign language. The final optimized models can then be deployed for real-time sign recognition on incoming MediaPipe keypoint data streams.

3.1.3 App Development

Professional UI/UX designers will design intuitive, accessible screens following brand guidelines. In this phase, we will focus on building essential features with appropriate caching and error handling. A modular architecture will ensure flexibility and scalability. MediaPipe will be integrated for on-device keypoint tracking. Machine learning models optimized in prior phase will be deployed for real-time sign recognition.

3.1.4 Content Development

Lessons will be a key part of the learning content. They will be developed to cover foundational signs, grammar structures and build core vocabulary categorized by topic area. Both videos and texts will be created so users can engage with the content in their preferred format. creating lessons for multiple sign languages, translations will need to be carefully vetted by relevant language communities.

3.1.5 Testing and Refinement

Testing will be crucial phase to delivering a high-quality user experience. During testing, emphasis will be placed on identifying and fixing bugs. Unit testing will evaluate individual features to verify they are functioning as intended before integration testing. Analytics will also be analysed to discover any confusing elements within the app.

3.2 System Design Diagram

3.2.1 System Architecture Diagram

Model-View-Controller (MVC) architecture as shown in Figure 3.5 is implemented for the sign language learning application, the Model consists of the SQLite database for local data storage and the TensorFlow Lite model for computer vision processing. The SQLite database manages user data, while the TensorFlow Lite model analyzes sign language gestures to provide real-time feedback. The View, constructed using XML layouts, handles the user interface, capturing video, and displaying feedback. The Controller, implemented in Java, acts as an intermediary, managing user input, communicating with the Model for data processing, and updating the View accordingly.

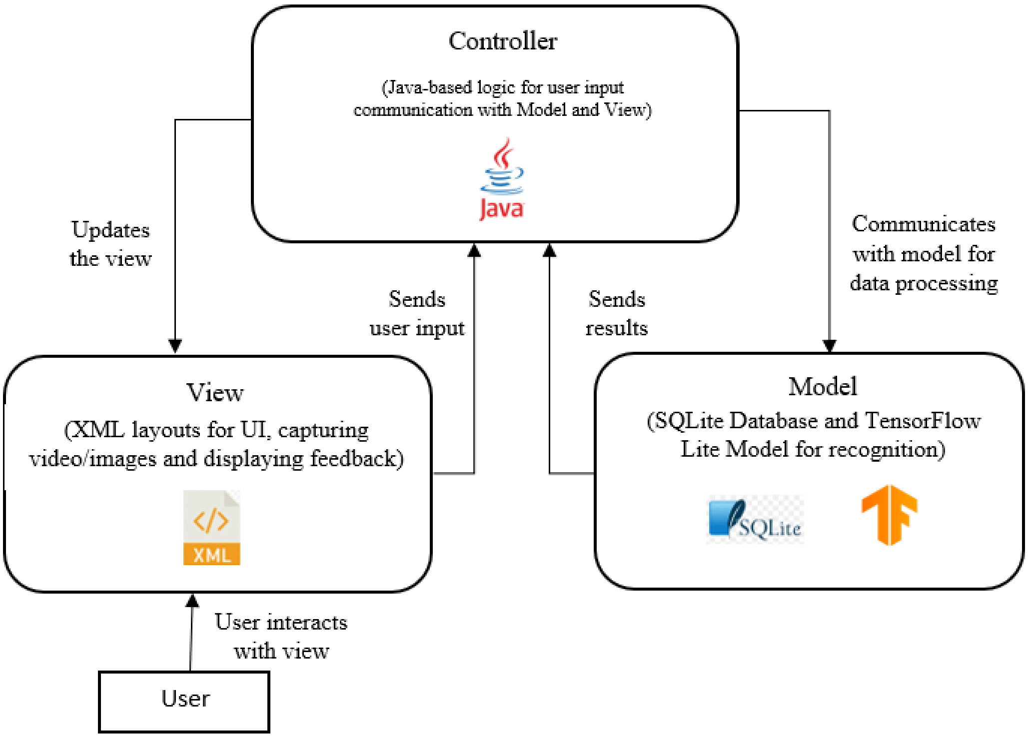


Figure 3. 5 System Architecture Diagram

3.2.2 Use Case Diagram and Description

Based on Figure 3.6, there is only one main actor in this system, which is the user, and a total of 7 main use cases. Users will be divided into 2 groups, which are new users and registered users. Firstly, the registered account use case will be required by new users, and the log in account use case will be required by registered users to access the other use cases. After signing in, the first use case of the system is the select speech-to-sign translator, where the user will translate their speech to text, and then view the video that corresponds to the text. Next, selecting real-time recognition is the second use case, where users will choose between their device's gallery or front camera to recognize their hand sign, followed by the prediction of sign class and comparison with correct data to display and highlight the key points as feedback to users. Another use case is when users select an interactive quiz. Users will select their desired category and view the video demonstration displayed, then users will select their answer and check the result. As for the use case where users select a learning module, users will also select a category and view the displayed video demonstration. Users may choose to try the real-time signing test where their key points will be compared and displayed on screen, and results will be calculated. The last use case is where users select user profile to view their personal information. Users may also enter new details to update their profile or select view progress to view their learning progress.

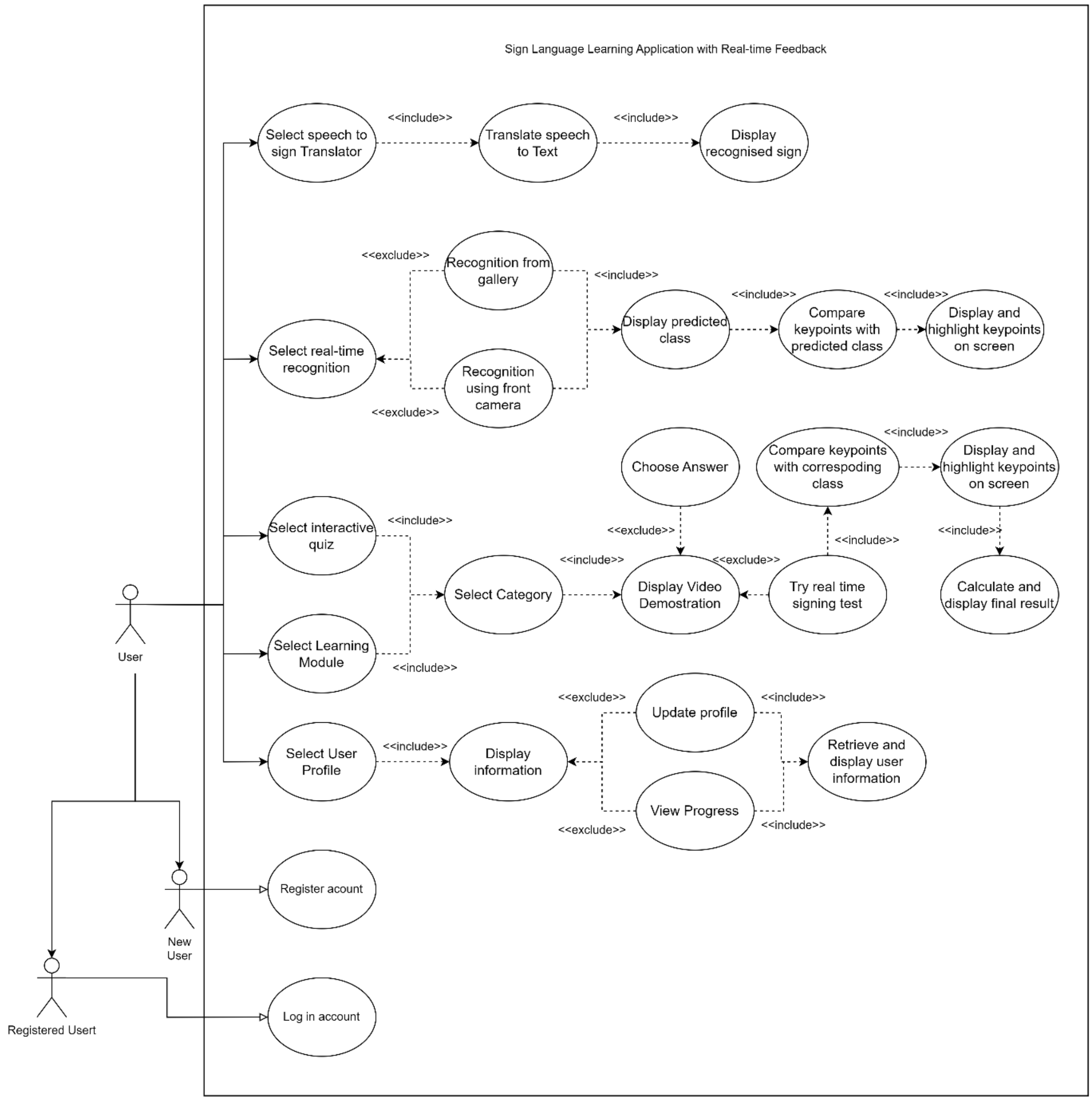


Figure 3. 6 Use Case Diagram

3.2.3 Activity Diagram

Sign In and Sign Up

Figure 3.7 shows the activity diagram of the sign in and sign up page. User will enter their username and password in the sign in page, and the system will retrieve the particular information from the database for verification. If the username and password match, user will be redirected to the main page. In the sign in page, user may also choose to proceed to the sign up page, where they will enter their username and password to be saved into the database as a new account.

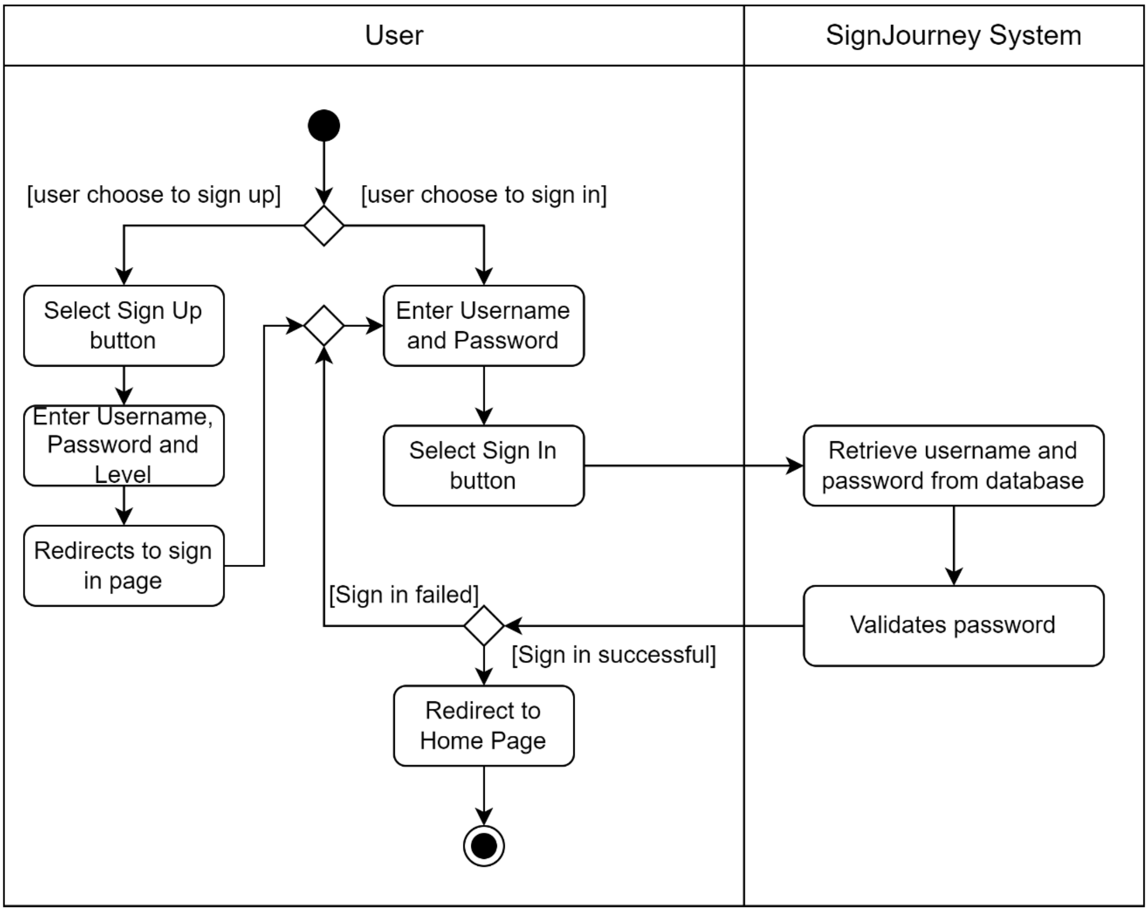


Figure 3. 7 Sign in and Sign Up Activity Diagram

User Profile

Figure 3.8 shows the activity diagram of the user profile. When users enter this page, the system will first retrieve the previously saved information and display it on the screen. If user wish to modify any of the information, they may enter it and press the update button so that the system will update the database.

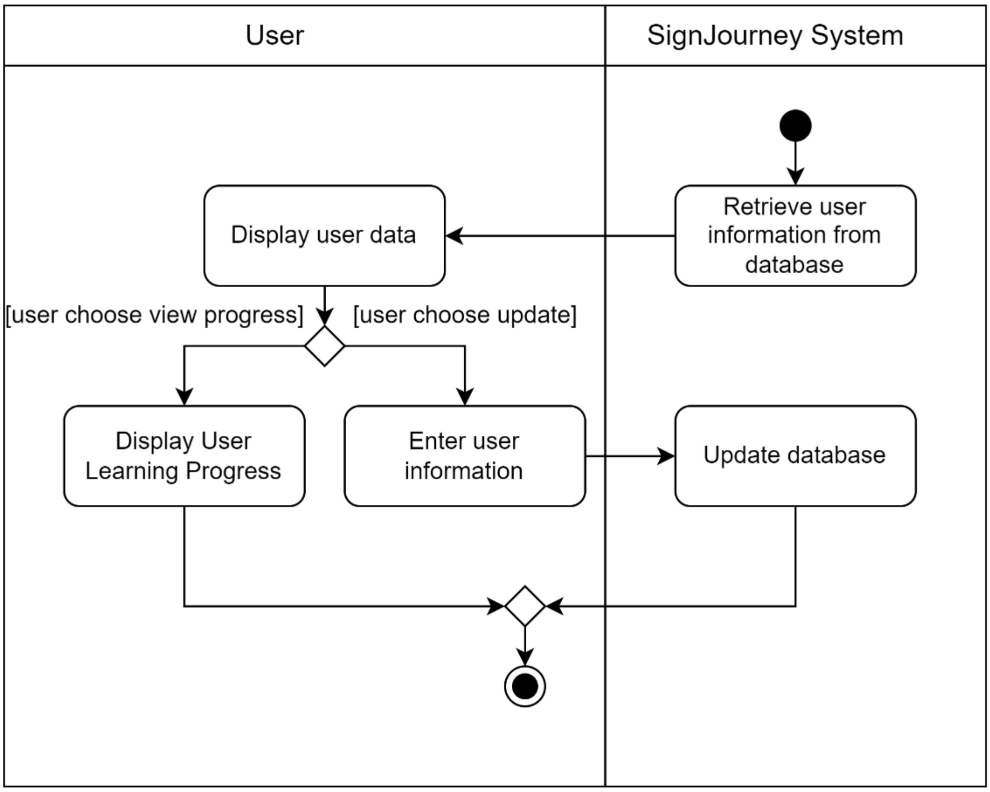


Figure 3. 8 User Profile Activity Diagram

Learning Modules

When users enter the learning page, they are required to select a desired category, then the related data will be retrieved. Video will be displayed automatically for each module. When users reaches the last video of the category, the end button will be selected, and they will be redirected back to the learning page.

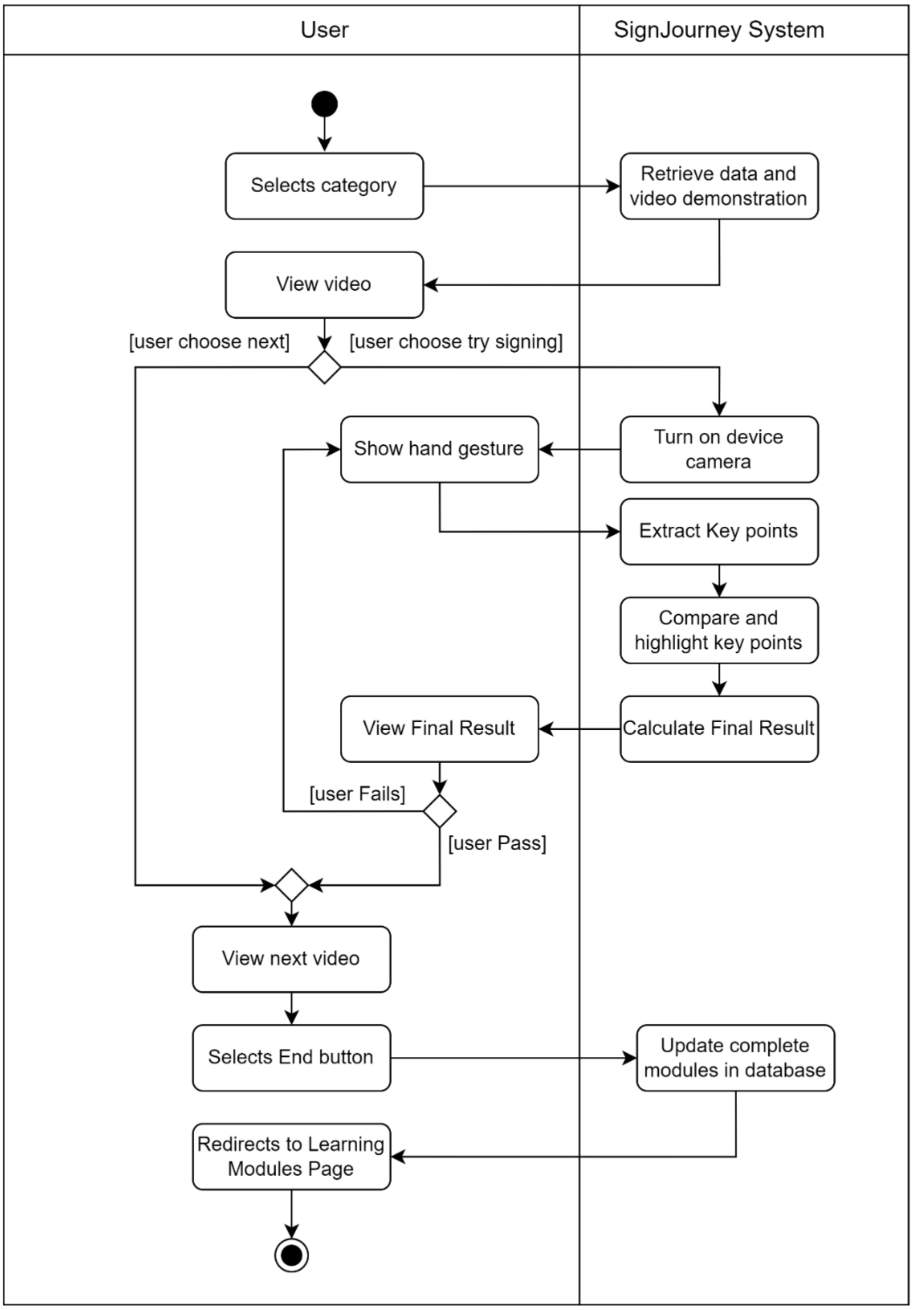


Figure 3. 9 Learning modules activity diagram

Quiz

In the quiz module page, users will select a category, and the system will retrieve the related data, including 3 other incorrect answers that are selected randomly. The video with 4 choices will be shown on the screen. If users select the correct answer, the buttons will turn green and they may proceed to the next screen. If user selects the incorrect answer, the buttons will turn red, and user may reselect their choice.

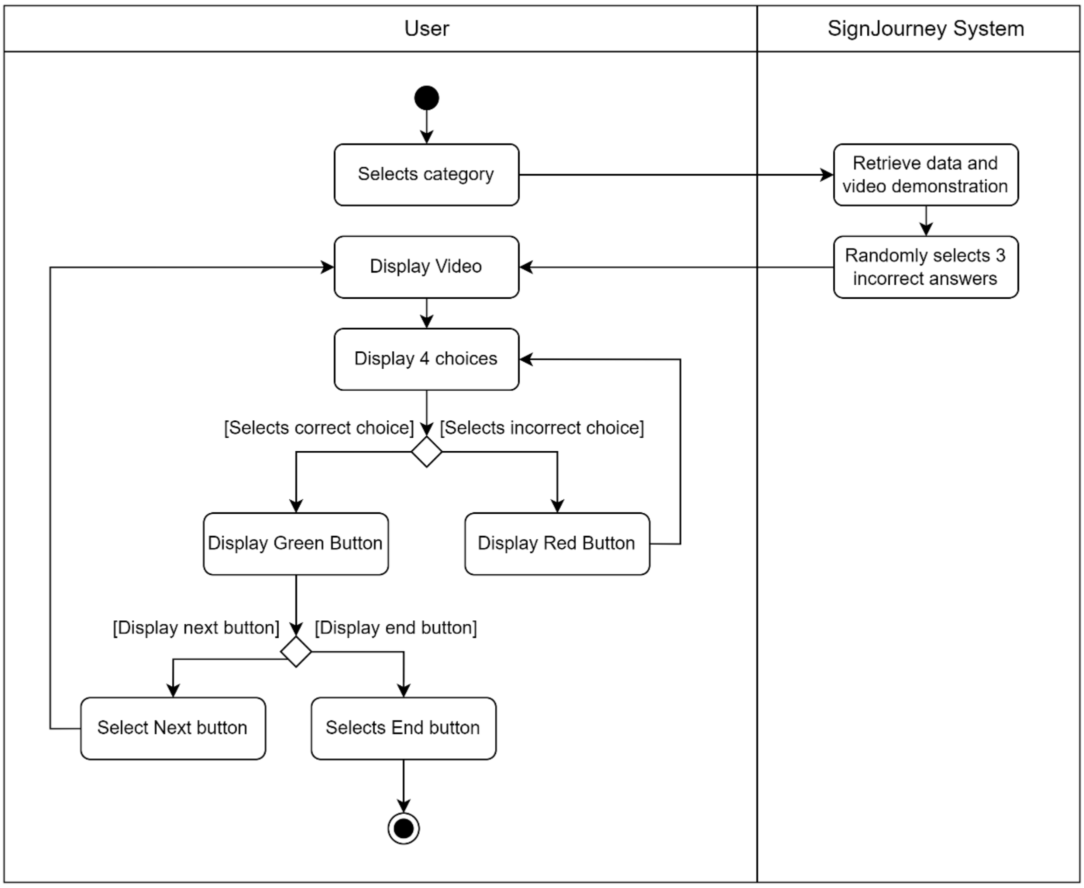


Figure 3. 10 Quiz module activity diagram

Real-time Recognition

When users enter the real-time recognition page, they may choose camera or gallery for the recognition. If camera is chosen, the system will turn on the camera, and user may show their hand gestures in front of the camera. If gallery was chosen, user may select an image from their gallery. Then, the key points from each frame will be retrieved and a prediction of the class will be made using the deep learning model. The predicted class will be displayed. Based on the predicted class, each key point will be compared to the one in the correct key points of the particular class. If the key point matches, green key point will be displayed, while red key point will be displayed if it does not match.

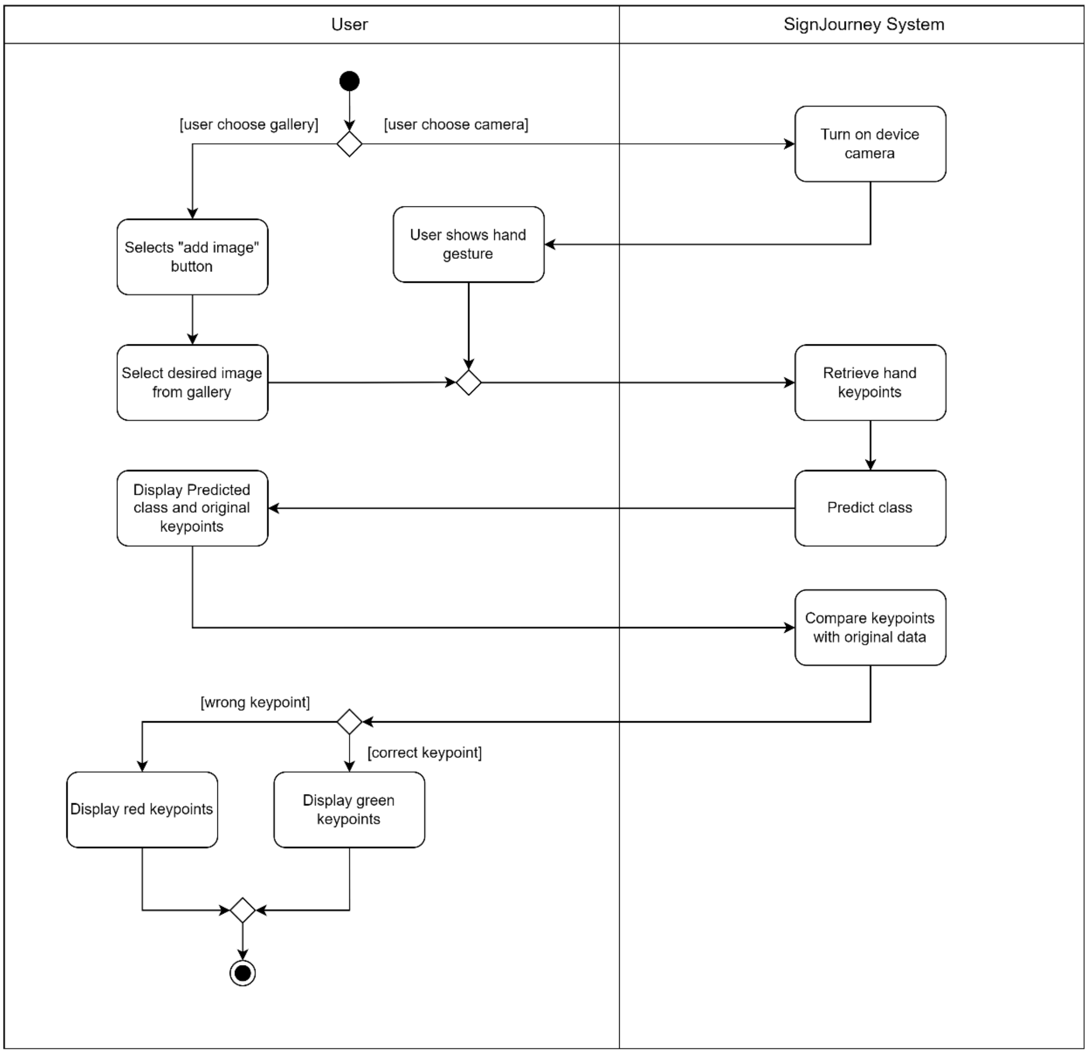


Figure 3. 11 Real-time recognition page activity diagram

Speech-to-Sign Translator

When user enters the Speech-to-Sign Translator page, they may select the "Speak" button to start their speech. The system will then utilize the Android SDK Speech Recognition API to recognize speech and translate speech to text. Using the text translated, the system will retrieve the corresponding video demonstration and display it to user. User may replay, pause, rewind or fast forward the video.

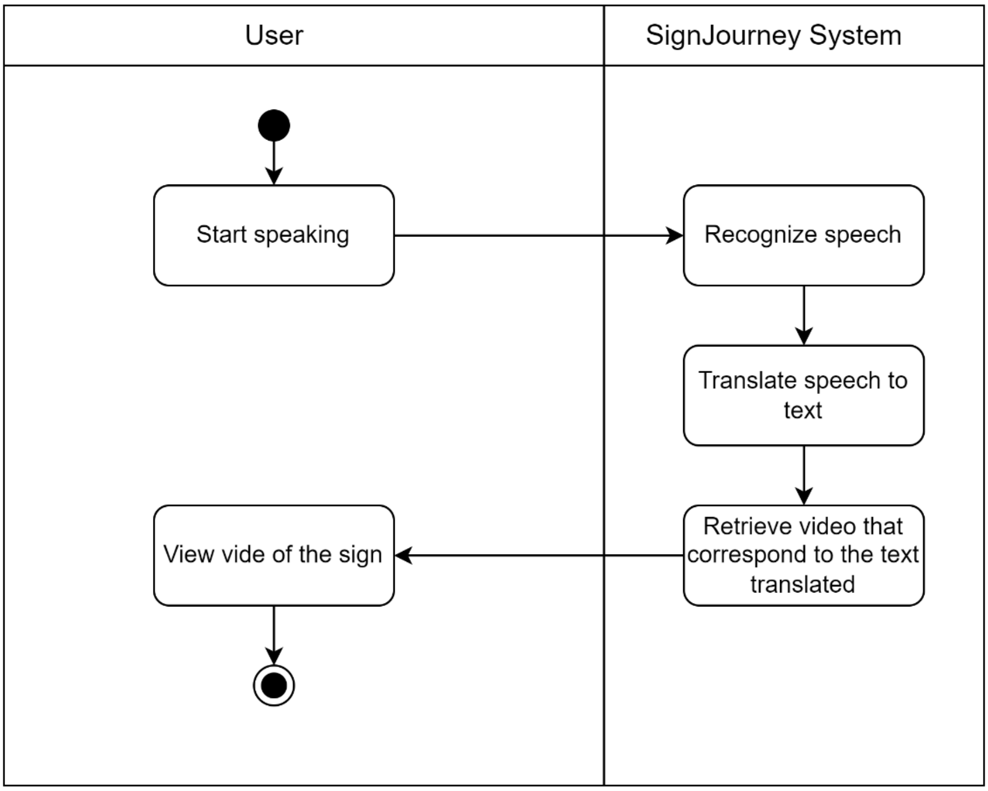


Figure 3. 12 Speech to Sign Activity Diagram

3.3 Project Timeline

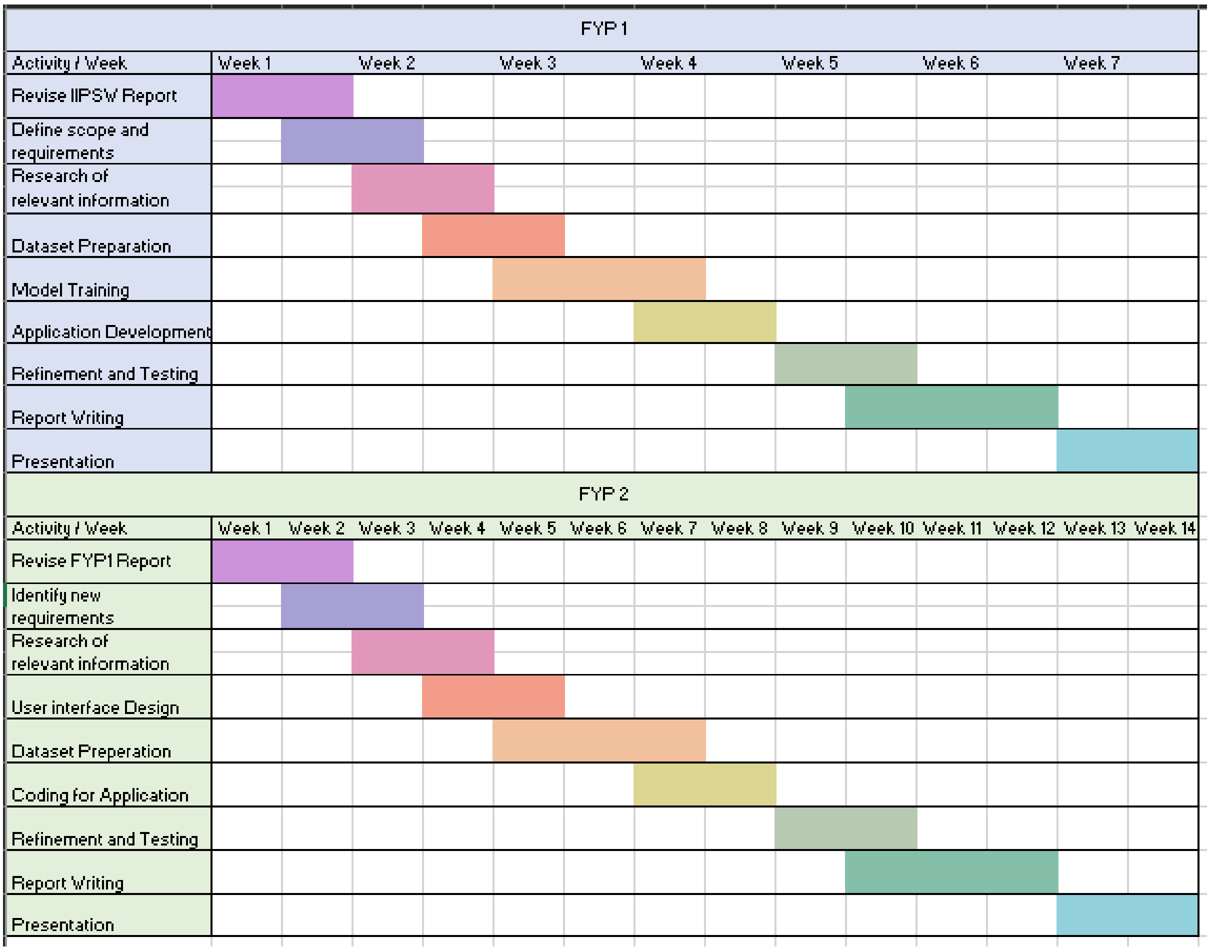


Figure 3. 13 Project Timeline

CHAPTER 4 System Design

4.1 System Flowchart

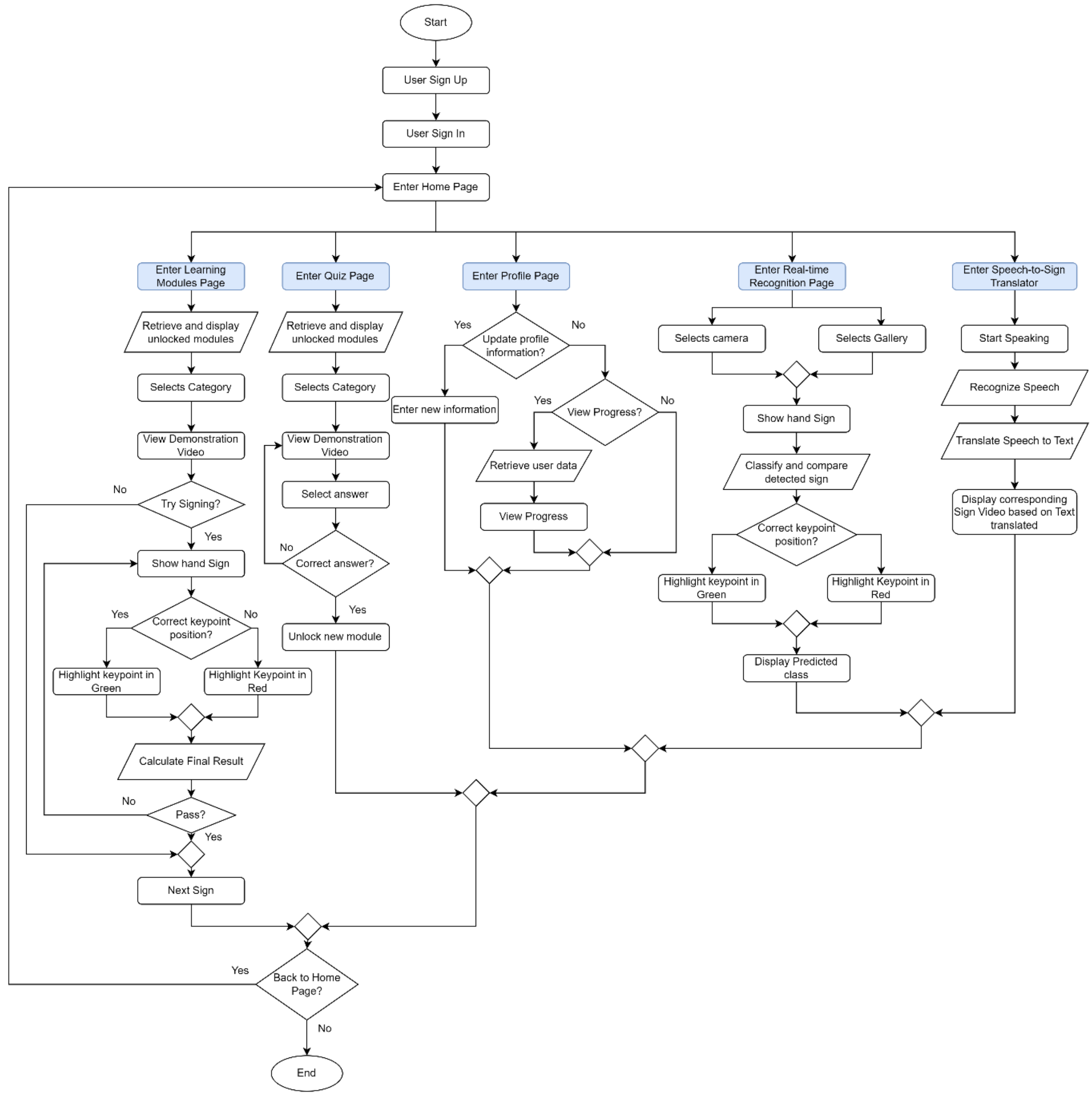


Figure 4. 1 System Flowchart

4.2 Database Design

Table: UserInfo

Table 4. 1 Database Design (UserInfo table)

Column Name	Data Type	Constraints	Description	Example value
ID	Integer	Primary key, Autoincrement	Unique identifier	1
USERNAME	Text	Not Null	Username of the user	Alexis_lee
PASSWORD	Text	Not Null	User's password	12345
EMAIL	Text	-	Email address of the user	alexis@gmail.com
NAME	Text	-	User's name	Alexis Lee
DOB	Text	-	User's date of birth	2002-09-21
AGE	Integer	-	User's age	22
LEVEL	Text	-	User's level	Intermediate
COMPLETED_MODULES	Integer	Default 1	Number of completed modules	3
COMPLETED_QIUZZES	Integer	Default 1	Number of completed quizzes	5

4.3 System Block Diagram

Figure 4.2 shows the system block diagram of the system. This high-level overview of the system architecture represents the key components of the system and their interactions with the system. In this system, there are 6 main components: Sign in and sign up, Learning Module, Quiz, Real-time static recognition, speech-to-sign and user profile section. The overview, data flow and key interactions of each component will be discussed in chapter 4.4.

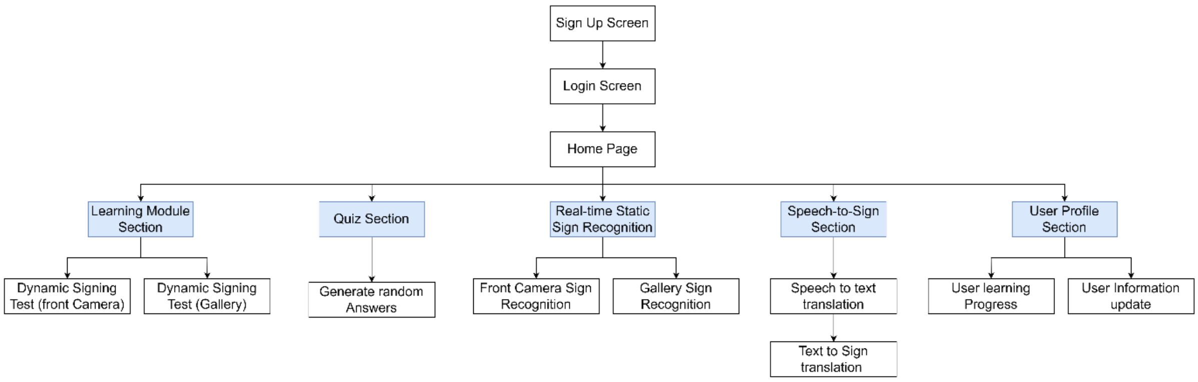


Figure 4. 2 System Block Diagram

4.4 System Component Specifications

<u>Sign Up Screen</u>

Purpose:

To allow new users to create an account in the application. Upon successful registration, the system will redirect user to sign in page.

Dependencies:

Depends on database system to store new user account information.

Depends on navigation component to move users to the sign in page after successful registration.

<u>Sign In Screen</u>

Purpose:

To allow registered users to log in to the application using their credentials. Upon successful log in, system will redirect users to Home Page.

Dependencies:

Depends on database system for user authentication.

Depends on navigation component to move users to the main page after successful login.

<u>Home Page</u>

Purpose:

To provide a welcoming interface for users upon log in and to facilitate the navigation to the main functions of the system, including learning modules, quizzes, user profile management, real-time recognition, and speech-to-sign translation.

Dependencies:

Depends on navigation component to handle the transition between the Home Page and other sections of the app.

<u>Learning Modules</u>

Purpose:

To allow users to access video demonstrations of hand signs in unlocked categories. Users can interact with the video player to control video playback.

Dependencies:

Depends on video player component and library to enable video playback and control functionalities.

Dynamic Signing Test

Purpose:

To allow users to assess their signing accuracy by comparing their hand key points with correct data, providing feedback through color-coded points (green for correct, red for incorrect) and display test results upon completion. Passed test will be redirected back to the Learning Module page.

Dependencies:

Depends on the functionality provided by the MediaPipe library for accurate hand key point detection.

Depends on the device's camera API to capture real-time video input.

Quiz Section

Purpose:

To allow users to assess their knowledge and understanding of sign language signs by presenting them with quiz questions related to unlocked categories. Users can choose their answers and receive immediate feedback on their selections. Upon selecting the correct answer, users can proceed to the next quiz question or unlock new categories based on their progress.

Dependencies:

Depends on video player component and library to enable video playback and control functionalities.

Real-time Static Sign Recognition

Purpose:

The purpose of the Real-time Static Recognition component is to enable users to recognize sign language signs either through live camera input or by selecting videos or images from the gallery. The system will use MediaPipe for hand key point extraction, feed the extracted key points into the recognition model, generate a predicted sign class, compare the user's key points with correct data, and provide feedback through color-coded points (green for correct, red for incorrect).

Dependencies:

Depends on the functionality provided by the MediaPipe library for accurate hand key point detection.

Depends on the recognition model to predict the sign class.

Depends on the device's camera API to capture real-time video input.

Speech-to-Sign Section

Purpose:

The purpose of the Speech to Sign Translator component is to enable users to convert spoken language into sign language for improved accessibility. It utilizes the Android SDK Speech Recognition API to convert speech to text and then displays the corresponding sign language video based on the recognized text.

Dependencies:

Depends on the functionality provided by the Android SDK Speech Recognition API for speech to text conversion.

<u>User Profile Section</u>

Purpose:

The purpose of the User profile component is to enable users to allow users to view and update their profile information within the application. Users may also view their learning progress in this page. It provides a user-friendly interface for users to manage their personal details easily.

Dependencies:

Depends on the SQLite database to store, retrieve, and update user information securely.

4.5 System Function

The detailed function implementation will be discussed in chapter 5.6 based on the system function below:

Table 4. 2 System Function Table

Function ID	Functions	Description
F001	Sign Up	To register new account.
F002	Login	To verify user authentication before accessing the system using username and password
F003	Edit User Profile	To update user personal information
F004	View Progress	To track user completed learning and quiz modules
F005	Learning Module	To view and display video demonstration of sign based on categories.
F006	Dynamic Signing Test (front camera)	To detect and compare user's real-time signing using device front camera, providing feedback by highlighting correct and incorrect points, as well as final results of user signing test.
F007	Dynamic Signing Test (gallery)	To detect and compare user's signing using videos selected from device gallery, providing feedback by highlighting correct and incorrect points, as well as final results of user signing test.
F008	Quiz Module	To test user understanding of Learning Module using interactive User Interface
F009	Speech to Sign Translation	To translate user speech into video of sign that correspond to the speech recognized
F010	Real time static recognition n feedback (front camera)	To detect, recognize and compare user's real-time signing using device front camera, providing feedback by highlighting correct and incorrect points, as well as final results of user signing test.
F011	Real time static recognition n feedback (Gallery)	To detect, recognize and compare user's signing using videos selected from device gallery, providing feedback by highlighting correct and incorrect points, as well as final results of user signing test.

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 Tools to Use

5.1.1 Hardware Specification

The hardware involved in this project is a computer and an android mobile device. The computer is used to develop the mobile application, which involves the machine learning and deep learning part for the functions in the application, especially the sign language detection and recognition function. As for the mobile device, it will be used for testing, to make sure that the user interface and functionalities defined are achieved in a real-world environment. Table 5.1 and 5.2 shows the specifications of the devices.

Table 5. 1 Specifications of computer

Description	Specifications
Model	Lenovo IdeaPad S340
Processor	Intel(R) Core (TM) i3-1005G1
Operating System	Windows 11
Graphic	Intel(R) UHD Graphics
Memory	8GB DDR4 RAM
Storage	256 SSD + 1TB HDD

Table 5. 2 Specifications of Mobile device

Description	Specifications
Model	Galaxy S23 Ultra
Processor	Octa-core (1x3.36 GHz Cortex-X3 & 2x2.8 GHz Cortex-A715 & 2x2.8 GHz Cortex-A710 & 3x2.0 GHz Cortex-A510)
Operating System	Android 13, One UI 5.1
Graphic	Adreno 740
Camera	Front Camera: 12 MP
RAM	12GB
Storage	512 GB

5.1.2 Software Specifications

The following software is needed to complete this project:

I. Integrated Development Environment (IDE) – Android studio

Android Studio will be the primary IDE for this project due to its powerful and integrated development features optimized specifically for building Android apps. It handles all aspects of the development cycle from the initial project setup to publishing releases. Developers can easily configure new projects with basic app scaffolding through Android Studio's unified UI. The IDE's built-in emulator simulator tests apps across different devices and resolutions without needing real hardware. Android Studio provides a fully featured environment that simplifies and enhances every aspect of creating high-quality Android apps for this project.

II. MediaPipe

Mediapipe is a framework built by Google for building multimodal applications. It provides solutions for hand and body tracking using machine learning. Some useful Mediapipe solutions in this project are the hands solution for detecting and tracking hands and fingertips in real-time.

III. Programming Languages

The primary programming language used will be Java due to its long-standing use in Android development, robust features and cross-platform capabilities. Java allows building complex logic, algorithms and integrating machine learning models. It will power the core app functions, backend server development and data processing tasks. Besides, Kotlin will also be used for the real-time recognition part due to its seamless integration with the existing Java code.

IV. Database Management System

For the database, SQlite will be used to store user information. SQlite is a lightweight and embedded database system that provides simplicity and efficiency for mobile application development.

V. Android SDK Speech Recognition API

Android devices come with a built-in speech recognition engine that performs the actual conversion of spoken words into text. This engine uses language models and algorithms to process audio input and generate text output. Some devices and Android versions support offline speech recognition, allowing recognition to occur locally on the device without an internet connection.

VI. Jupyter Notebook

Jupyter Notebook is a tool widely used for developing, training, and documenting machine learning models like sign language recognition models. Jupyter Notebook act as an interactive computing environment for training the sign language recognition model and then exporting it as a TensorFlow Lite (TFLite) model.

5.2 Hardware Setup

Before running the android project in actual device, some setup should be done to turn on developer option in the device.

Firstly, in the device's "Settings", select the "About Phone". Then, select "Software Information" as shown in Figure 5.1.

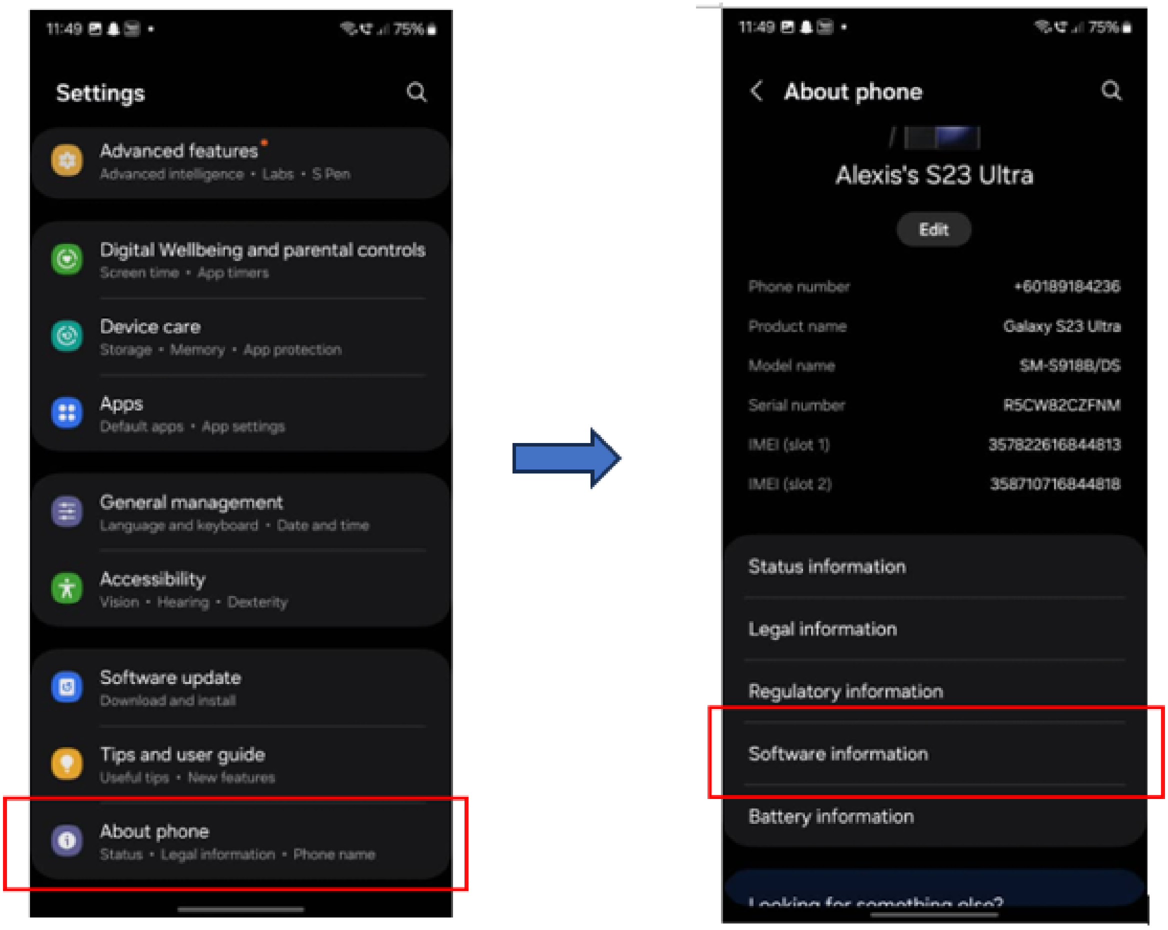


Figure 5. 1 Device Setup (1)

Next, tap on the “Build Number” for 7 times to turn on developer options (Figure 5.2)

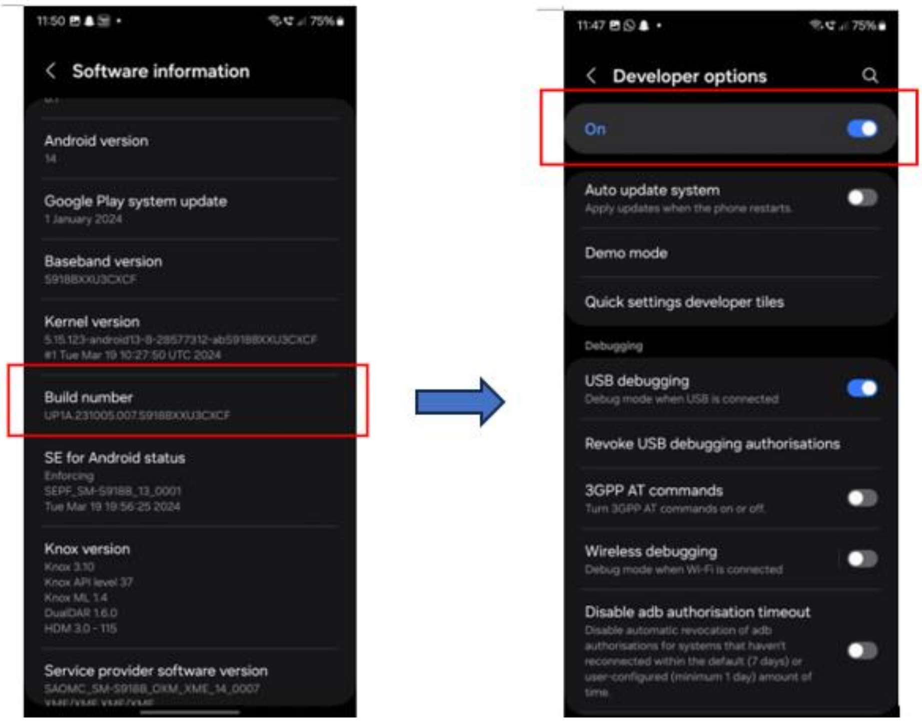


Figure 5. 2 Device Setup (2)

After turning on the “Developer Options”, turn on the USB debugging as shown in Figure 5.3, then connect the PC to the device using USB cable.

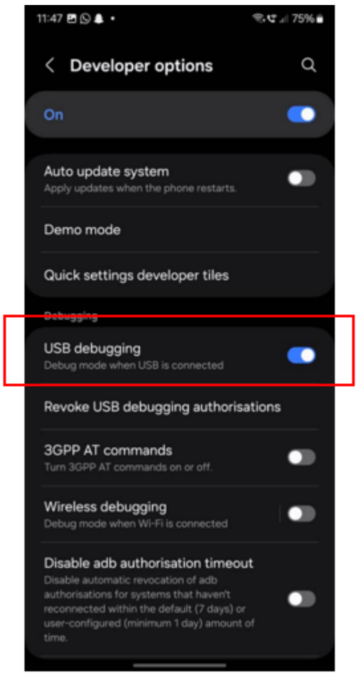


Figure 5. 3 Device Setup (3)

5.3 Software Setup

Before developing the sign language learning mobile application, there are two software needed to be installed and downloaded in my laptop:

1. Android Studio Giraffe 2022.3.1
2. Jupyter Notebook (to run the deep learning model and export it as tflite file)

Android studio Setup

Firstly, to start a new project, the "Empty View Activity" was selected as shown in Figure 5.4.

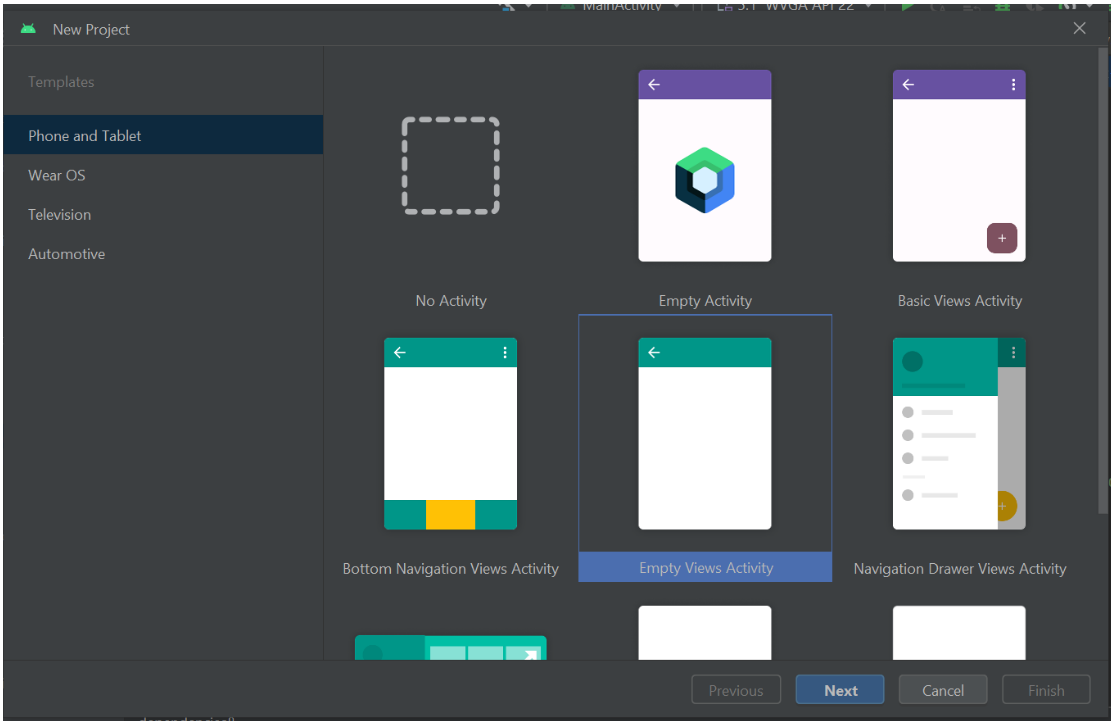


Figure 5. 4 Empty view activity

Then, the language is set to Java and the minimum SDK to run this application is API 24 ("Nougat"), android 7.0 (Figure 5.5).

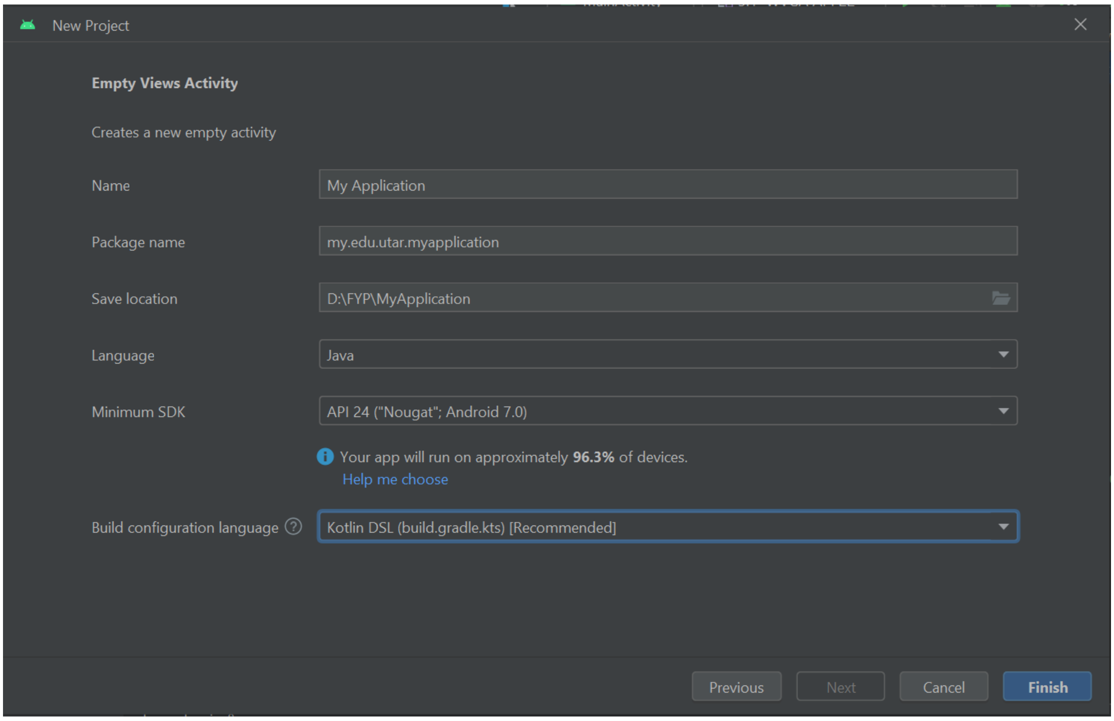


Figure 5. 5 Activity configurations

To run the project in a virtual android device, the hardware name 5.1'' WVGA was selected the system with API level 24 was chosen. (Figure 5.6 and Figure 5.7).

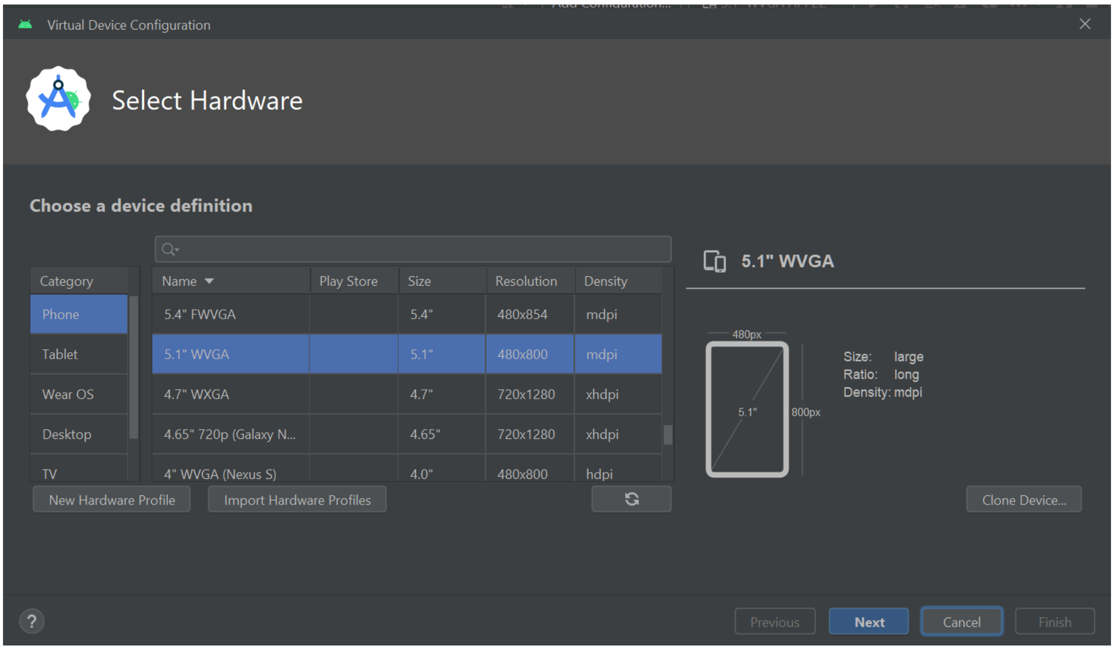


Figure 5. 6 Virtual Hardware configuration (1)

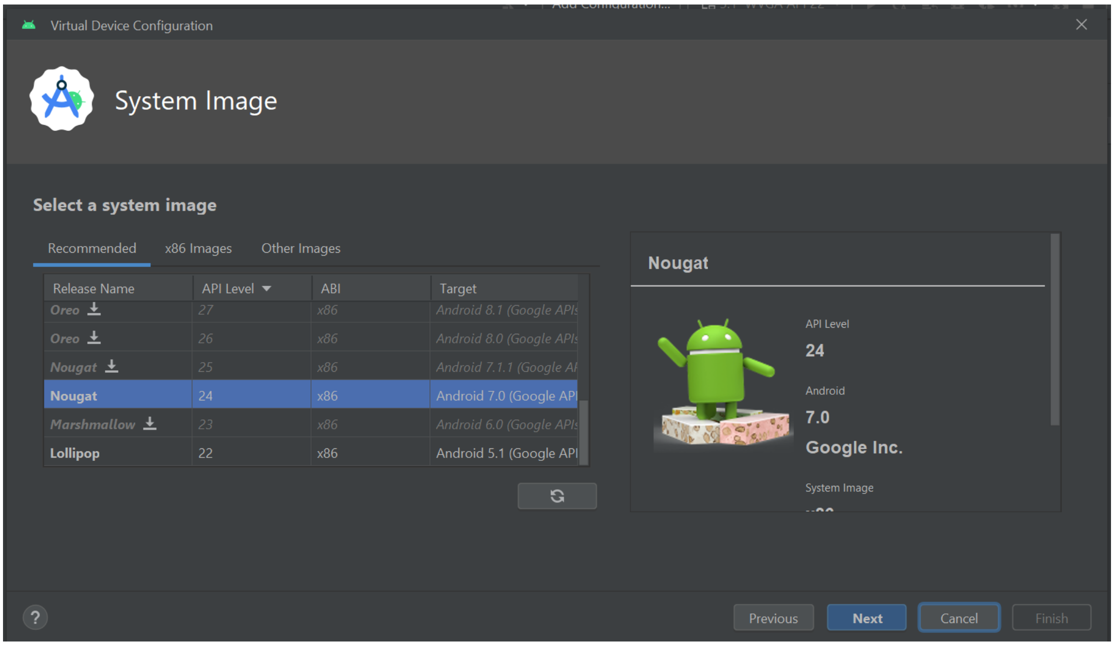


Figure 5. 7 Virtual Hardware configuration (2)

To run the project on an actual device, proceed to the physical section in the device manager, then turn on the phone settings, turn on developer mode and allow USB debugging from the computer as shown in Figure 5.8

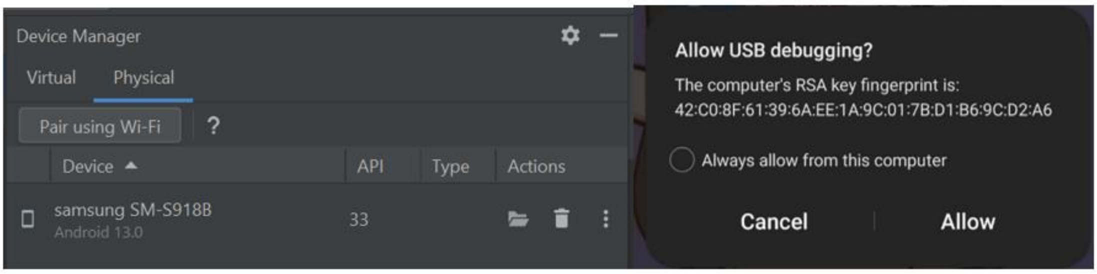


Figure 5. 8 Actual device configurations

5.4 System Requirements

5.4.1 Functional

Sign up:

1. Users should be able to register an account.

Log in:

1. Users should be able to login to the application using valid username and password.

Main Page:

1. Users should be able to view their username and level.
2. Users should be able to view bottom navigation bar.
3. Users should be able to enter Learning Module Page.
4. Users should be able to enter Quiz Page.
5. Users should be able to enter Real-time Recognition Page.
6. Users should be able to enter Speech-to-Sign Translation Page.

Learning Module Page (Video Demonstration):

1. Users should be able to view their unlocked categories.
2. Users should be able to select a category.
3. Users should be able to view video demonstrations of each sign in the chosen category.
4. Users should be able to pause the video.
5. Users should be able to replay the video.
6. Users should be able to rewind the video.
7. Users should be able to fast forward the video.

8. Users should be able to select and view next video.

9. Users should be able to unlock new category.

Learning Module Page (Real-time Signing Test):

1. Users should be able to view the front camera of their device.

2. Users should be able to choose to select video from Gallery.

3. Users should be able to view the key points of their hand highlighted in Red or Green.

4. Users should be able view a sample of correct key points on the screen.

5. Users should be able to view their results at the end of the signing test.

6. Users should be able to retry the signing test.

7. Users should be able to proceed to the next sign.

Quiz Page:

1. Users should be able to view their unlocked categories.

2. Users should be able to select a category.

3. Users should be able to view video demonstrations of each sign in the chosen category.

4. Users should be able to pause the video.

5. Users should be able to replay the video.

6. Users should be able to rewind the video.

7. Users should be able to fast forward the video.

8. Users should be able to view 4 random options under the video demonstration.

9. Users should be able to choose their answer.

10. Users should be able to see the answer in Red or Green.

11. Users should be able to select and view next quiz.

12. Users should be able to unlock new category.

Real-time Recognition Page (Static Sign Recognition):

1. Users should be able to view the front camera of their device.

2. Users should be able to choose to select video or image from Gallery.

3. Users should be able to view the key points of their hand highlighted in Red or Green.

4. Users should be able view a sample of correct key points on the screen.

5. Users should be able to view the predicted class of their sign.

6. Users should be able to view the inference time of each frame.

7. Users should be able to select the number of hands to be detected.

Speech-to-Sign Translation Page:

1. Users should be able to select the speak button to start speaking.

2. Users should be able to view the recognized speech in text.

3. Users should be able to view the video that corresponds to the text.

4. Users should be able to pause the video.

5. Users should be able to replay the video.

6. Users should be able to rewind the video.

7. Users should be able to fast forward the video.

User Profile Page:

1. Users should be able to view their pre-existing information.

2. Users should be able to update their latest information.

3. Users should be able to view their learning progress.

5.4.1 Non-Functional

Performance:

1. System should be able to respond to user actions within acceptable time limits.

2. System should be able to predict the class of the detected hand sign accurately and effectively.

Reliability:

1. System should be able to provide effective error handling to handle unexpected situations.

2. System should be able to provide meaningful error messages to users.

3. System should be able to run smoothly.

Security:

1. System should be able to ensure that only authorized users can access the system.

Usability:

1. System should be able to ensure that the user interface is intuitive, easy to navigate and visually appealing.

Integration and Compatibility:

1. System should be able to integrate seamlessly with external APIs for functionality.

2. System should be compatible with various devices to ensure wide user reach.

5.5 System Operation

5.5.1 Sign-In and Sign-Up User Interface

Figure 5.9 shows the application splash screen when user starts the application. Then a sign in page will be displayed for user to enter their username and password. After pressing on the sign in button, the username and password will be verified and redirected to the main page. User may also choose to proceed to sign up page.

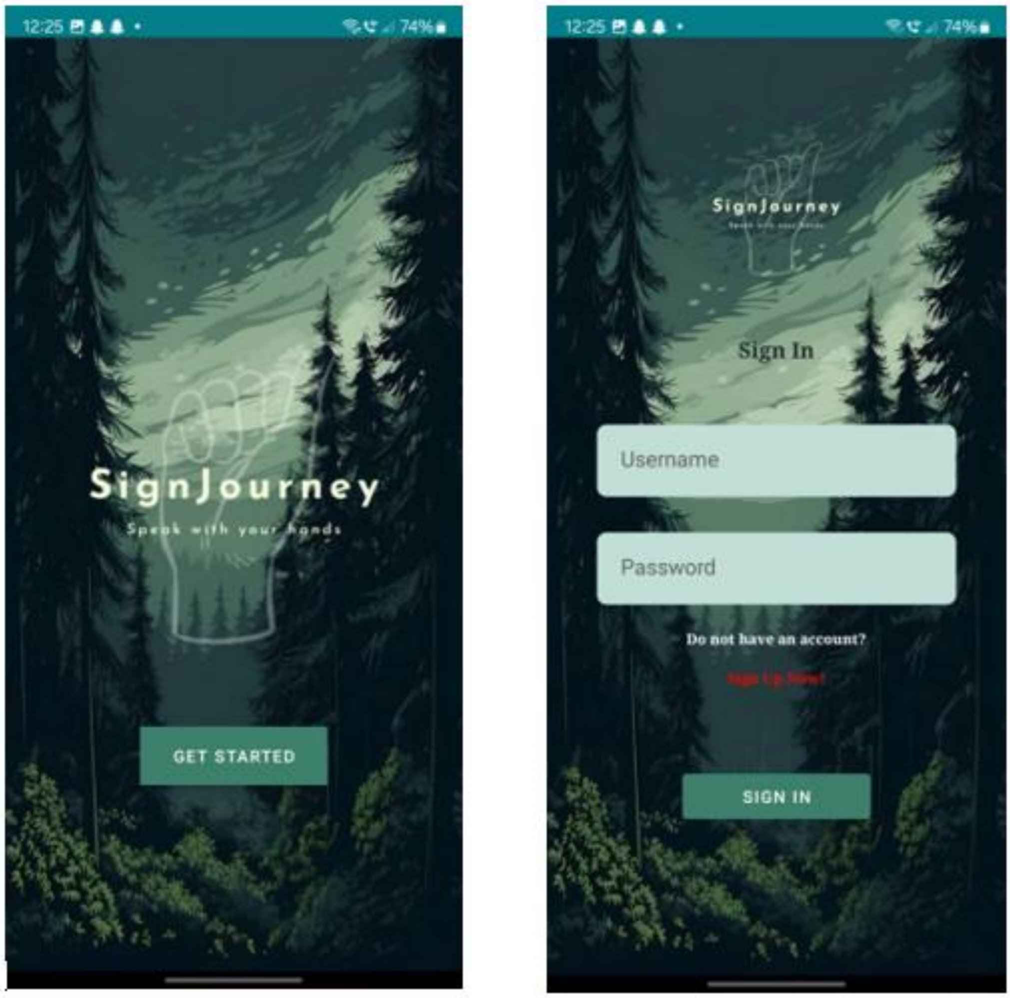


Figure 5. 9 Splash Screen and Sign in Screen

In the sign-up page as shown in Figure 5.10, user may enter their username and password for the new account. User should also choose their proficiency level, default level is set to “Beginner”.

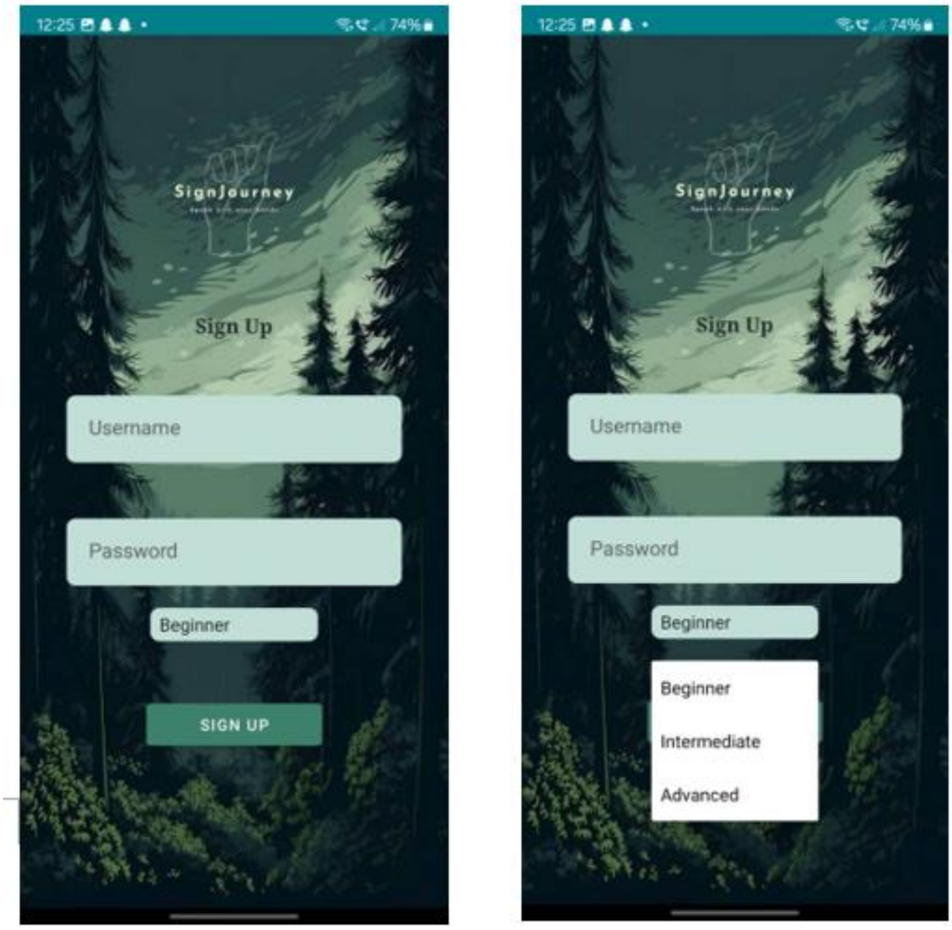


Figure 5. 10 Sign Up Page

5.5.2 Home Page User Interface

For the main page (Figure 5.11), there will be 5 main choices which is the learning modules, quizzes, real-time recognition, user profile and speech-to-sign translator. User may also navigate to these pages using the bottom navigation bar.

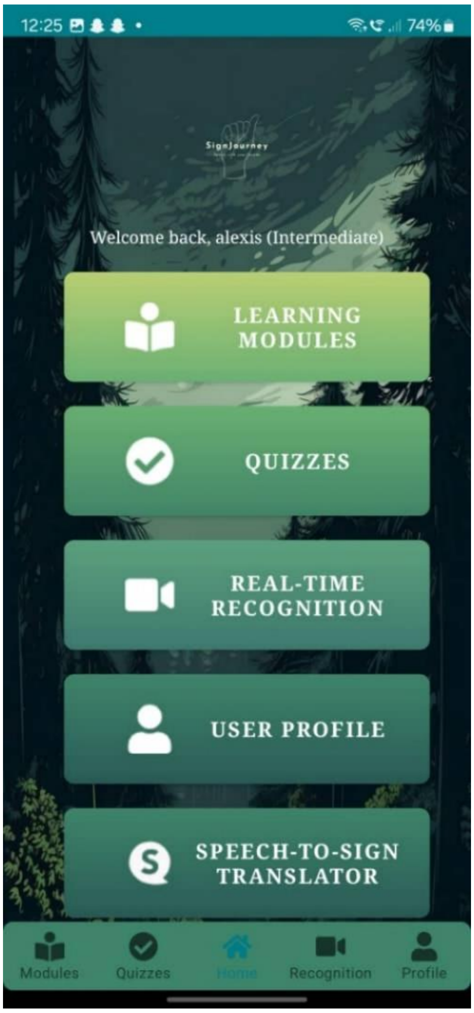


Figure 5. 11 Home Page

5.5.3 Learning Module (Basic Learning) User Interface

In the sign-up page, each user was initialized with a proficiency level, according to the level retrieved from database, user will have different number of modules unlocked initially as shown in Figure 5.12. Initially, Beginner level will have only the first module unlocked, while intermediate level will be able to unlock the first three modules, and Advanced level will have the first six modules unlock (Figure 5.12).

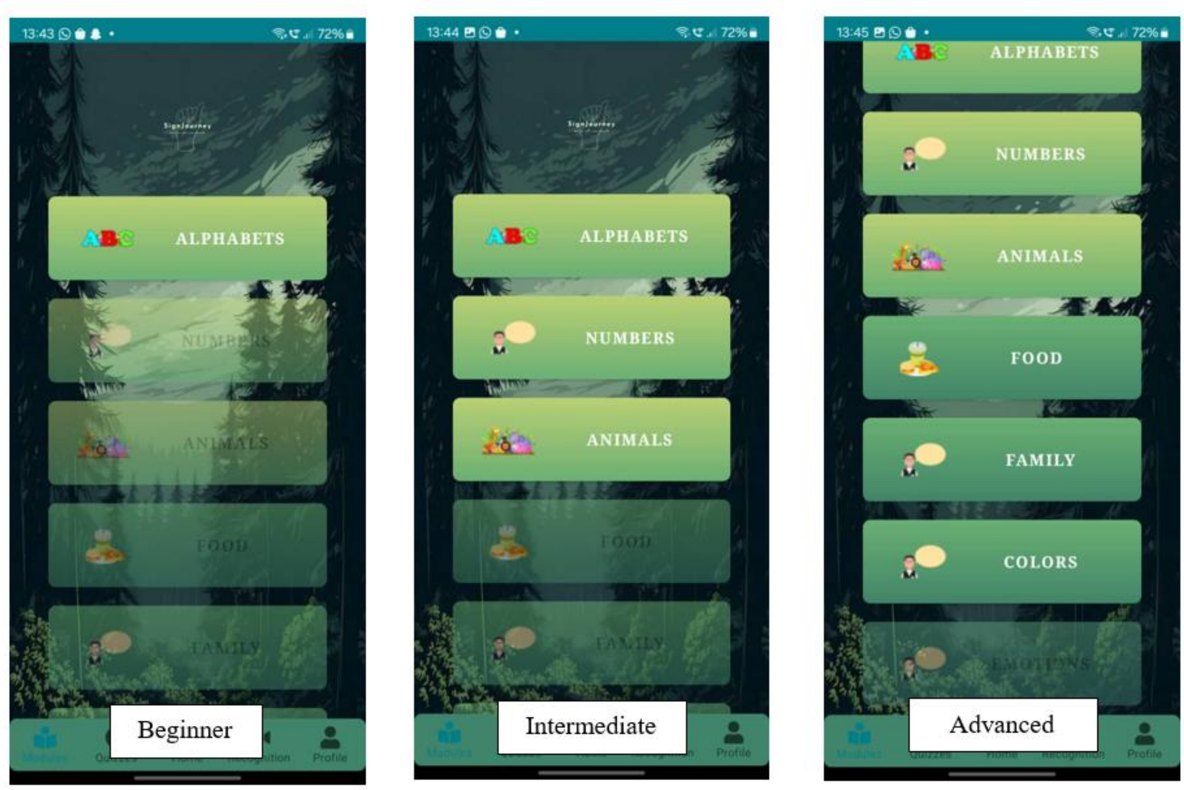


Figure 5. 12 Initial Learning Module Page

When users select a module, a demonstration video will be shown, users are allowed to pause, replay, fast forward or rewind the video. If users wish to proceed to the next sign, they may select the next button. When users reach the end of the module, End button will be displayed.

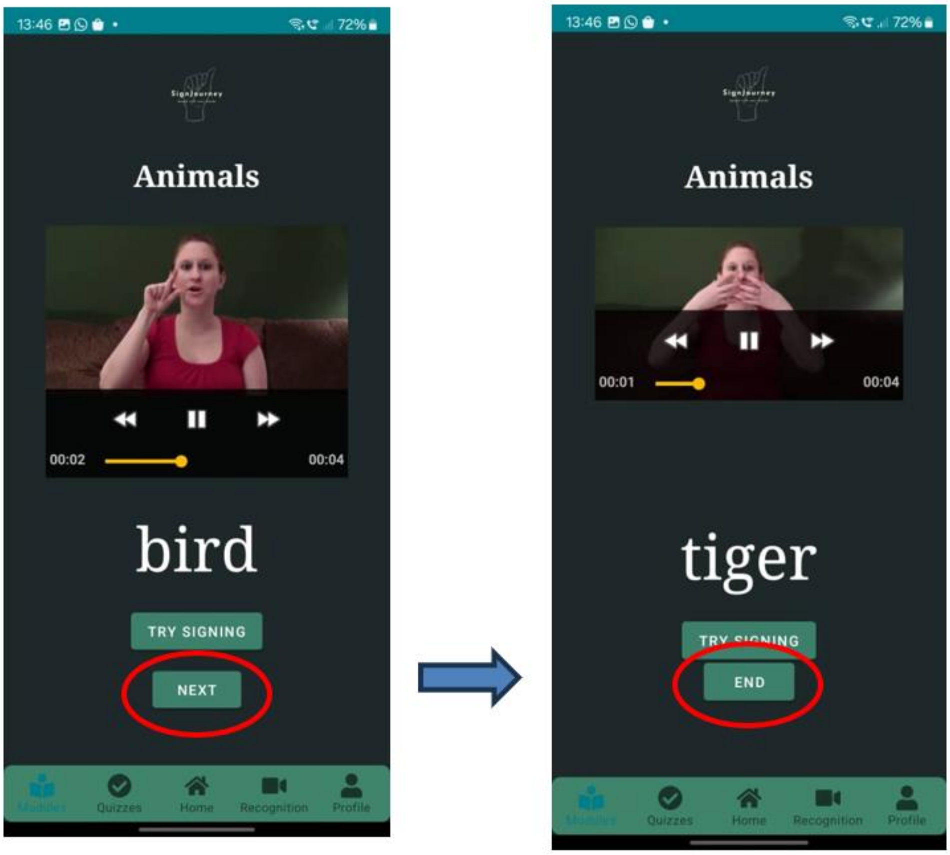


Figure 5. 13 Video Demonstration

When users complete the last unlocked module, a new module will be unlocked (Figure 5.14). The system will update the database with the latest progress of users.

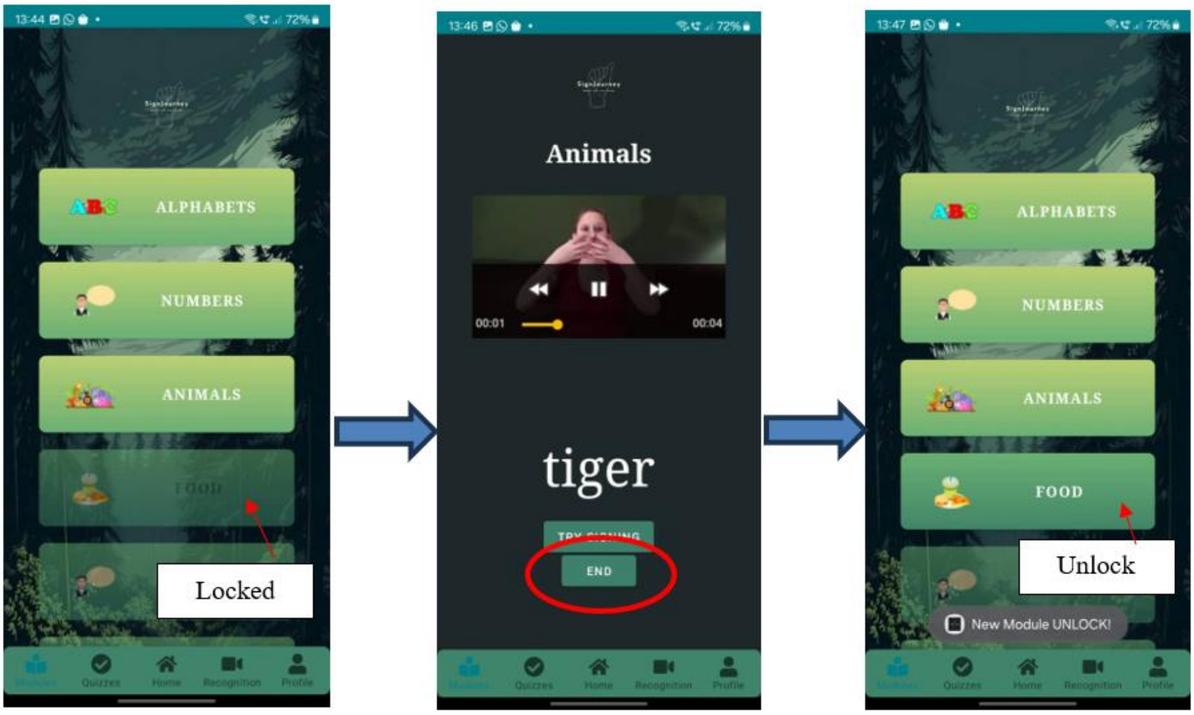


Figure 5. 14 Unlock Learning Module

5.5.4 Learning Module (Hand Sign Detection) User Interface

In the Learning Module, a hand sign detection and feedback function can be accessed by selecting the “Try Signing” button. When “Try Signing” button was selected, the system will turn on the front camera of the device as shown in Figure 5.15.

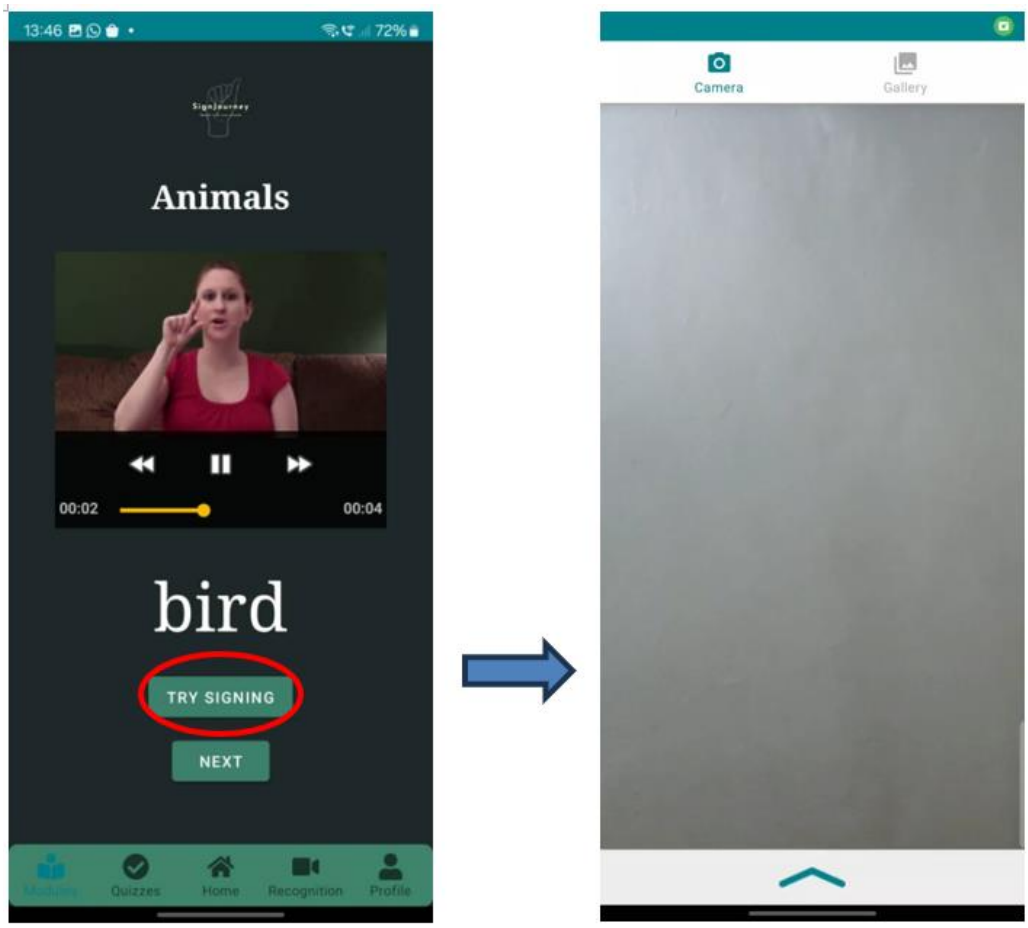


Figure 5. 15 Hand Sign Detection and Feedback

When hand was detected, the system will start to retrieve each key points of the hand detected in each frame using Mediapipe. The generated key points will then be used to compare with the corresponding data to calculate the accuracy. Key points highlighted in green color are correct, while those highlighted in red are incorrect. After completing the sign, the result will be calculated using the result of all frames (dynamic sign requires a series of movement, total number of frames may vary). A threshold of 50.00 was set, if users achieved a result greater than the threshold, "Next" button will be displayed to proceed to the next sign. Users may also choose to retry the same sign. Figure 5.16 shows an example of a passing test.

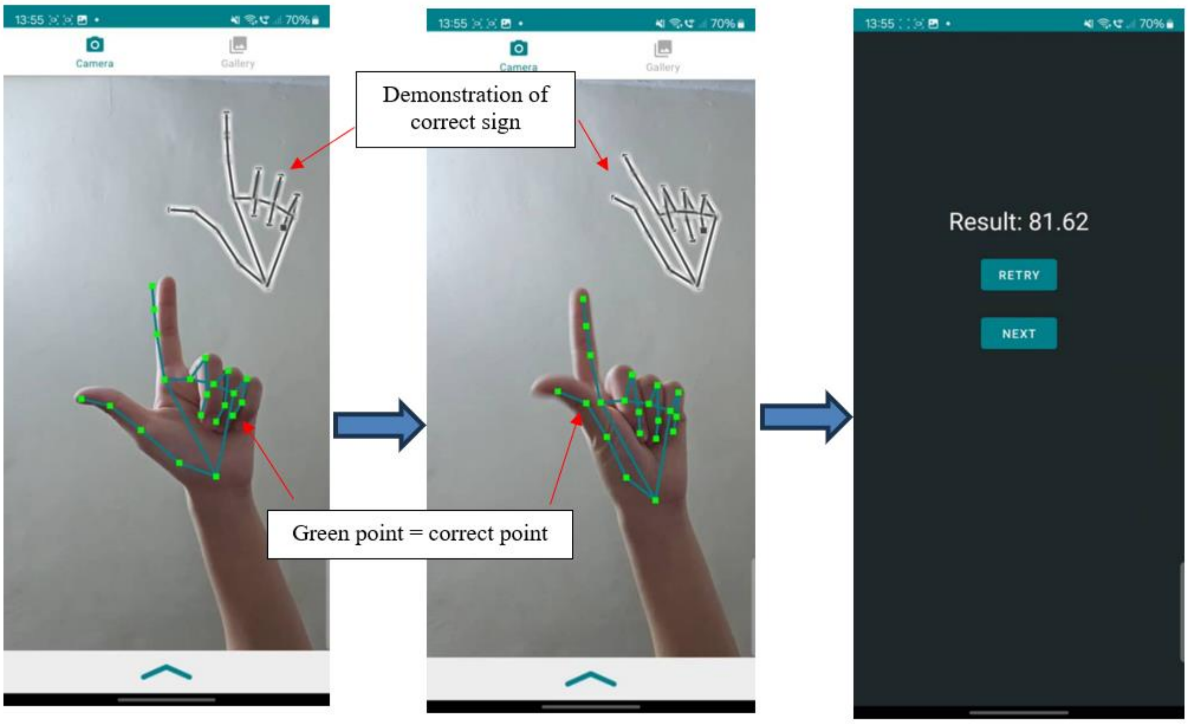


Figure 5. 16 Dynamic Sign Detection and Comparison (1)

When users fail to achieve 50 marks, only retry button will be displayed, users must retry the same sign until they pass the test. Figure 5.17 shows a fail test, red points represent the incorrect points.

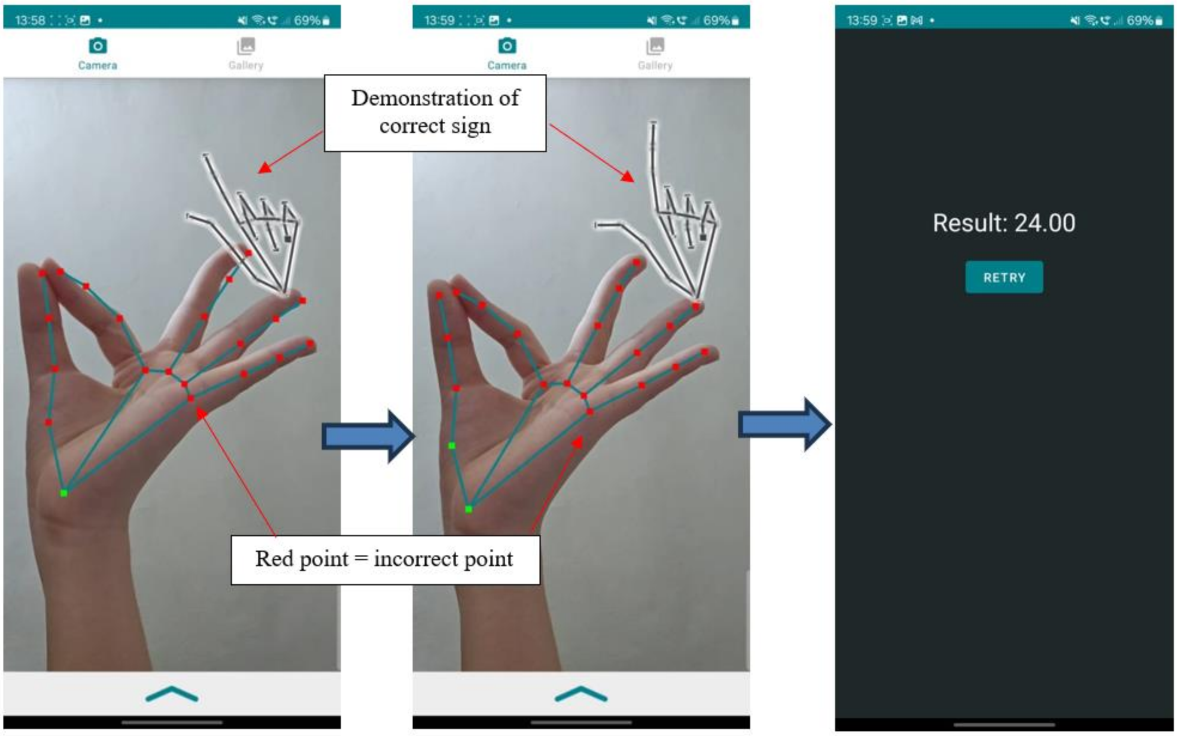


Figure 5. 17 Dynamic Sign Detection and Comparison (2)

Besides using front camera to detect the sign, users may also choose the "Gallery" option on top (Figure 5.18). The system will then access the gallery, and users may choose to upload a file for detection and test.

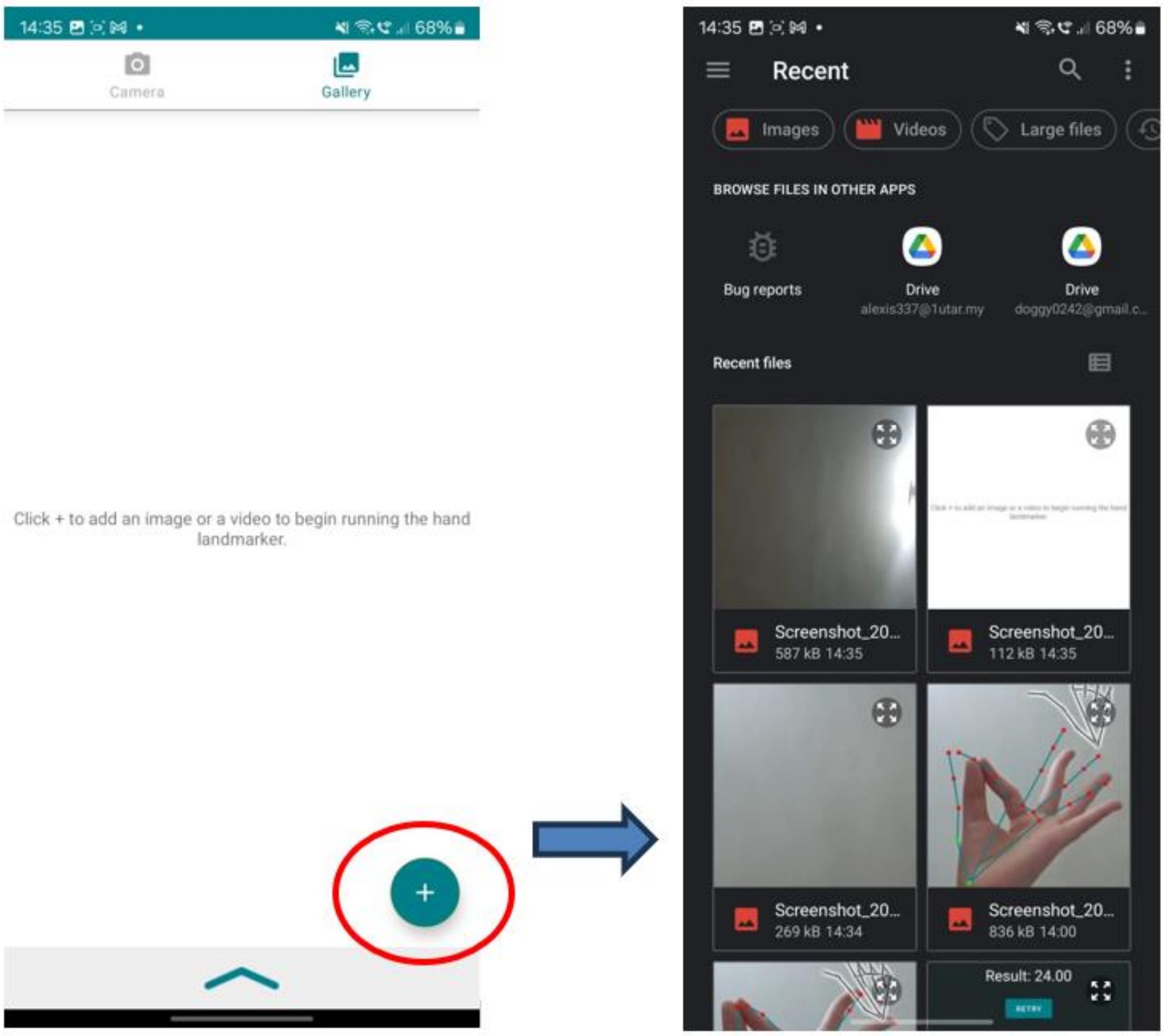


Figure 5. 18 Detection of sign from Gallery

5.5.5 Real time Sign Recognition and Feedback User Interface

In this module, static sign such as alphabets will be recognized using deep learning model. The system will receive each frame, when hand was detected in the frame, key points of the hands will be extracted using Mediapipe, following by preprocessing of these data before acting as an input of the recognition model. The predicted class from the recognition model will be displayed, along with the predicted left or right hand. Then, based on the predicted class, a comparison will be made between user's sign and the corresponding data. A threshold is set, if the point differs from the correct data, the point will be highlighted in Red, else, correct point will be highlighted in Green. Besides, a demonstration of the correct sign will also be displayed as a reference for users (Figure 5.19).

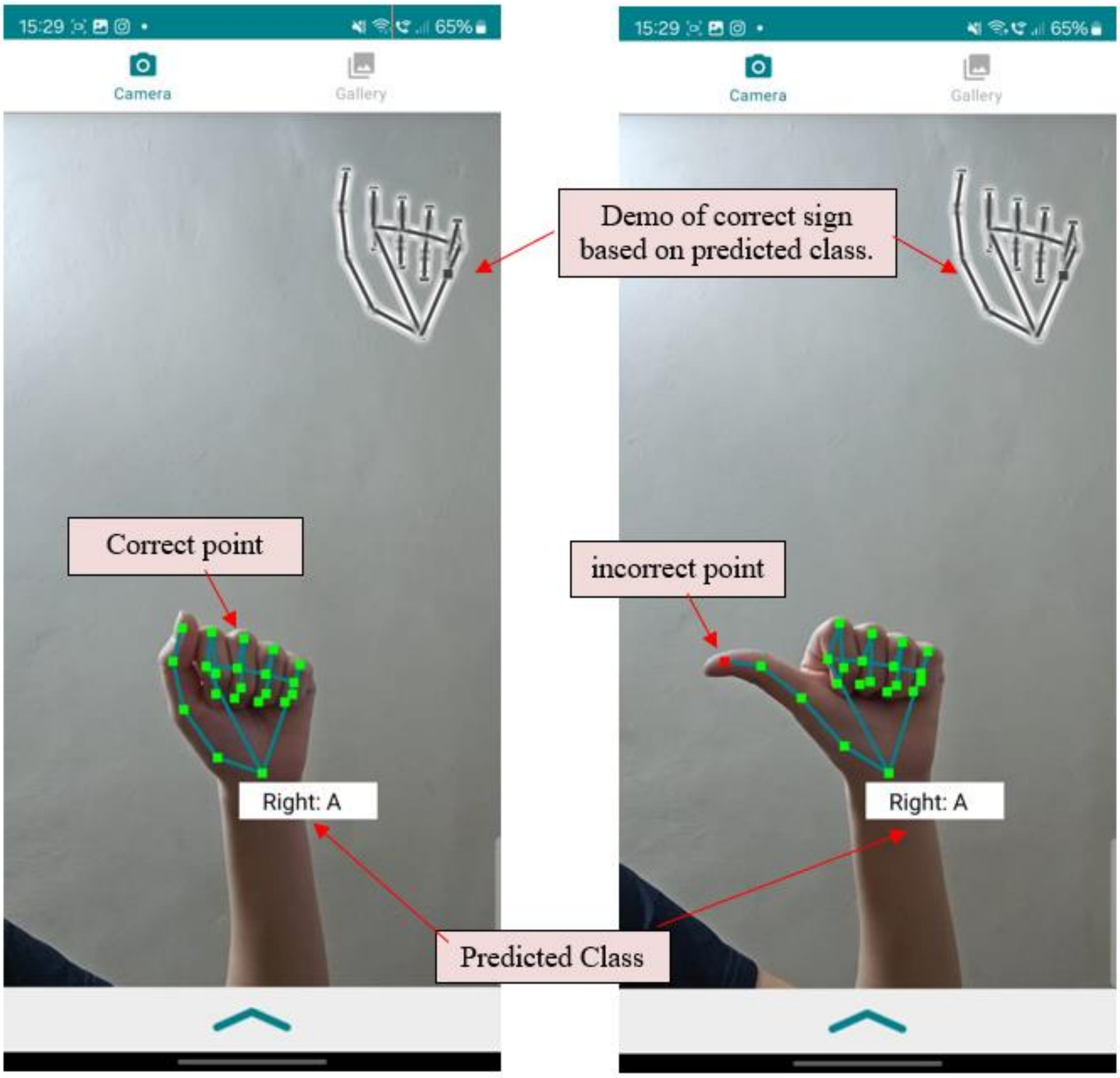


Figure 5. 19 Sign Recognition and Feedback

The recognition module allows users to recognise a maximum of 2 hands in each frame. As shown in Figure 5.20, both hands with different predicted class will be displayed, along with the left or right hand). If users wish to recognise only one hand, they may select the minus button. Figure 5.20 shows the scenario when users select single hand recognition, only the first hand detected in the frame will be recognised.

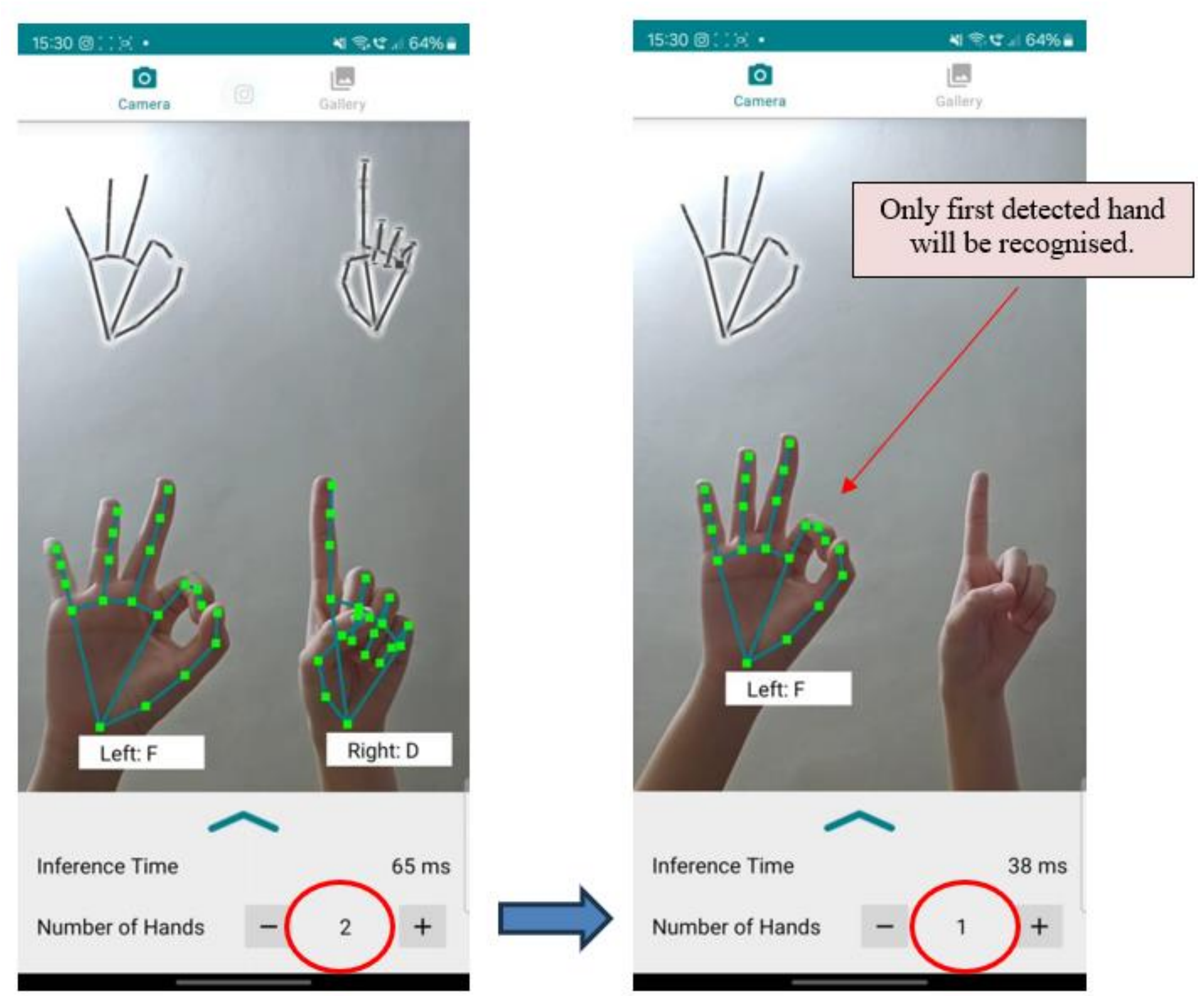


Figure 5. 20 Single hand recognition

User may also choose to recognise images or videos from Gallery as shown in Figure 5.21.

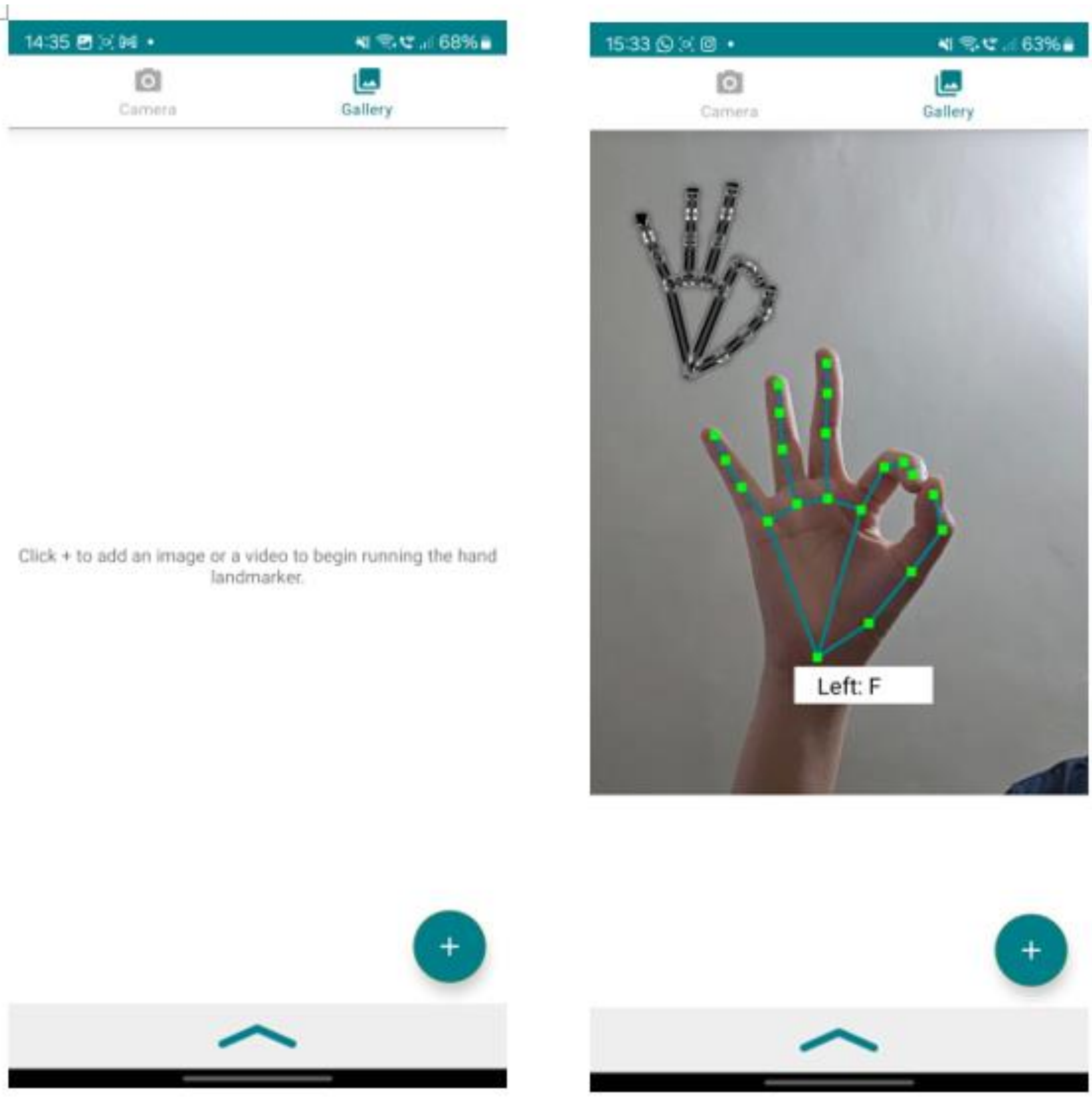


Figure 5. 21 Recognition from Gallery

5.5.6 Speech-to-Sign Language Translator Interface

In the Speech-to-Sign translator page, users may select the "Speak" button to start the speech recognition. The speech will then be recognized and translate into text. Then, based on the text translated, the system will display the corresponding video demonstration of the sign, along with the translated text below the video. User can also replay, pause, fast forward or rewind the video.

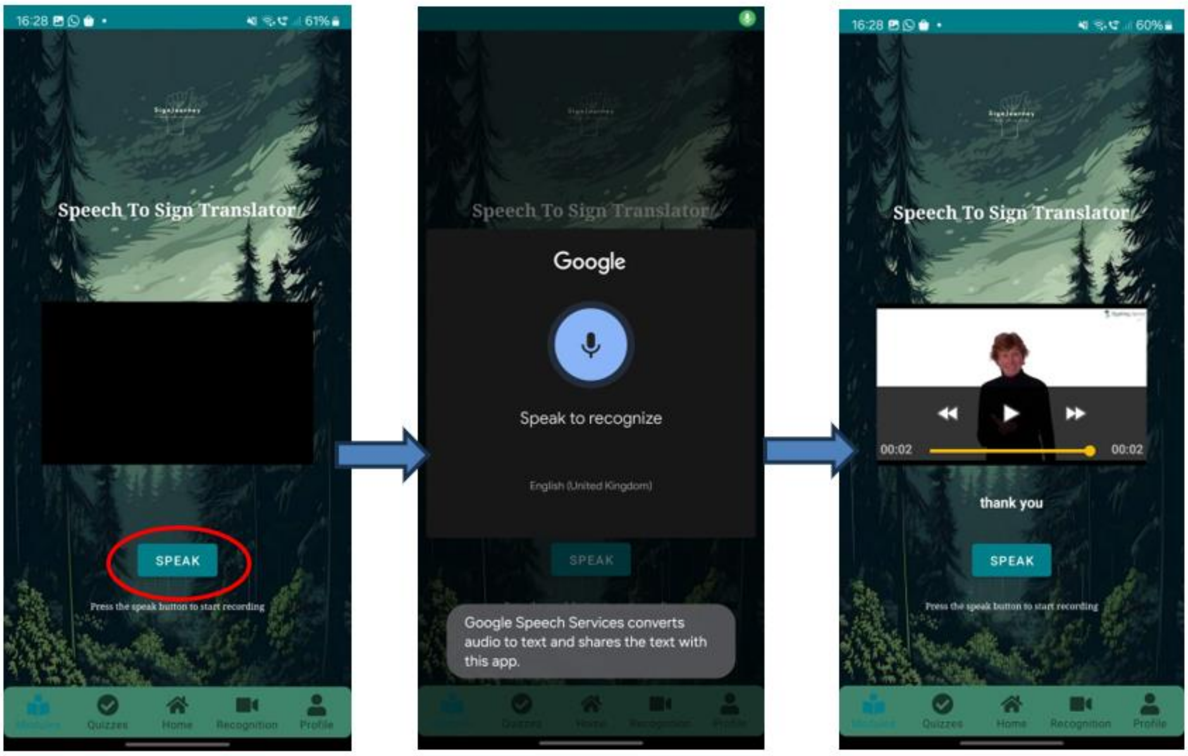


Figure 5. 22 Speech-to-Sign Translator

5.5.7 Interactive Quiz User Interface

Like Learning Module, the Interactive Quiz Module also unlocks different stages based on the proficiency level of users. When users select a quiz module, a demonstration video will be shown, along with 4 choices below, where one of them is the correct answer that corresponds to the video (Figure 5.23).

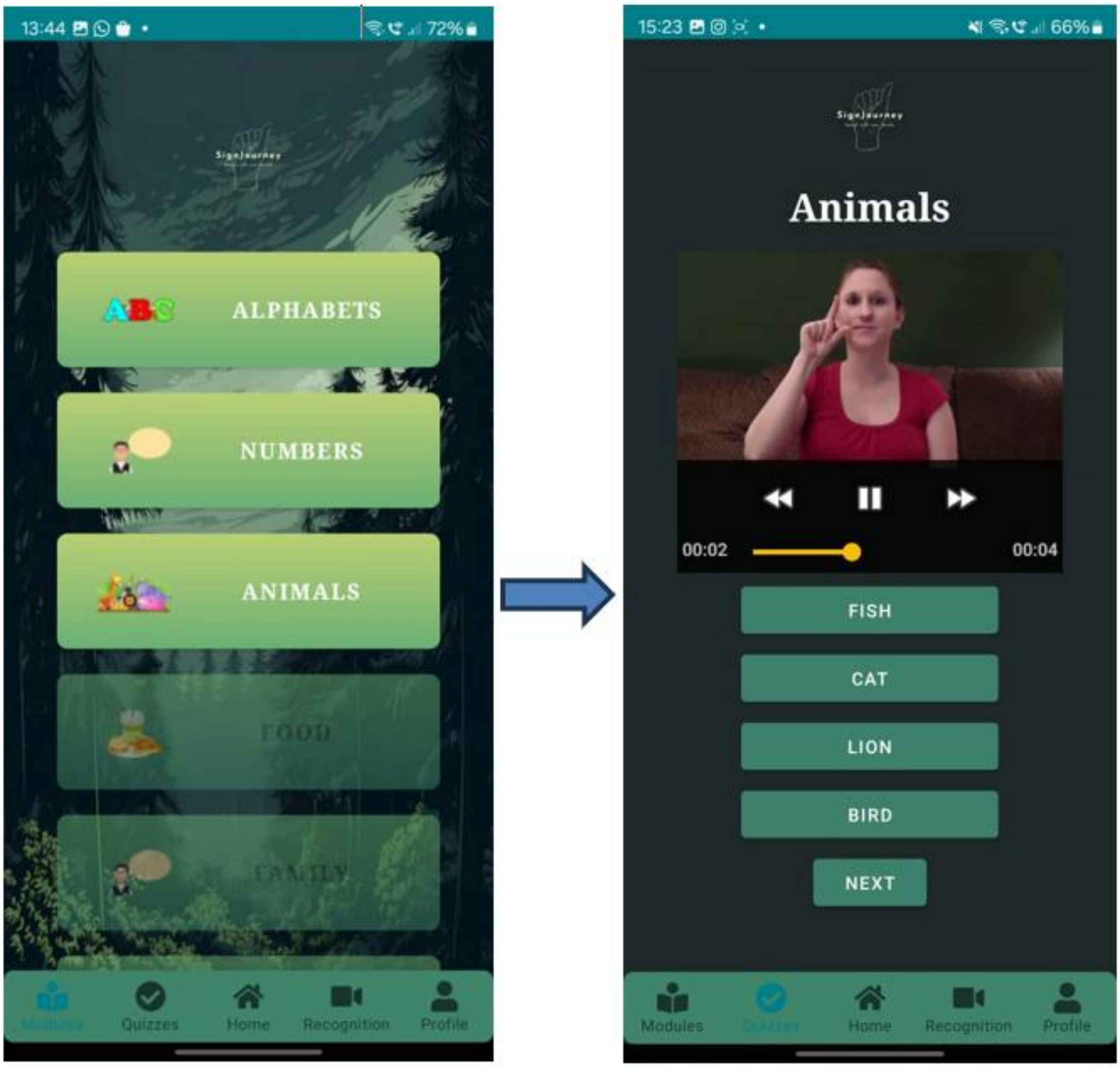


Figure 5. 23 Interactive Quiz User Interface (1)

After watching the video, users may choose one of the choices given, if the selected answer is correct, the choices will turn Green and a "Next" button will be displayed for users to proceed to the next question, else, the choices will turn Red, and user must retry until they select the correct answer as shown in Figure 5.24. When users complete the quiz, they will unlock a new quiz module, and their proficiency level will be updated in the database.

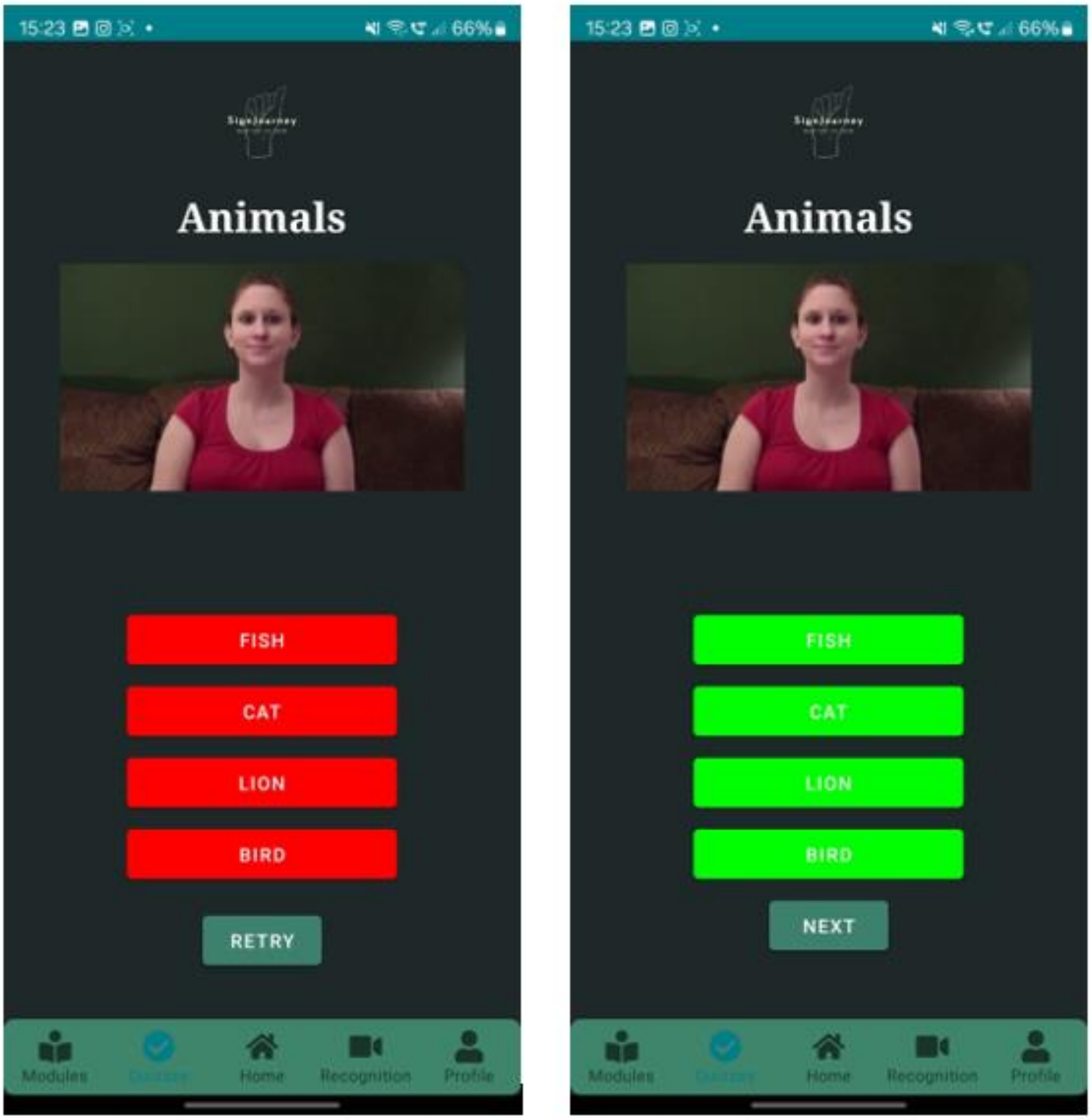


Figure 5. 24 Interactive Quiz User Interface (2)

5.5.8 Profile User Interface

The user profile records the basic information of users, in this page, users may enter their information. When “Update Profile” button was pressed, the data in the fields will be updated in the database, and the data will be retrieved from database and be displayed as shown in Figure 5.25.

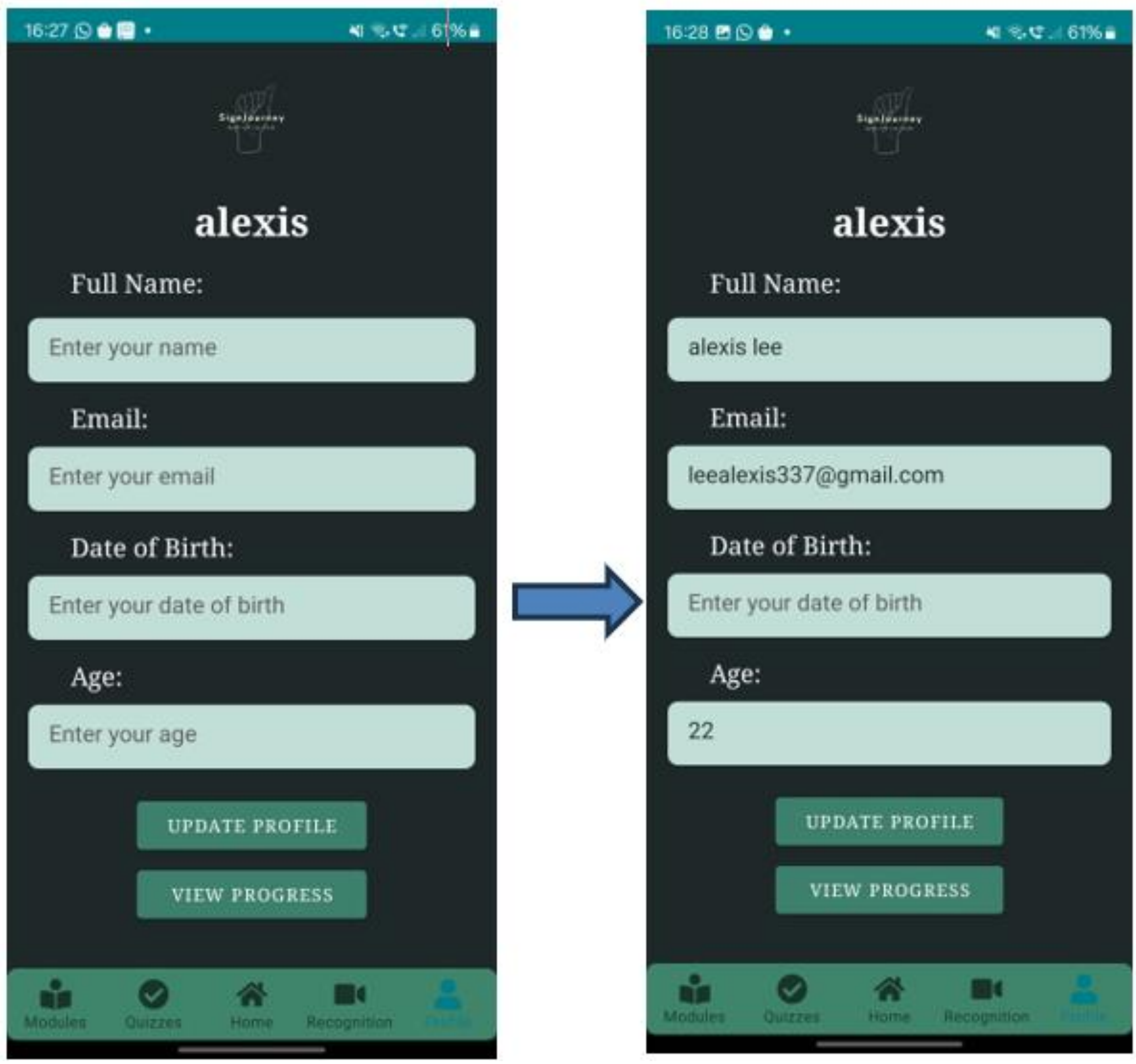


Figure 5. 25 User Profile Interface

In the Profile Page, a “View Progress” button was displayed for user to view their own learning Progress as shown in Figure 5.26. Users will see their number of completed Learning Modules and Quizzes, along with their achievements.

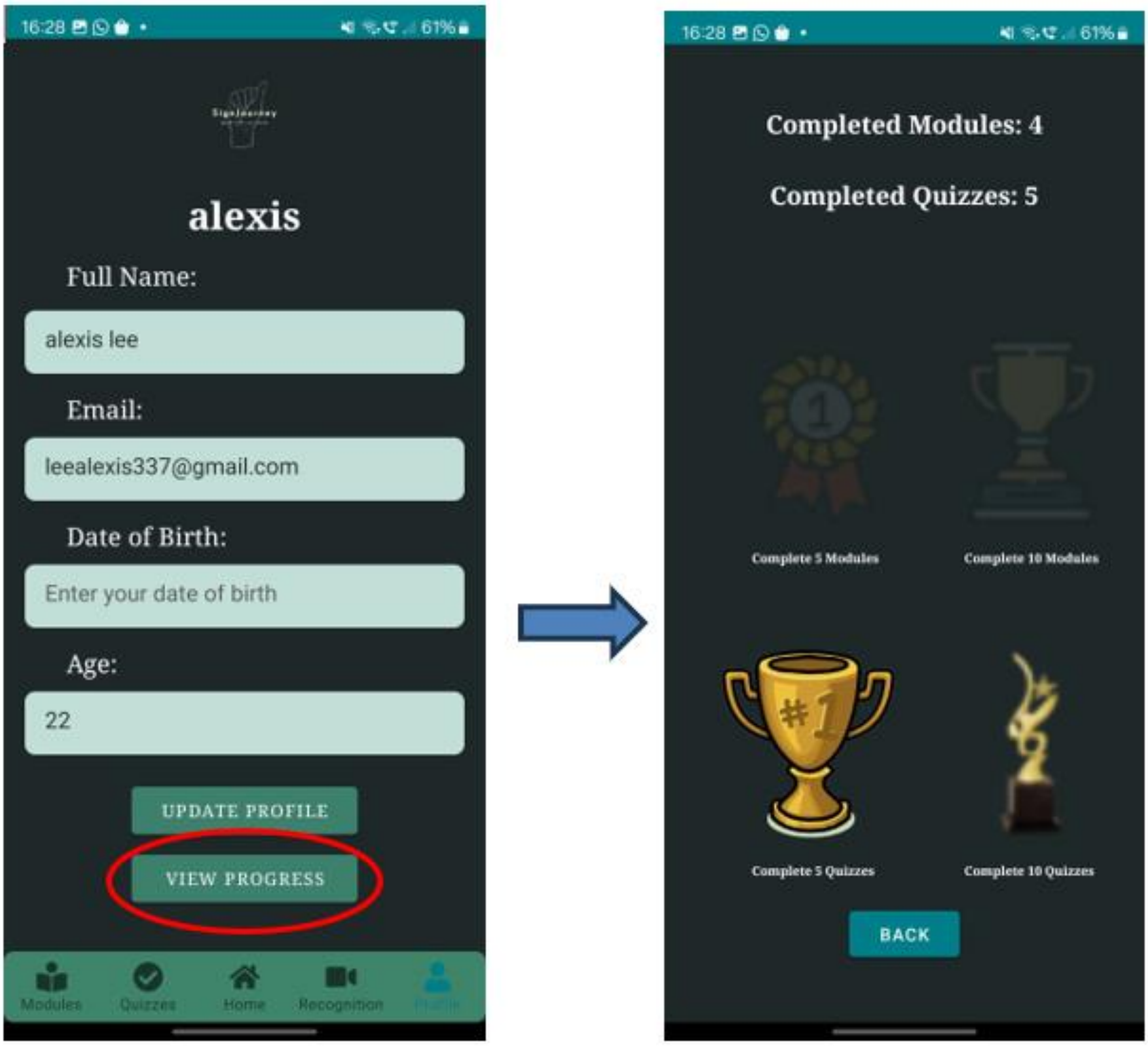


Figure 5. 26 View Progress Interface

5.6 Function Implementation Details

5.6.1 Sign in (F001)

The sign-up function plays a crucial role in user authentication and account management within the system.

Function overview:

The sign-up function is responsible for allowing new users to register accounts. Users will be required to enter information such as username, password and proficiency level. When user submits the sign-up form, a new account will be created, and the system will confirm successful registration with a confirmation message. After successful registration, users will be redirected to Sign in page.

Imports required for database interaction:

SQLiteDatabase is a class that can be used for reading and writing data. It provides methods like query(), insert(), update(), and delete() to perform database operations. SQLiteOpenHelper is a helper class used to manage database creation and version management. It includes methods like onCreate() for creating the database and onUpgrade() for upgrading the database schema.

Database Interaction:

Figure 5.27 shows the When users confirm the creation of a new account, the registration data will be processed. As shown in Figure 5.27, a new record will be created in the UserInfo table using the insert () method.

```
// Save data to database
SQLiteDatabase db = dbHelper.getWritableDatabase();
ContentValues values = new ContentValues();
values.put(DatabaseContract.UserEntry.COLUMN_USERNAME, username);
values.put(DatabaseContract.UserEntry.COLUMN_PASSWORD, password);
values.put(DatabaseContract.UserEntry.COLUMN_LEVEL, selectedLevel); // Save selected level

long newRowId = db.insert(DatabaseContract.UserEntry.TABLE_NAME, nullColumnHack: null, values);
```

Figure 5. 27 Database Interaction (Insert)

5.6.2 Sign in (F002)

The Sign in function is a critical part of the system's user authentication process, allowing users to access their accounts securely.

Function Overview:

The system requires users to enter their login credentials, which is the username and password. The entered credentials will then be verified against the stored user data to authenticate user's identity. Upon successful validation, the system will redirect users to the Home Page, along with the username and proficiency level (Figure 5.28). Both username and level will be displayed in the Home Page.

```
Intent intent = new Intent( packageContext: this, MainPage.class);
intent.putExtra( name: "USERNAME", username);
intent.putExtra( name: "LEVEL", level);
startActivity(intent);
```

Figure 5. 28 Intent example code

Database Interaction:

To verify user's identity, the system will retrieve user account information from the database as shown in Figure 5.29.

```
// Check credentials in the database
SQLiteDatabase db = dbHelper.getReadableDatabase();
String[] projection = {
        DatabaseContract.UserEntry.COLUMN_PASSWORD,
        DatabaseContract.UserEntry.COLUMN_LEVEL,
        DatabaseContract.UserEntry.COLUMN_COMPLETED_MODULES,
        DatabaseContract.UserEntry.COLUMN_COMPLETED_QUIZZES
};
String selection = DatabaseContract.UserEntry.COLUMN_USERNAME + " = ?";
```

Figure 5. 29 Database Interaction (Read)

5.6.3 Edit User Profile (F003)

Function Overview:

The system provides a user-friendly interface for users to input and update their profile details such as name, email, date or birth and age.

Database Interaction:

When users first enter the profile page, the system will retrieve the existing information from database as shown in Figure 5.30 using the query () method.

```
public Cursor getProfileData(String username) {
    SQLiteDatabase db = this.getReadableDatabase();

    String[] projection = {
            DatabaseContract.UserEntry.COLUMN_EMAIL,
            DatabaseContract.UserEntry.COLUMN_USERNAME,
            DatabaseContract.UserEntry.COLUMN_NAME,
            DatabaseContract.UserEntry.COLUMN_DOB,
            DatabaseContract.UserEntry.COLUMN_AGE
    };

    String selection = DatabaseContract.UserEntry.COLUMN_USERNAME + " = ?";
    String[] selectionArgs = {username};

    return db.query(
            DatabaseContract.UserEntry.TABLE_NAME,
            projection,
            selection,
            selectionArgs,
             groupBy: null,
             having: null,
             orderBy: null
    );
```

Figure 5. 30 Database Interaction (query)

After users confirm to update new information, the system will user the update () method to update data in the database.

```
public boolean updateProfile(String username, String name, String email, String dob, String age)
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

    values.put(DatabaseContract.UserEntry.COLUMN_NAME, name);
    values.put(DatabaseContract.UserEntry.COLUMN_EMAIL, email);
    values.put(DatabaseContract.UserEntry.COLUMN_DOB, dob);
    values.put(DatabaseContract.UserEntry.COLUMN_AGE, age);

    String selection = DatabaseContract.UserEntry.COLUMN_USERNAME + " = ?";
    String[] selectionArgs = {username};

    int rowsUpdated = db.update(
            DatabaseContract.UserEntry.TABLE_NAME,
            values,
            selection,
            selectionArgs);

    db.close();
```

Figure 5. 31 Database Interaction (update)

5.6.4 View Progress (F004)

Function Overview:

The system provides an interface to view their learning progress.

Database Interaction:

When user enters the view progress page, the system will retrieve user's completed modules and quizzes based on the username using the query () method. Then, based on the completed modules and quizzes, the opacity of the achievements is adjusted. The opacity of completed achievements will be set to fully visible.

5.6.4 Learning and Quiz Module (F005, F008)

Function Overview:

In the learning Module, users will choose their desired category to learn the sign based on video demonstration.

Method:

When user enters the learning module page, the system will retrieve the completed modules from the database using the query () method, the number of unlocked modules = completed modules + 1. User will then choose a category, based on the category, the system will retrieve videos that falls under the category. Before playing the videos, android.widget.MediaController needs to be imported, then MediaController will be set up as shown in Figure 5.32. The video will start automatically once it was loaded. MediaController allows users to play, pause, rewind or fast forward the video. For quiz module, the system will randomly generate 3 incorrect answers, along with one correct answer.

```
MediaController mediaController = new MediaController( context: this);
videoView.setMediaController(mediaController);
mediaController.setAnchorView(videoView);

videoView.setOnPreparedListener(mediaPlayer -> videoView.start());
```

Figure 5. 32 MediaController set up

When users completed all videos in the chosen category, if it is the first completion of the category, the system will update the completed modules or completed quiz in database using the update () method as shown in Figure 5.33.

```
// Define the update query
ContentValues values = new ContentValues();
values.put(DatabaseContract.UserEntry.COLUMN_COMPLETED_MODULES, 2); // Increment completed modules by 1

// Update the database
int rowsAffected = db.update(DatabaseContract.UserEntry.TABLE_NAME,
        values,
        whereClause: DatabaseContract.UserEntry.COLUMN_USERNAME + " = ?",
        new String[]{username});
```

Figure 5. 33 Database Interaction (update)

5.6.6 Dynamic Signing Test (F006, F007)

Function Overview:

The dynamic signing Test can be completed in 2 ways, which is through device's front camera, or by selecting videos from the device's gallery.

Libraries and Extensions:

Figure 5.34 shows the libraries and extensions needed for the system to access CameraX's features and utilities for camera-related task. Mediapipe library was also used to for hand landmark detection.

```
// CameraX core library
def camerax_version :String = '1.2.0-alpha02'
implementation "androidx.camera:camera-core:$camerax_version"

// CameraX Camera2 extensions
implementation "androidx.camera:camera-camera2:$camerax_version"

// CameraX Lifecycle library
implementation "androidx.camera:camera-lifecycle:$camerax_version"

// CameraX View class
implementation "androidx.camera:camera-view:$camerax_version"

  // MediaPipe Library
  implementation 'com.google.mediapipe:tasks-vision:0.10.0'
```

Figure 5. 34 CameraX and Mediapipe Libraries

Camera Permission:

To access the device's camera, permission should also be declared in the manifest file.

```
<uses-feature android:name="android.hardware.camera" /> <!-- Declare permissions -->
<uses-permission android:name="android.permission.CAMERA" />
```

Figure 5. 35 Camera Permission

Configurations:

Before using the mediapipe for hand tracking purposes, the configuration options was set as shown in Figure 5.36.

```kotlin
companion object {
    const val TAG = "HandLandmarkerHelper"
    private const val MP_HAND_LANDMARKER_TASK = "hand_landmarker.task"

    const val DELEGATE_CPU = 0
    const val DELEGATE_GPU = 1
    const val DEFAULT_HAND_DETECTION_CONFIDENCE = 0.5F
    const val DEFAULT_HAND_TRACKING_CONFIDENCE = 0.5F
    const val DEFAULT_HAND_PRESENCE_CONFIDENCE = 0.5F
    const val DEFAULT_NUM_HANDS = 1
    const val OTHER_ERROR = 0
    const val GPU_ERROR = 1
}
```

Figure 5. 36 Mediapipe configurations

Hand Key points Comparison and feedback:

With proper libraries, extension and permissions, the system will turn on the device's front camera when users start this function. From each frame received, mediapipe was utilized to extract the 21 key points from the detected hand. Then, the output of Mediapipe hand tracking model will be preprocess and normalized as shown in Figure 5.37, each key point will then be compared with the correct data. If the coordinate of the point exceeds or falls below the threshold, it will be considered as incorrect points, and will be displayed in Red. Else, correct points will be displayed in green.

```kotlin
fun preProcessLandmark(landmarkList: List<List<Float>>): List<Float> {
    val tempLandmarkList = ArrayList<ArrayList<Float>>()
    for (landmarkPoint in landmarkList) {
        tempLandmarkList.add(ArrayList(landmarkPoint))
    }

    // Convert to relative coordinates
    var baseX = 0f
    var baseY = 0f
    for ((index, landmarkPoint) in tempLandmarkList.withIndex()) {
        if (index == 0) {
            baseX = landmarkPoint[0]
            baseY = landmarkPoint[1]
        }

        tempLandmarkList[index][0] = landmarkPoint[0] - baseX
        tempLandmarkList[index][1] = landmarkPoint[1] - baseY
    }

    // Convert to a one-dimensional list
    val flattenedLandmarkList = ArrayList<Float>()
    for (landmarkPoint in tempLandmarkList) {
        flattenedLandmarkList.addAll(landmarkPoint)
    }

    // Normalization
    val maxValue = flattenedLandmarkList.maxOrNull() ?: 1.0

    fun normalize(n: Float): Float {
        return n / maxValue.toFloat()
    }

    return flattenedLandmarkList.map { normalize(it) }
```

Figure 5. 37 Key Point preprocess and normalization

5.6.7 Speech to Sign Translation (F009)

Function Overview:

The speech to sign translator allows user to translate spoken language into video of the sign recognized.

Permissions:

Figure 5.38 shows the permission required for the system to record audio using the device.

```
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

Figure 5. 38 Record Audio Permission

Method:

Using the Android Speech API, the system will translate user's speech into text, then from the text recognized, corresponding video will be retrieved and display to user using MediaController. Figure 5.39 shows the method used to translate speech to text. RecognizerIntent class was used. RecognizerIntent is a class in Android that provides constants and methods for working with speech recognition functionality.

```
private void startSpeechToText() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, value: "Speak to recognize");

    try {
        startActivityForResult(intent, REQUEST_CODE_SPEECH_INPUT);
    } catch (Exception e) {
        Toast.makeText( context: this, text: "Speech recognition not supported", Toast.LENGTH_SHORT).show();
    }
}
```

Figure 5. 39 RecognizerIntent

5.6.8 Real time static recognition and feedback (F010, F011)

Function Overview:

This function enables static hand sign to be recognized and compared to provide feedback.

Libraries required:

To run the Tensorflow Lite model in Android studio, the libraries shown in Figure 5.40 needs to be implemented to ensure the functionality and performance of the model.

```
implementation 'org.tensorflow:tensorflow-lite-support:0.1.0'
implementation 'org.tensorflow:tensorflow-lite-metadata:0.1.0'
implementation 'org.tensorflow:tensorflow-lite-gpu:2.3.0'
```

Figure 5. 40 Tensorflow Lite Libraries

Method:

A sequential neural network classifier model was built using TensorFlow's Keras API. First, each frame from the device's front camera will act as an input for the MediaPipe's hand tracking model. The output of the hand tracking model, which is a list of x and y coordinate will be preprocessed and normalized, this will then be the input for the recognition model. The output of the model is the class of the sign detected. Based on the predicted class, the system will retrieve the corresponding correct data of the sign, comparison will then be made to determine the final color of each point.

5.7 Concluding Remark

The system implementation is a series of procedures that are performed to develop the system according to the system design. Section 5.1 above shows the tools, setup and configurations that are implemented in the system. This chapter also shows the implemented User Interface along with the system operation. Lastly, chapter 5.6 also shows the detailed function implementation with relevant code snippet.

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

In this chapter, black box testing will be conducted to determine the performance of each module.

Firstly, for the sign up and sign, testing will be conducted to verify that users can successfully sign up with valid credentials and sign in using their registered credentials, both valid and invalid input will be tested. Next, the navigation from the main menu and bottom navigation bar to each main function will be tested to ensure seamless transitions can correct routing. Besides, the learning module will be verified through the video demonstration playback, real-time hand sign detection and comparison accuracy, verification will also be made to verify that learning modules are locked or unlocked based on user proficiency levels. As for the quiz module, testing will be conducted on the quiz question generation, random choice selection and locking or unlocking levels based on user proficiency. Furthermore, the real-time hand sign detection and recognition module will be testing through the accuracy using sample sign gestures, correct feedback will also be verified. User Profile will be verified based on user profile creation, update and viewing functions, user progress should also be accurately tracked and displayed. Lastly, testing will be conducted to verify the speech-to-text translation accuracy and corresponding video displayed.

Besides basic function testing, the performance metrics, timing, and efficiency of the deep learning model for static sign recognition will also be evaluated. The performance metrics used to evaluate the model's performance includes accuracy, precision, recall, F1 score and confusion matrix. The time taken by the model to process the input data and generate predictions will be measured in the inference test.

6.2 Testing Setup and Results

6.2.1 Sign In Function Testing

Table 6. 1 Sign in Function Testing

No.	Test Case	Test Data	Expected Output	Actual Output	Result (pass/fail)
1	Starts the application	-	System display splash screen, then requests username and password.	The system display splash screen, then requests username and password.	Pass
2.	Enters valid username and password	Username: alexis Password: 12345	System directs user to home page.	System direct user to home page.	Pass
3.	Enters non-existent username	Username: alex Password: 12345	System prompts "User not found".	System prompts "User not found".	Pass
4	Enters invalid password	Username: alexis Password: 123	System prompts "Incorrect Password".	System prompts "Incorrect Password".	Pass
5	Username not entered	Username: Password: 12345	System prompts "Please fill in all fields"	System prompts "Please fill in all fields"	Pass
6	Password not entered	Username: alexis Password:	System prompts "Please fill in all fields"	System prompts "Please fill in all fields"	Pass
7	Selects the "Sign Up Now" link	-	System directs user to Sign Up page	System directs user to Sign Up page	Pass

6.2.2 Sign up Function Testing

Table 6. 2 Sign up Function Testing

No.	Test Case	Test Data	Expected Output	Actual Output	Result (pass/fail)
1	Enters valid username, password and selects a proficiency level	Username: alex Password: 12345 Level: Intermediate	System directs user to Sign in Page, create new user record in database.	System directs user to Sign in Page, create new user record in database.	Pass
2	Enters existing username	Username: alexis Password: 12345 Level: Intermediate	System prompts “Username already exists”.	System prompts “Username already exists”.	Pass
3	Username not entered	Username: Password: 12345 Level: Intermediate	System prompts “Please fill in all fields”	System prompts “Please fill in all fields”	Pass
4	Password not entered	Username: alex Password: Level: Intermediate	System prompts “Please fill in all fields”	System prompts “Please fill in all fields”	Pass
5	Level not selected	Username: alex Password: 12345	System set default level to Beginner	System set default level to Beginner	Pass

6.2.3 Main Page Function Testing

Table 6. 3 Main Page Function Testing

No.	Test Case	Test Data	Expected Output	Actual Output	Result (pass/fail)
1	Selects Learning Modules button	-	System Directs user to the Learning Modules Page.	System Directs user to the Learning Modules Page.	Pass
2	Selects Quiz Modules button	-	System Directs user to the Quiz Modules Page.	System Directs user to the Quiz Modules Page.	Pass
3	Selects Real-Time Recognition button	-	System Directs user to the Recognition Page.	System Directs user to the Recognition Page.	Pass
4	Selects User Profile button	-	System Directs user to the User Profile Page.	System Directs user to the User Profile Page.	Pass
5	Selects Speech-to-Sign Translator button	-	System Directs user to the Speech-to-Sign Translator Page.	System Directs user to the Speech-to-Sign Translator Page.	Pass
6	Presses the application logo	-	System Directs user to Main Page.	System Directs user to Main Page.	Pass

6.2.4 Bottom Navigation Bar Function Testing

Table 6. 4 Bottom Navigation Bar Function Testing

No.	Test Case	Test Data	Expected Output	Actual Output	Result (pass/fail)
1	Selects Learning Modules button	-	System Directs user to the Learning Modules Page.	System Directs user to the Learning Modules Page.	Pass
2	Selects Quiz Modules button	-	System Directs user to the Quiz Modules Page.	System Directs user to the Quiz Modules Page.	Pass
3	Selects Real-Time Recognition button	-	System Directs user to the Recognition Page.	System Directs user to the Recognition Page.	Pass
4	Selects User Profile button	-	System Directs user to the User Profile Page.	System Directs user to the User Profile Page.	Pass

6.2.5 Learning Module Function Testing

Table 6. 5 Learning Module Function Testing

No.	Test Case	Test Data	Expected Output	Actual Output	Result (pass/fail)
1	Enters Learning Module page	Username: alexis Completed modules: 4	System retrieve number of completed modules of user from database, display 5 unlocked levels.	System retrieve number of completed modules of user from database, display 5 unlocked levels.	Pass
2	Selects a category	Category: Animals - bird	System displays corresponding videos with the name (bird) below.	System displays corresponding videos with the name below.	Pass
3	Selects pause video button	Category: Animals - bird	System pauses the bird sign video.	System pauses the bird sign video.	Pass
4	Selects rewind video button	Category: Animals - bird	System rewinds the bird sign video.	System rewinds the bird sign video.	Pass
5	Selects the fast forward button	Category: Animals - bird	System fast forwards the bird sign video.	System fast forwards the bird sign video.	Pass
6	Selects the play button after video finishes.	Category: Animals - bird	System replays the bird sign video.	System replays the bird sign video.	Pass

7	Selects the "next" button	Category: Animals Current sign: bird Next sign: cat	System refreshes the page with the next sign - cat	System refreshes the page with the next sign - cat	Pass
8	Selects the "End" button in the last sign of the category for the first time.	Category: Animals Sign: Tiger (last) Status: First time completion. Completed modules: 4 Next Category: Food	System directs user to the learning module page, update the completed module in database to 5, unlock the next category-Food	System directs user to the learning module page, update the completed module in database to 5, unlock the next category-Food	Pass
9	Selects the "End" button in the last sign of the category.	Category: Animals Sign: Tiger (last) Status: Second time completion. Next Category: Food	System directs user to the learning module page.	System directs user to the learning module page.	Pass
10	Selects the "Try Signing" button	Category: Animals Sign: bird	System turns on the front camera of user's device	System turns on the front camera of user's device	Pass
11	Starts showing sign using front camera	Category: Animals Sign: bird	System detects, compares and display hand key points and starts to calculate the	System detects, compares and display hand key points and starts to calculate the	Pass

			results of each frame.	results of each frame.	
12	Selects “Gallery”	Category: Animals Sign: bird	System accesses and displays the device’s gallery.	System accesses and displays the device’s gallery.	Pass
13	Selects a video from Gallery	Category: Animals Sign: bird Selected video: bird.mp4	System plays the selected video, detects, compares and display hand key points and starts to calculate the results of each frame.	System plays the selected video, detects, compares and display hand key points and starts to calculate the results of each frame.	Pass
14	Displays incorrect points	Category: Animals Sign: bird Key points: incorrect	System highlights incorrect key points in Red Color.	System highlights incorrect key points in Red Color.	Pass
15	Displays incorrect points	Category: Animals Sign: bird Key points: correct	System highlights incorrect key points in Green Color.	System highlights incorrect key points in Green Color.	Pass
16	Passes the test	Category: Animals Sign: bird Result: 80.00	System display results, along with a “retry” and “next” button.	System display results, along with a “retry” and “next” button.	Pass
17	Fails the test	Category: Animals Sign: bird Result: 30.00	System display results, along with a “retry” and hides the “next” button.	System display results, along with a “retry” and hides the “next” button.	Pass

18	Selects "retry" button	Category: Animals Sign: bird	System turns on the camera and start detecting and comparing the bird sign again.	System turns on the camera and start detecting and comparing the bird sign again.	Pass
19	Selects the "next" button	Category: Animals Sign: bird Next sign: cat	System turns off the camera, directs user to the next sign- cat.	System turns off the camera, directs user to the next sign- cat.	Pass

6.2.6 Quiz Module Function Testing

Table 6. 6 Quiz Module Function Testing

No.	Test Case	Test Data	Expected Output	Actual Output	Result (pass/fail)
1	Enters Quiz Module page	Username: alexis Completed quizzes: 4	System retrieve number of completed quizzes of user from database, display 5 unlocked levels.	System retrieve number of completed quizzes of user from database, display 5 unlocked levels.	Pass
2	Selects a category	Category: Animals - bird	System displays corresponding videos with 4 random choices below.	System displays corresponding videos with 4 random choices below.	Pass
3	Selects pause video button	Category: Animals - bird	System pauses the bird sign video.	System pauses the bird sign video.	Pass
4	Selects rewind video button	Category: Animals - bird	System rewinds the bird sign video.	System rewinds the bird sign video.	Pass
5	Selects the fast forward button	Category: Animals - bird	System fast forwards the bird sign video.	System fast forwards the bird sign video.	Pass
6	Selects the play button after video finishes.	Category: Animals - bird	System replays the bird sign video.	System replays the bird sign video.	Pass

7	Selects the incorrect answer	Category: Animals – bird Selected answer: fish	System display answer in red color, then display a “retry” button	System display answer in red color, then display a “retry” button	Pass
8	Selects the correct answer	Category: Animals – bird Selected answer: bird	System display answer in green color, then display a “next” button	System display answer in green color, then display a “next” button	Pass
9	Selects the next button	Category: Animals Current sign: bird Next sign: cat	System refreshes the page with the next sign - cat	System refreshes the page with the next sign - cat	Pass
10	Selects the “End” button in the last sign of the category for the first time.	Category: Animals Sign: Tiger (last) Status: First time completion. Completed quiz: 4 Next Category: Food	System directs user to the quiz module page, update the completed quiz in database to 5, unlock the next category-Food	System directs user to the quiz module page, update the completed quiz in database to 5, unlock the next category-Food	Pass
11	Selects the “End” button in the last sign of the category.	Category: Animals Sign: Tiger (last) Status: Second time completion. Next Category: Food	System directs user to the quiz module page.	System directs user to the quiz module page.	Pass

6.2.7 Profile Page Function Testing

Table 6. 7 Profile Page Function Testing

No.	Test Case	Test Data	Expected Output	Actual Output	Result (pass/fail)
1	Enters Profile page with no data saved previously	Username: alexis Full Name: Email: Date of Birth: Age:	System displays input fields for Full name, Email, Date of Birth and Age	System displays input fields for Full name, Email, Date of Birth and Age	Pass
2	Enters Profile page with previously saved information	Username: alexis Full Name: Alexis Lee Email: alexis@gmail.com Date of Birth: 21/9/2002 Age: 22	System retrieves and auto fills the existing data in each field.	System retrieves and auto fills the existing data in each field.	Pass
3	Enter new information	Age: 22 New Age: 25	System updates the record in database and display “Profile update successfully”.	System updates the record in database and display “Profile update successfully”.	Pass
4	Selects “View Progress”	Username: alexis Completed modules: 4 Completed quiz: 5	System displays user’s progress, along with the achievements below.	System displays user’s progress, along with the achievements below.	Pass

6.2.8 Real-time Recognition Function Testing

Table 6. 8 Real-time Recognition Function Testing

No.	Test Case	Test Data	Expected Output	Actual Output	Result (pass/fail)
1	Enters the recognition page	-	System turns on the front camera of the device.	System turns on the front camera of the device.	Pass
2	Starts showing sign using front camera (Single hand)	Sign shown: “A” Hand: Right Number of hands (Settings): 2	System detects hands, display the predicted class as A-Right, sample of correct sign and the key points of user’s hand.	System detects hands, display the predicted class as A-Right, sample of correct sign and the key points of user’s hand.	Pass
3	Starts showing sign using front camera (Both hand)	Sign shown (Right) : “A” Sign Shown (Left): “B” Number of hands (Settings): 2	System detects each hands, display the predicted class as A-Right and B-Left, sample of each correct sign and the key points of user’s hand.	System detects each hands, display the predicted class as A-Right and B-Left, sample of each correct sign and the key points of user’s hand.	Pass
4	Selects “Gallery”	-	System accesses and displays the device’s gallery.	System accesses and displays the device’s gallery.	Pass
5	Selects a video from Gallery	Sign Shown: “A” Selected video: a.mp4	System plays the selected video, detects hands, display the predicted class as	System plays the selected video, detects hands, display the predicted class as	Pass

			A-Right, sample of correct sign and the key points of user's hand.	A-Right, sample of correct sign and the key points of user's hand.	
6	Selects an image from Gallery	Sign Shown: "A" Selected video: a.png	System plays the selected image, detects hands, display the predicted class as A-Right, sample of correct sign and the key points of user's hand.	System plays the selected image, detects hands, display the predicted class as A-Right, sample of correct sign and the key points of user's hand.	Pass
7	Displays incorrect points	Sign: "A" Key points: incorrect	System highlights incorrect key points in Red Color.	System highlights incorrect key points in Red Color.	Pass
8	Displays incorrect points	Sign: "A" Key points: correct	System highlights incorrect key points in Green Color.	System highlights incorrect key points in Green Color.	Pass
9	Changes the number of hands to "1"	First sign shown (Right): "A" Second sign Shown (Left): "B" Number of hands (Settings): 1	System detects only the first hand shown, display the predicted class as A-Right, and the key points of user's hand.	System detects only the first hand shown, display the predicted class as A-Right, and the key points of user's hand.	Pass

6.2.9 Speech-to-Sign Translator Function Testing

Table 6. 9 Speech-to-Sign Translator Function Testing

No.	Test Case	Test Data	Expected Output	Actual Output	Result (pass/fail)
1	Selects the "Speak" button	Speech: "Thank you"	System starts the speech to text translator, display the text translated- "Thank you".	System starts the speech to text translator, display the text translated- "Thank you".	Pass
2	Finish speaking	Speech: "Thank you"	System displays the sign video that correspond to the translated text.	System displays the sign video that correspond to the translated text.	Pass
3	No matching video	Speech: "Congratulations"	System prompts "no matching videos found"	System prompts "no matching videos found"	Pass
4	Selects pause video button	Speech: "Thank you" Video: thankyou.mp4	System pauses the bird sign video.	System pauses the bird sign video.	Pass
5	Selects rewind video button	Speech: "Thank you" Video: thankyou.mp4	System rewinds the bird sign video.	System rewinds the bird sign video.	Pass
6	Selects the fast forward button	Speech: "Thank you" Video: thankyou.mp4	System fast forwards the bird sign video.	System fast forwards the bird sign video.	Pass
7	Selects the play button	Speech: "Thank you"	System replays the bird sign video.	System replays the bird sign video.	Pass

	after video finishes.	Video: thankyou.mp4			

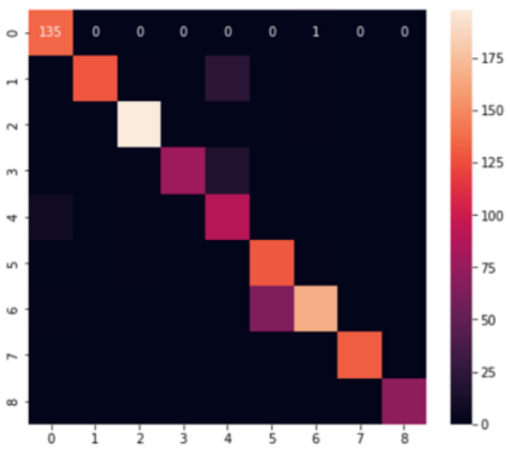
6.2.10 Static Sign Recognition Model's Performance Evaluation

To evaluate the performance of the recognition model, a classification result and confusion matrix has been produced to visualize the results clearly (Figure 6.1).

Interpretation of Metrics:

- **Precision:** Measures the accuracy of positive predictions.
- **Recall:** Measures the ability of the model to correctly identify positive instances.
- **F1-Score:** Harmonic mean of precision and recall, providing a balanced measure.
- **Support:** Number of actual occurrences of each class in the dataset.
- **Accuracy:** Measures the overall correctness of predictions across all classes.
- **Macro Average:** The average precision, recall, and F1-score across all classes, without considering class imbalance.
- **Weighted Average:** The weighted average of precision, recall, and F1-score, considering class imbalance.

The model achieves an overall accuracy of 90%, indicating that 90% of predictions across all classes were correct. Based on the classification report and confusion matrix, the model demonstrates a strong performance with high precision, recall and F1-scores for most classes. Overall, the model's accuracy of 90% and balanced metrics across classes suggests that it performs well.



Classification Report

	precision	recall	f1-score	support
0	0.92	0.99	0.96	136
1	1.00	0.84	0.91	153
2	0.98	0.99	0.99	199
3	1.00	0.80	0.89	97
4	0.69	0.90	0.78	99
5	0.67	0.98	0.80	131
6	0.98	0.72	0.83	233
7	1.00	1.00	1.00	131
8	1.00	1.00	1.00	71
accuracy			0.90	1250
macro avg	0.91	0.91	0.91	1250
weighted avg	0.92	0.90	0.90	1250

Figure 6. 1 Confusion Matrix and Classification report

To measure the efficiency of the real-time recognition function, inference and measure timing are performed. Based on the results in Figure 6.2, the execution time is 0.003 seconds, indicating that the code block, including invoking the TensorFlow Lite interpreter and performing inference only took 3 milliseconds to complete. This suggest that the inference process is fast enough to provide real-time or near-real-time responses. Overall, the inference testing indicates the efficient process and responsiveness of the system.

```
import time

interpreter = tf.lite.Interpreter(model_path=tflite_save_path)
interpreter.allocate_tensors()
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
interpreter.set_tensor(input_details[0]['index'], np.array([X_test[0]]))

# Introduce a delay of 1 second (1000 milliseconds)
time.sleep(1)

# Perform inference and measure timing
start_time = time.time()
interpreter.invoke()
end_time = time.time()
execution_time = end_time - start_time

# Print execution time
print(f"Execution time: {execution_time:.3f} seconds")
```

```
Execution time: 0.003 seconds
```

Figure 6. 2 Execution time of model

6.3 Project Challenges

During the development of the mobile application using Android Studio, challenges were encountered due to limited familiarity with the Android Studio environment. Despite the availability of online resources, significant time was required to attain a comprehensive understanding of its functionalities. Therefore, effective time management for self-learning proved essential.

Additionally, in the initial stages, the recognition module was developed using Python with OpenCV and a webcam due to greater familiarity with these technologies. However, compatibility issues arose with the webcam, necessitating a transition to another PC to resolve the issue. This highlights the importance of considering minimum hardware requirements before commencing a project.

Furthermore, integrating the Mediapipe framework and recognition model into Android Studio posed challenges. Limited online resources were available regarding the integration process, with only example codes provided on the official page. Consequently, considerable time was dedicated to comprehending the complex code structure of the framework and model integration.

6.4 Objectives Evaluation

After conducting test and validation on the system's functionalities, evaluation of each objective was made based on the outcomes:

Objective (a): Develop an interactive sign language learning mobile application.

This objective has been successfully achieved. The curriculum-based courses have effectively covered the main categories of sign language ranging from beginners to advanced levels. The inclusion of video demonstration has significantly contributed to users' sign vocabulary mastery. User interactions with the interactive modules such as quizzes leads to an improved sign learning progress, the user progress function highlight the effectiveness of these modules in facilitating learning and assessment. Besides, the speech to sign translation feature has proven to be beneficial in helping users to learn realistic conversations and enhance their overall communication abilities.

Objective (b): Develop a sign language learning mobile application that provides real-time feedback to users.

The real-time sign recognition and assessment using deep learning algorithms and computer vision have proven to be reliable and effective. The real-time feedback provided by the system helps users to improve their signing accuracy and performance over time, this contributed significantly to users' learning experience and progress.

6.5 Concluding Remark

In this chapter, the evaluation of the system is conducted to assess its overall performance and determine whether it successfully meets the system objectives and goals. The process of evaluation involves the testing of the system's functionalities to verify each component performs as planned. Every aspect of the system's functioning is assessed through extensive user case testing to guarantee that the system accurately and effectively completes the assigned tasks. The evaluation results achieve a highly satisfactory result, highlighting they system's readiness for deployment and use in real-world scenarios.

CHAPTER 7 CONCLUSION AND RECOMMENDATION

7.1 Conclusion

To sum up, this Sign Language learning mobile application with real-time feedback utilizing computer vision and machine learning techniques improves the accessibility of education. Traditional sign language learning methods such as in-person classes and printed materials face limitations in flexibility and interactivity. Although existing sign language learning application improves the conveniency, they still lack personalized feedback due to the absence of computer vision and machine learning.

This project addresses the shortcomings of inflexible in-person classes and printed materials by integrating interactive quizzes, video demonstrations and real-time feedback. It plays a crucial role in improving social inclusion of the Deaf Community. This project aims to develop a mobile application using the Model-View-Controller (MVC) architecture. It strives to enable real-time feedback on sign language gestures to overcome the limitations of the existing learning application. The project's objectives include creating a bit-sized learning modules with proper video demonstration, interactive quizzes and real-time recognition modules to enable self-directed practice.

7.2 Recommendation

Apart from the implemented feature, the sign language recognition could be enhanced by adding dynamic sign recognition. Dynamic sign recognition involves capturing and interpreting continuous hand gestures. By implementing dynamic recognition, this system can provide a more accurate and detailed feedback to users, improving user experience and learning outcomes. This would greatly improve the application's ability to assist users in developing sign language proficiency.

Another valuable enhancement that could be made to this system is to integrate Sign-to-Speech translation capabilities into the system. While the current project focuses on Speech-to-Sign translation, adding Sign-to-Speech translation would offer a comprehensive two-way communication capability. Implementing Sign-to-Speech translation requires dynamic sign language recognition capabilities, which aligns with the recommendation for dynamic recognition mentioned above.

Last but not least, another recommendation for future development of this system is to incorporate face detection and recognition along with hand landmark detection. In sign language communication, facial expressions are essential for expressing emotions and grammatical meaning. By combining both face and hand recognition, the application can provide a more accurate interpretation of sign language recognition.

REFERENCES

[1] "Projected number of people with hearing loss worldwide 2019," Statista. https://www.statista.com/statistics/888569/number-of-people-with-hearing-loss-worldwide-projections/#statisticContainer

[2] "Rate of hearing loss by severity worldwide 2019," Statista. https://www.statista.com/statistics/1235595/rate-hearing-loss-by-severity-worldwide/

[3] Sign Solutions, "What are the different types of sign language?" Sign Solutions, May 03, 2021. https://www.signsolutions.uk.com/what-are-the-different-types-of-sign-language/

[4] "Introduction to MediaPipe | LearnOpenCV," Mar. 01, 2022. https://learnopencv.com/introduction-to-mediapipe/

[5] S. Garg, "9 Sign Language Learning Apps for Beginners," Geekflare, Dec. 09, 2022. https://geekflare.com/sign-language-learning-apps/

[6] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A. B. Gil-González, and J. M. Corchado, "Deepsign: Sign Language Detection and Recognition Using Deep Learning," Electronics (Switzerland), vol. 11, no. 11, Jun. 2022, doi: 10.3390/electronics11111780

[7] M. Al-Qurishi, T. Khalid, and R. Souissi, "Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues," IEEE Access, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 126917–126951, 2021. doi: 10.1109/ACCESS.2021.3110912

[8] "5 Best Apps For Learning Sign Language | Nagish," nagish.com. https://nagish.com/post/5-best-apps-for-learning-sign-language

[9] R. F. published, "SignSchool review," TopTenReviews, Oct. 28, 2021. https://www.toptenreviews.com/signschool-review

[10] SignSchool, "SignSchool | Learn American Sign Language Online," SignSchool, 2020.

[11] "Lingvano: Sign Language - ASL - Apps on Google Play," play.google.com. https://play.google.com/store/apps/details?id=com.lingvano.app&hl=en&gl=US&pli=1

[12] R. F. published, “Lingvano review,” TopTenReviews, Oct. 28, 2021. https://www.toptenreviews.com/lingvano-review

[13] A. Tayade and A. Halder, “Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning,” International Journal of Research Publication and Reviews, vol. 2, no. 5, 2021, doi: 10.13140/RG.2.2.32364.03203.

[14] V. Dey, “Understanding the Importance of Data Cleaning and Normalization,” Analytics India Magazine, Sep. 12, 2021. https://analyticsindiamag.com/understanding-the-importance-of-data-cleaning-and-normalization/

[15] K. Bantupalli and Y. Xie, “American Sign Language Recognition using Deep Learning and Computer Vision,” IEEE Conference Publication. https://ieeexplore.ieee.org/document/8622141

[16] S. Mhatre, S. Joshi and H. B. Kulkarni, “Sign Language Detection using LSTM,” IEEE Conference Publication. https://ieeexplore.ieee.org/document/10080705

[17] S. A. M. A. S. Senanayaka, R. A. D. B. S. Perera, W. Rankothge, S. S. Usgalhewa, H. D. Hettihewa, and P. K. W. Abeygunawardhana, “Continuous American Sign Language Recognition Using Computer Vision And Deep Learning Technologies,” in 2022 IEEE Region 10 Symposium, TENSYMP 2022, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/TENSYMP54529.2022.9864539.

[18] M. W. Foysol, S. E. A. Sajal, and M. J. Alam, “Vision-based Real Time Bangla Sign Language Recognition System Using MediaPipe Holistic and LSTM,” in Proceedings of 3rd IEEE International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2023, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 19–24. doi: 10.1109/ICCIKE58312.2023.10131801.

[19] “Gesture recognition task guide | MediaPipe,” Google for Developers. https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer

[20] LucianoSphere, “Exquisite hand and finger tracking in web browsers with MediaPipe’s machine learning models,” Medium, Feb. 15, 2022. https://towardsdatascience.com/exquisite-hand-and-finger-tracking-in-web-browsers-with-mediapipes-machine-learning-models-2c4c2beee5df

APPENDIX

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: October 2023	**Study week no.: 2**
Student Name & ID: Lee Xiao Xu Alexis 20ACB02090	
Supervisor: Dr Ng Hui Fuang	
Project Title: Mobile Application for Sign Language Learning With Real Time Feedback	

1. WORK DONE

- Revise FYP1 Report

- Plan the new functions to be include in FYP2

2. WORK TO BE DONE

- Research of relevant information on the dynamic recognition model

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

So far so good.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: October 2023	**Study week no.: 4**
Student Name & ID: Lee Xiao Xu Alexis 20ACB02090	
Supervisor: Dr Ng Hui Fuang	
Project Title: Mobile Application for Sign Language Learning With Real Time Feedback	

1. WORK DONE - Research on dynamic sign recognition - Try out the model using webcam
2. WORK TO BE DONE - Find way to integrate dynamic recognition model into android studio
3. PROBLEMS ENCOUNTERED **-** Model for dynamic recognition are written in python, unable to export and integrate it into android studio.
4. SELF EVALUATION OF THE PROGRESS - On track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: October 2023	**Study week no.: 6**
Student Name & ID: Lee Xiao Xu Alexis 20ACB02090	
Supervisor: Dr Ng Hui Fuang	
Project Title: Mobile Application for Sign Language Learning With Real Time Feedback	

1. WORK DONE - Implement the dynamic sign comparison and feedback on android
2. WORK TO BE DONE - Collect more dataset for dynamic sign comparison purpose - Find way to integrate dynamic recognition model into android studio
3. PROBLEMS ENCOUNTERED **-**
4. SELF EVALUATION OF THE PROGRESS - so far so good

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: October 2023	**Study week no.: 8**
Student Name & ID: Lee Xiao Xu Alexis 20ACB02090	
Supervisor: Dr Ng Hui Fuang	
Project Title: Mobile Application for Sign Language Learning With Real Time Feedback	

1. WORK DONE
- Added user proficiency level to the system
2. WORK TO BE DONE
- Add speech to sign function into the system - Research on speech to sign API
3. PROBLEMS ENCOUNTERED
-
4. SELF EVALUATION OF THE PROGRESS
- on track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: October 2023	**Study week no.: 10**
Student Name & ID: Lee Xiao Xu Alexis 20ACB02090	
Supervisor: Dr Ng Hui Fuang	
Project Title: Mobile Application for Sign Language Learning With Real Time Feedback	

1. WORK DONE - Successfully implement the speech-to-sign translation function - Added more data to the system
2. WORK TO BE DONE - Written report
3. PROBLEMS ENCOUNTERED **-**
4. SELF EVALUATION OF THE PROGRESS **-** So far so good

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

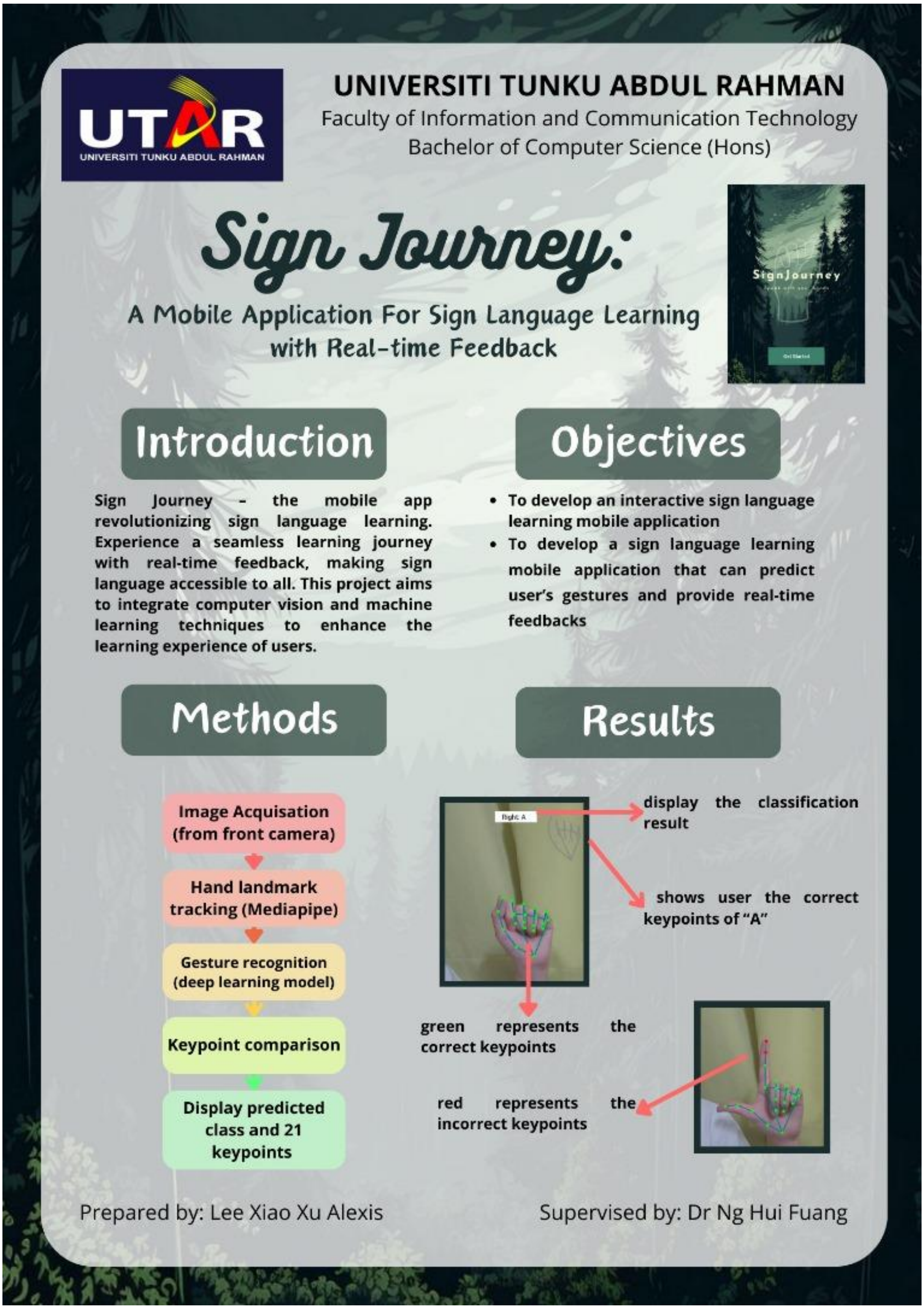
Trimester, Year: October 2023	**Study week no.: 12**
Student Name & ID: Lee Xiao Xu Alexis 20ACB02090	
Supervisor: Dr Ng Hui Fuang	
Project Title: Mobile Application for Sign Language Learning With Real Time Feedback	

1. WORK DONE - Completed some section of the FYP 2 Report
2. WORK TO BE DONE - Conduct system testing
3. PROBLEMS ENCOUNTERED **-**
4. SELF EVALUATION OF THE PROGRESS - On track

Supervisor's signature

Student's signature

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
UNIVERSITI TUNKU ABDUL RAHMAN
Faculty of Information and Communication Technology
Bachelor of Computer Science (Hons)
Sign Journey:
A Mobile Application For Sign Language Learning with Real-time Feedback
SignJourney
Introduction
Sign Journey - the mobile app revolutionizing sign language learning. Experience a seamless learning journey with real-time feedback, making sign language accessible to all. This project aims to integrate computer vision and machine learning techniques to enhance the learning experience of users.
Objectives
• To develop an interactive sign language learning mobile application
• To develop a sign language learning mobile application that can predict user's gestures and provide real-time feedbacks
Methods
Image Acquisation (from front camera)
Hand landmark tracking (Mediapipe)
Gesture recognition (deep learning model)
Keypoint comparison
Display predicted class and 21 keypoints
Results
display the classification result
shows user the correct keypoints of "A"
green represents the correct keypoints
red represents the incorrect keypoints
Prepared by: Lee Xiao Xu Alexis
Supervised by: Dr Ng Hui Fuang

PLAGIARISM CHECK RESULT

FYP2Report(turnitin).docx

ORIGINALITY REPORT

8%	6%	4%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	2%
2	Nike Sartika, Yuda Sukmana, Rosa Eliviani, Rabihi Awaludin. "Smart Online Class Monitoring Service Design", 2022 8th International Conference on Wireless and Telematics (ICWT), 2022 Publication	<1%
3	www.coursehero.com Internet Source	<1%
4	ijrpr.com Internet Source	<1%
5	Kshitij Bantupalli, Ying Xie. "American Sign Language Recognition using Deep Learning and Computer Vision", 2018 IEEE International Conference on Big Data (Big Data), 2018 Publication	<1%
6	S.A.M.A.S Senanayaka, R.A.D.B.S Perera, W. Rankothge, S.S. Usgalhewa, H.D Hettihewa,	<1%

	P.K.W. Abeygunawardhana. "Continuous American Sign Language Recognition Using Computer Vision And Deep Learning Technologies", 2022 IEEE Region 10 Symposium (TENSYMP), 2022 Publication	
7	www.mdpi.com Internet Source	<1 %
8	"Artificial Intelligence: Theory and Applications", Springer Science and Business Media LLC, 2024 Publication	<1 %
9	fastercapital.com Internet Source	<1 %
10	Sanjusha Cheemakurthi, Min Chen. "Master-ASL: Sign Language Learning and Assessment System", 2023 IEEE International Conference on Big Data (BigData), 2023 Publication	<1 %
11	www.notebook.hu Internet Source	<1 %
12	www.statista.com Internet Source	<1 %
13	dokumen.pub Internet Source	<1 %
14	kipdf.com Internet Source	

		<1 %
15	Nhidayu Kamarudin, Eileen Ong, WS Chung. "Science in mobile apps for education: tackling online learning within pandemic outbreak", International Journal of Engineering & Technology, 2020 Publication	<1 %
16	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
17	www.ajol.info Internet Source	<1 %
18	researchonline.ljmu.ac.uk Internet Source	<1 %
19	www.ijraset.com Internet Source	<1 %
20	www.ijritcc.org Internet Source	<1 %
21	R Sreemathy, MP Turuk, S Chaudhary, K Lavate, A Ushire, S Khurana. "Continuous Word Level Sign Language Recognition using an Expert System Based on Machine Learning", International Journal of Cognitive Computing in Engineering, 2023 Publication	<1 %

22 Shrey Sitapara, Akarsha D.P., Mahika Jain, Sakshi Singh, Nitesh Gupta. "Novel Approach for Real-Time Exercise Posture Correction Using Computer Vision and Cnn", 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), 2023 <1 %
Publication

23 Ye Chow Kuang. "Real-Time Malaysian Sign Language Translation using Colour Segmentation and Neural Network", 2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007, 05/2007 <1 %
Publication

24 Ching-Ta Lu, Yu-Chun Liu, Ying-Chen Pan. "An intelligent playback control system adapted by body movements and facial expressions recognized by OpenPose and CNN", Multimedia Tools and Applications, 2023 <1 %
Publication

25 Jefferson Amará, Victor Ströele, Regina Braga, José Maria N. David. "Chapter 31 From Context toForecast: Ontology-Based Data Integration andAI forEvents Prediction", Springer Science and Business Media LLC, 2024 <1 %
Publication

fict.utar.edu.my

26	Internet Source	<1 %
27	play.google.com Internet Source	<1 %
28	Mohammed, Bandar, and Malek Mouhoub. "Evaluation of an online shopping system under Preferences and Constraints", 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), 2014. Publication	<1 %
29	Sivaraj Selvaraj. "Chapter 1 Understanding the Laravel Ecosystem", Springer Science and Business Media LLC, 2024 Publication	<1 %
30	export.arxiv.org Internet Source	<1 %
31	Hirohiko Sagawa. "A Method for Analyzing Spatial Relationships Between Words in Sign Language Recognition", Lecture Notes in Computer Science, 1999 Publication	<1 %
32	umpir.ump.edu.my Internet Source	<1 %
33	Mohib Ullah, Rafi Ullah Khan, Irfan Ullah Khan, Nida Aslam, Sumayh S. Aljameel, Muhammad Inam Ul Haq, Muhammad	<1 %

	Arshad Islam. "Profile Aware ObScure Logging (PaOSLo): A Web Search Privacy-Preserving Protocol to Mitigate Digital Traces", Security and Communication Networks, 2022 Publication	
34	dev.to Internet Source	<1 %
35	eprints.utm.my Internet Source	<1 %
36	max-success.eu Internet Source	<1 %
37	theses.gla.ac.uk Internet Source	<1 %
38	www.cse.unsw.edu.au Internet Source	<1 %
39	huggingface.co Internet Source	<1 %
40	pubmed.ncbi.nlm.nih.gov Internet Source	<1 %
41	www.preprod.pointe-noire.agency Internet Source	<1 %
42	www.researchsquare.com Internet Source	<1 %

43	Betsy Angelin S, Nithya E, Devianugraga M S. "A Survey on Conversion of Audio and Text to Sign Language and vice versa using Deep learning", 2023 9th International Conference on Smart Structures and Systems (ICSSS), 2023 Publication	<1 %
44	Jasitha P, Pournami P N. "Glomeruli Detection Using Faster R-CNN and CenterNet", 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), 2023 Publication	<1 %
45	Liliia Ivanska, Tetyana Korotyeyeva. "Mobile real-time gesture detection application for sign language learning", 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022 Publication	<1 %
46	Lim, Kian Ming, Alan W.C. Tan, and Shing Chiang Tan. "A feature covariance matrix with serial particle filter for isolated sign language recognition", Expert Systems with Applications, 2016. Publication	<1 %
47	Md. Walid Foysol, Sk. Estiaque Ahmed Sajal, Mohammad Jahangir Alam. "Vision-based Real Time Bangla Sign Language Recognition	<1 %

	System Using MediaPipe Holistic and LSTM", 2023 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2023 Publication	
48	Mohammad H. Ismail, Shefa A. Dawwd, Fakhrulddin H. Ali. "Arabic Sign Language Detection Using Deep Learning Based Pose Estimation", 2021 2nd Information Technology To Enhance e-learning and Other Application (IT-ELA), 2021 Publication	<1 %
49	Thananan Luangaphirom, Sirirat Lueprasert, Phopthorn Kaewvichit, Siraphong Boonphotsiri et al. "Real-time weight training counting and correction using MediaPipe", Advances in Computational Intelligence, 2024 Publication	<1 %
50	journal.ijmdes.com Internet Source	<1 %
51	scholarsmine.mst.edu Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)

Form Number: FM-IAD-005	Rev No.: 0	Effective Date:	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lee Xiao Xu Alexis
ID Number(s)	20ACB02090
Programme / Course	FICT/CS
Title Of Final Year Project	Mobile Application for Sign Language Learning with Real Time Feedback

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 8 **% Similarity by source** Internet Sources: 6 % Publications: 4 % Student Papers: 0 %	
Number of individual sources listed of more than 3% similarity: 0	

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

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

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

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

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

Signature of Supervisor	Signature of Co-Supervisor
Name: Ng Hui Fuang	Name:
Date: 25/4/2024	Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB02090
Student Name	Lee Xiao Xu Alexis
Supervisor Name	Dr Ng Hui Fuang

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
x	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

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

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)

Date: 25 April 2024