

واصف

MOBILE APPLICATION FOR DEAF AND MUTE INDIVIDUALS

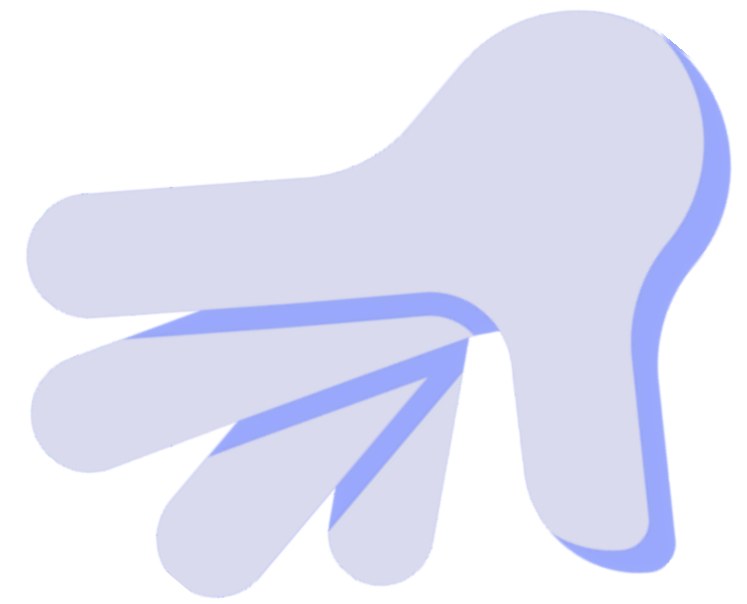
Layan Yahya Almegbil
Abeer Sultan AlAbdullatif
Rimas Turki AlAqil
Haifa Abdulkarim Alkhalaf

Supervised By
Dr. Eman Abdullah Aldakheel

2024 - 2025

جامعة الأميرة
نورة بنت عبدالرحمن
كلية علوم الحاسب والمعلومات





01

Introduction

02

Background Information & Related Work

03

Requirements Determination & Analysis

04

System Design

05

Implementation

06

Testing

07

Conclusion

1.1

PROBLEM STATEMENT & SIGNIFICANCE

MILLIONS OF DEAF AND MUTE INDIVIDUALS FACE CHALLENGES
DAILY DUE TO THE LACK OF ACCESSIBLE COMMUNICATION TOOLS



1.2 PROPOSED SOLUTION

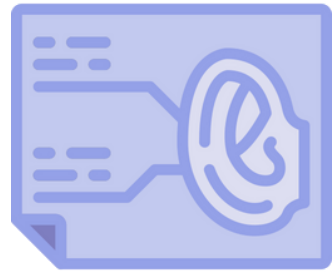


aims to connect deaf and mute individuals with the wider community effectively using clear visuals and customizable features to support inclusivity.

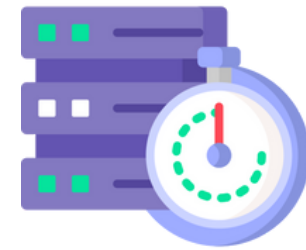
1.3

PROJECT DOMAIN & LIMITATIONS

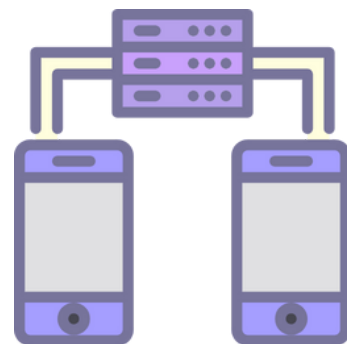
PROJECT DOMAIN



ASSISTIVE TECHNOLOGY



REAL-TIME



**CROSS-PLATFORM
APPLICATION**



**FOCUSE ON ARABIC
LANGUAGE**

PROJECT LIMITATION



LIGHTING SENSITIVITY



**STATIC GESTURE
REPETITION**



**LIMITED VOCABULARY
SCOPE**



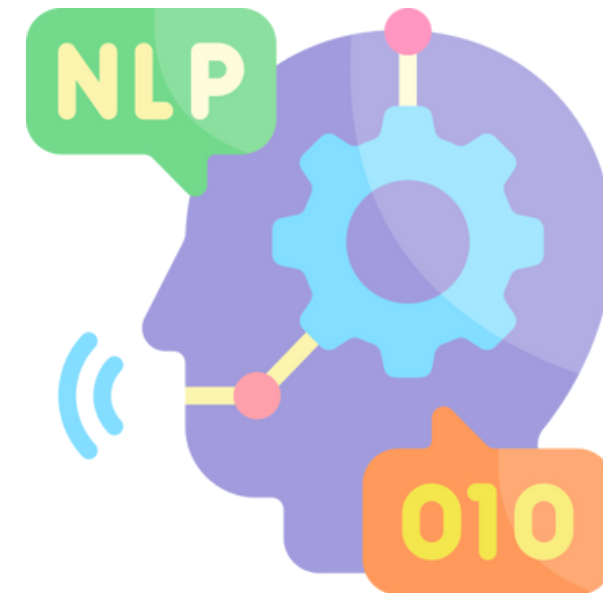
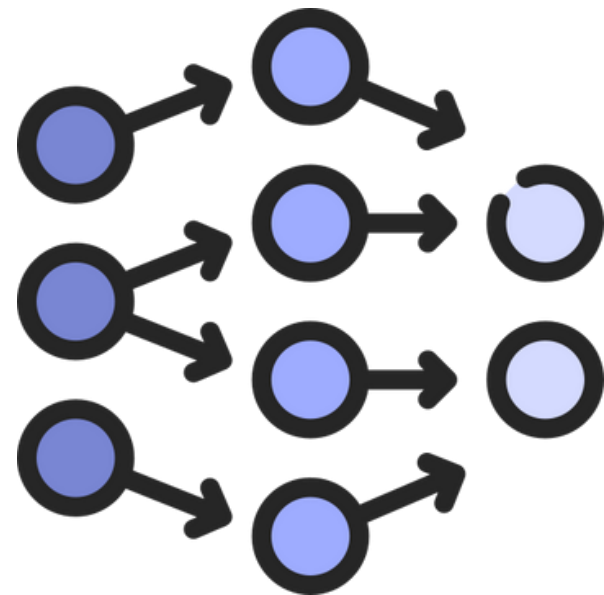
**NO DYNAMIC EMOTION
IN AVATAR**

BACKGROUND INFORMATION & RELATED WORK

2.1

BACKGROUND INFORMATION

Sign language is a **visual-spatial** language that relies on hand gestures, movements, and facial expressions to communicate meaning. It serves as the primary mode of communication for the deaf and mute community, deeply tied to their identity, culture, and linguistic expression. To bridge the communication gap, Artificial Intelligence and computer vision are USING technologies such as:



2.2

RELATED WORK SURVEY

Al Khuzayem et al. proposed a lightweight 3DGCN model for Saudi Sign Language, achieving 97% accuracy with minimal computational cost—ideal for mobile and real-time use.

Alsulaiman et al. enhanced this by adding attention mechanisms and training on 293 signs, improving focus and scalability while maintaining 97% accuracy.

Unlike other models, our solution supports two-way interaction, bridging the communication gap with real-time AI, Generative Language Models, and a cross-platform mobile app.

2.3

PROPOSED & SIMILAR SYSTEM COMPARISON

FEATURES	واصف	EFHAMNI	KSU-SSL	ARSL UTILIZING CNN
Accuracy	97.96%	94.46%	97.25%	99.9%
Approaches	CNN, BiLSTM, BERT, YOLOv8, MediaPipe	CNN, BiLSTM	3DGCN, MediaPipe	CNN, PCA, LDA, LSTM
Direction way	Bldirectional	Bldirectional	One-Way	Bldirectional
Hardware	Mobile Camera	Mobile Camera	High-speed RGB, infrared, mobile cameras	N/A
Dataset	KArSL-502	KSU-SSL	KSU-SSL	N/A
Evaluation metrics	Average Metric: 96.33%	Average Metric: 94.56%	N/A	N/A

REQUIREMENTS DETERMINATION & ANALYSIS

3.1

REQUIREMENTS DETERMINATION & COLLECTION

To shape **WASIF** system, we conducted a targeted user survey exploring public interaction with the deaf/mute community. The survey examined:

- Current communication methods
- Satisfaction with existing tools
- Interest in a new AI-powered app

Top desired features:

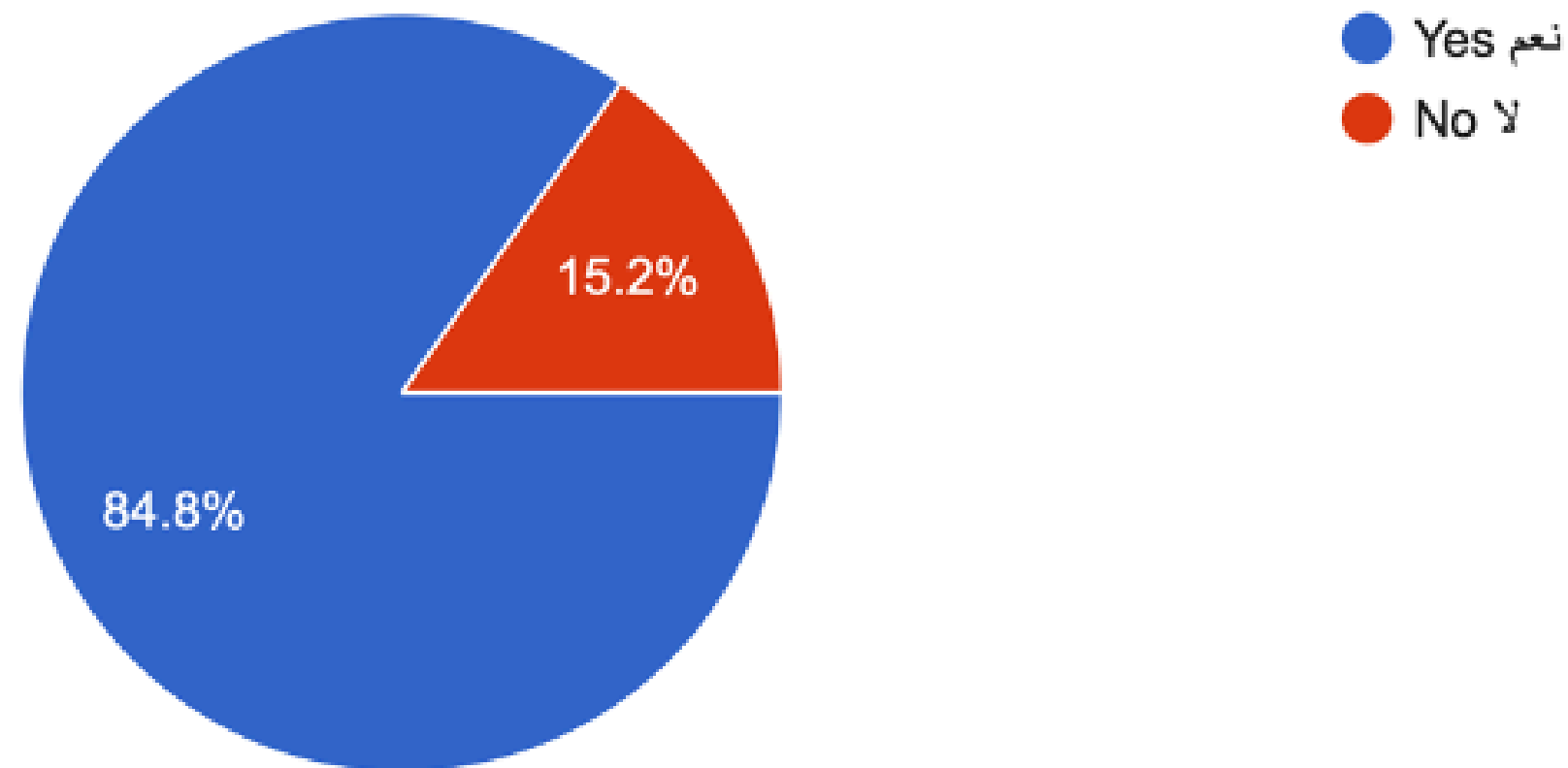
- Real-time voice recognition
- Sign-to-text & text-to-sign translation
- Smart suggestions via AI

3.1

REQUIREMENTS DETERMINATION & COLLECTION

Would you be interested in using a mobile application for easier communication with deaf and mute individuals?
هل أنت مهتم باستخدام تطبيق هاتف محمول لتسهيل التواصل مع الأفراد الصم والبكم؟

217 responses

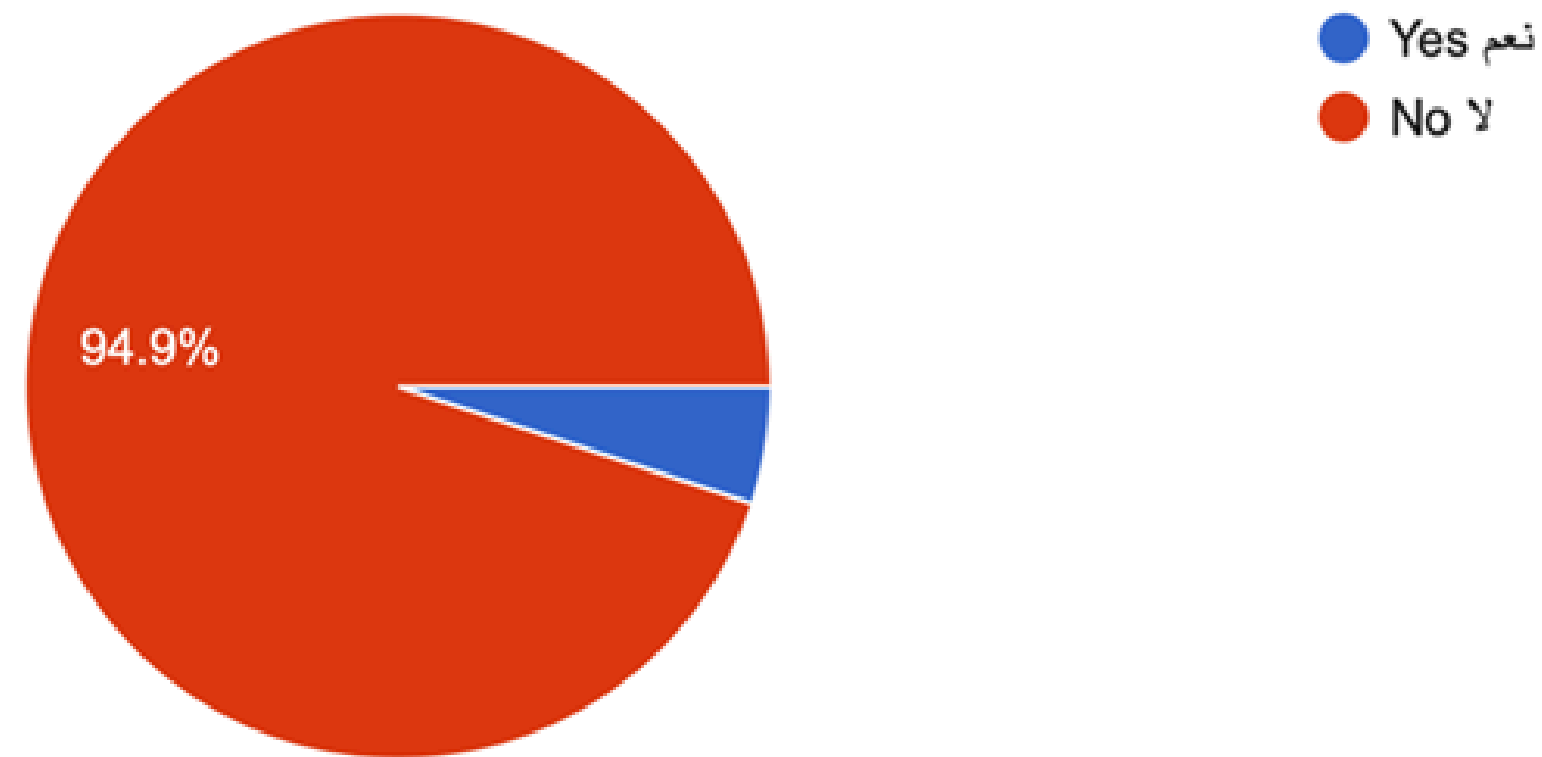


3.1

REQUIREMENTS DETERMINATION & COLLECTION

هل تستخدم
Do you currently use any communication tools or systems for deaf and mute individuals?
حالياً أي أدوات أو أنظمة تواصل مع الأفراد الصم والبكم؟

217 responses



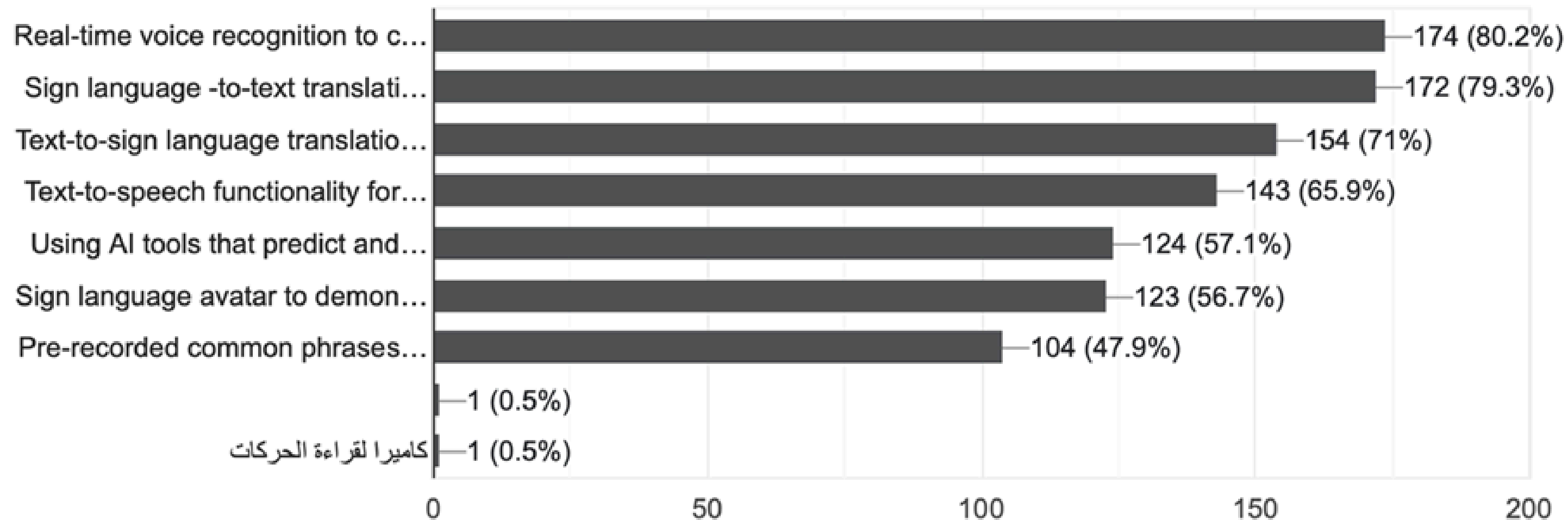
3.1

REQUIREMENTS DETERMINATION & COLLECTION

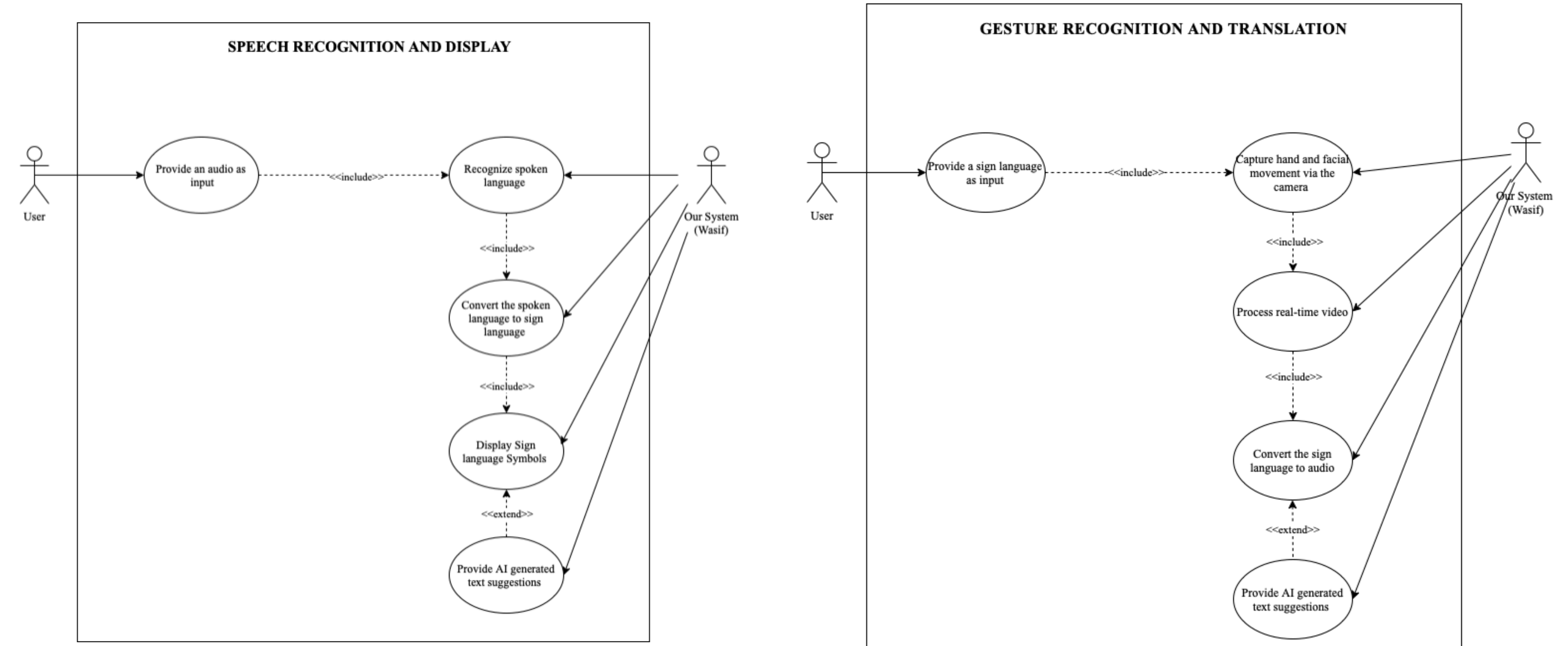
What type of communication features would be most helpful in the app? (Select all that apply) ما نوع

الميزات التي ستكون مفيدة أكثر في التطبيق؟ اختر جميع ما ينطبق

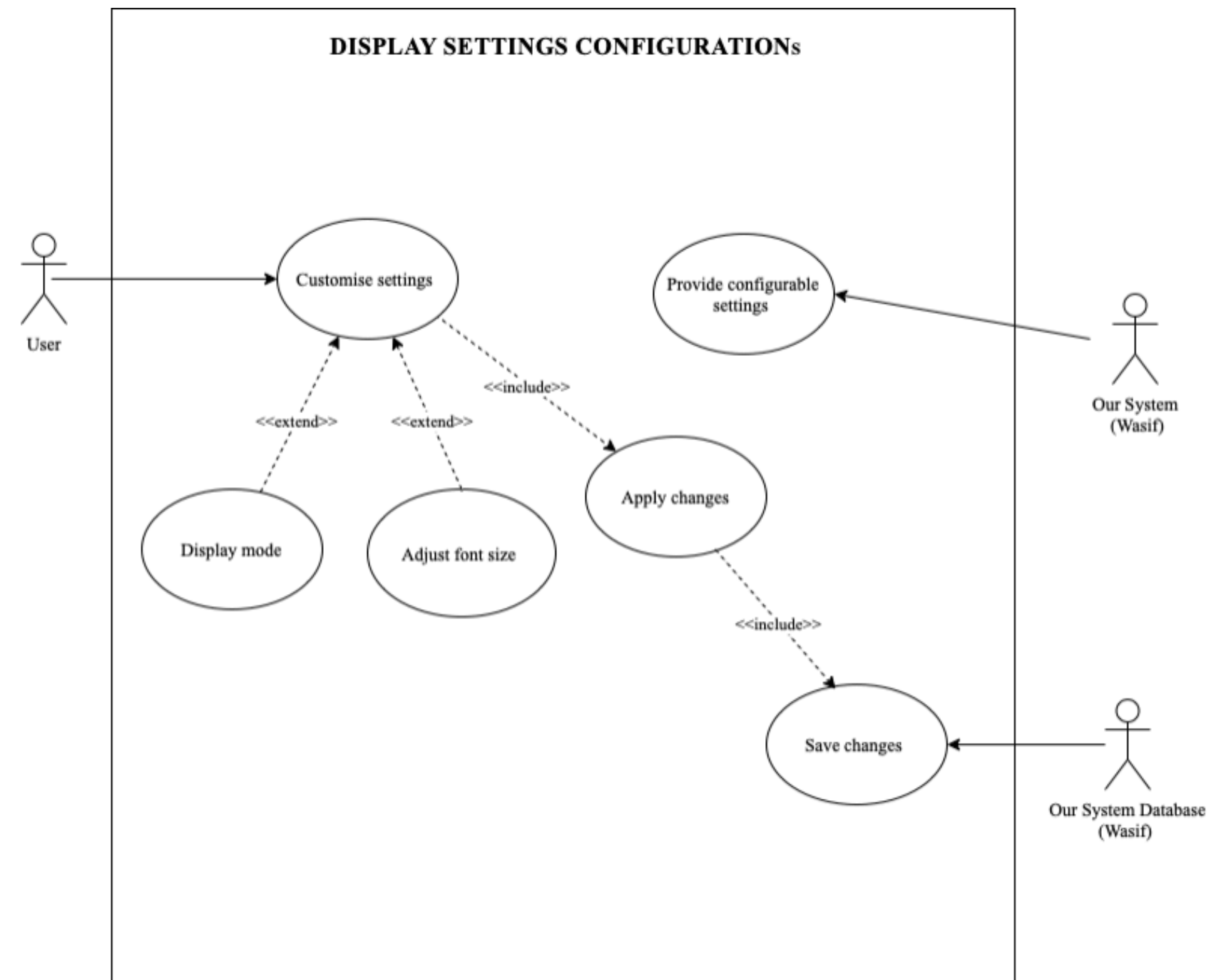
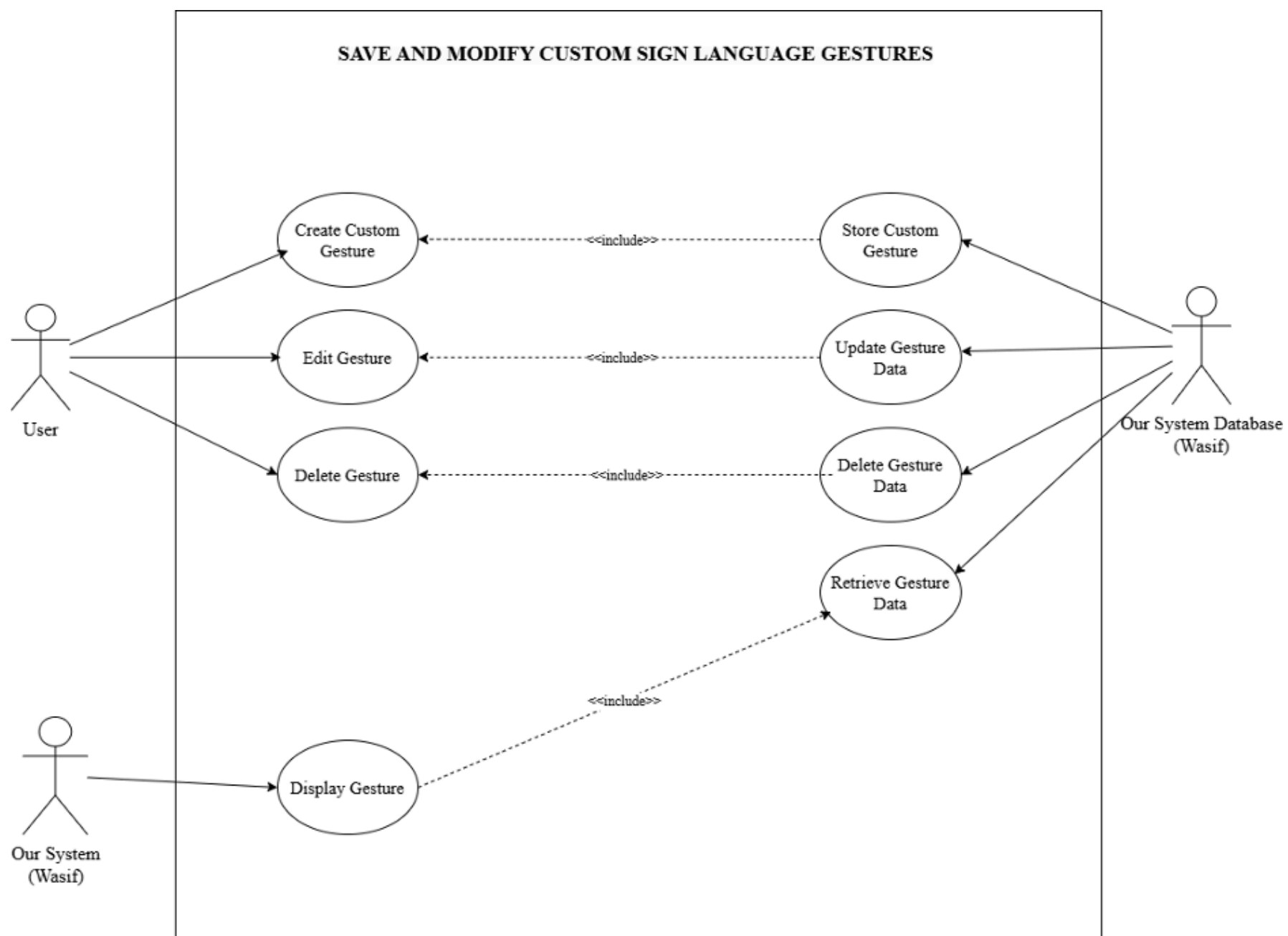
217 responses



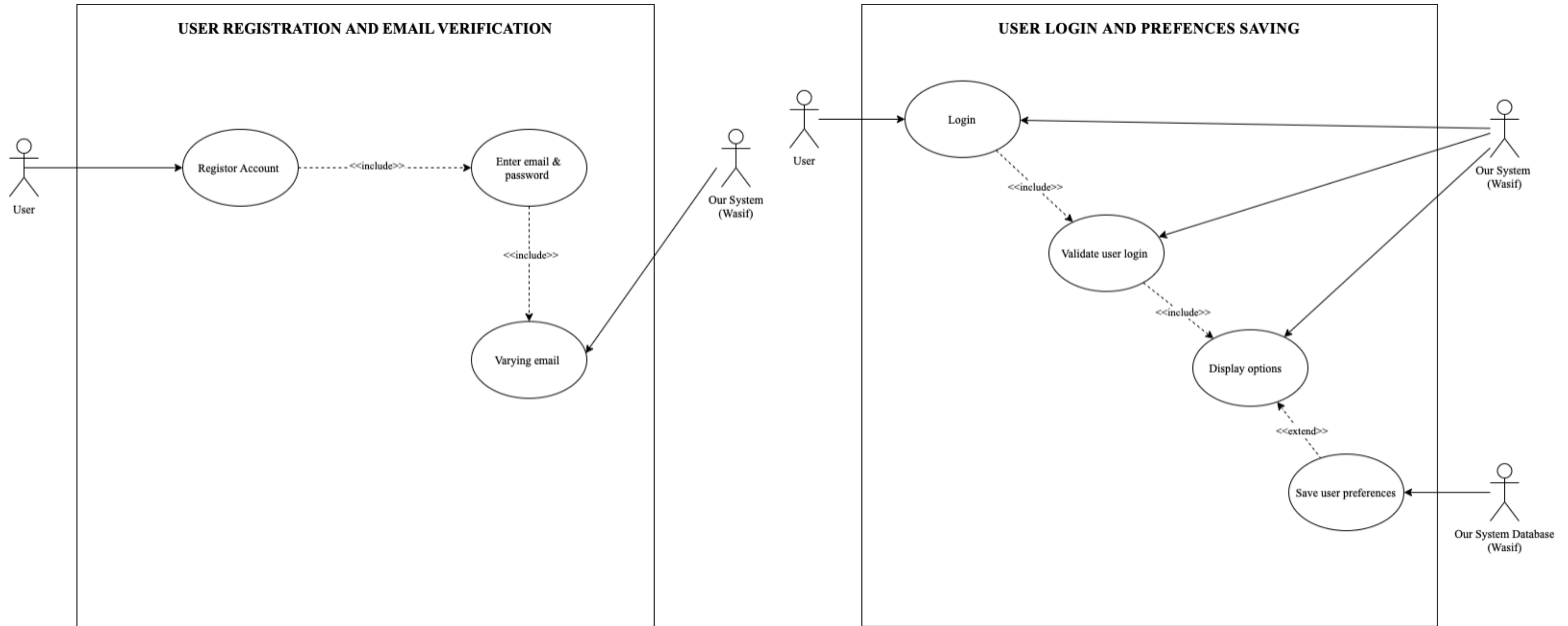
3.2 REQUIREMENT ANALYSIS - USE CASES



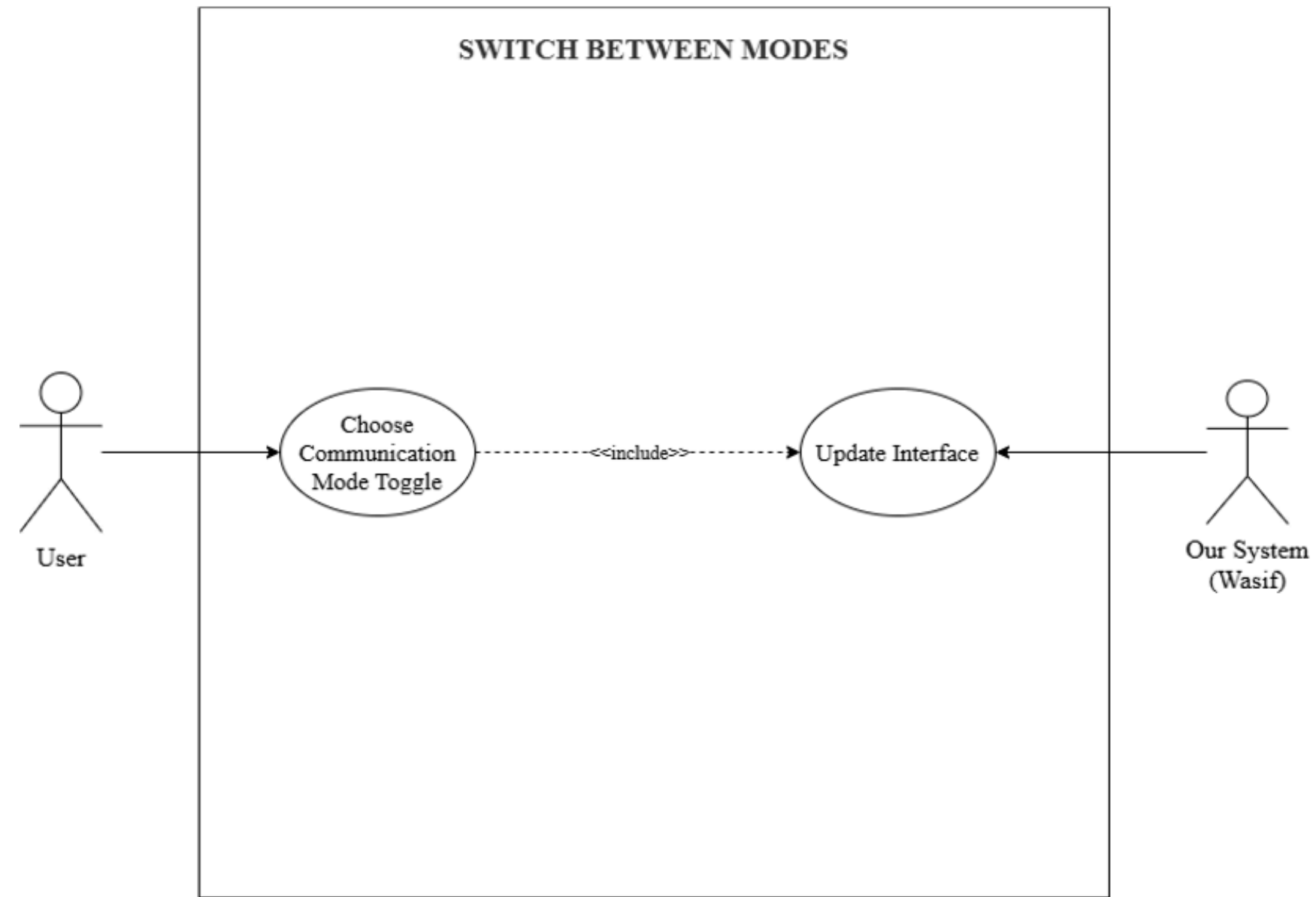
REQUIREMENT ANALYSIS - USE CASES



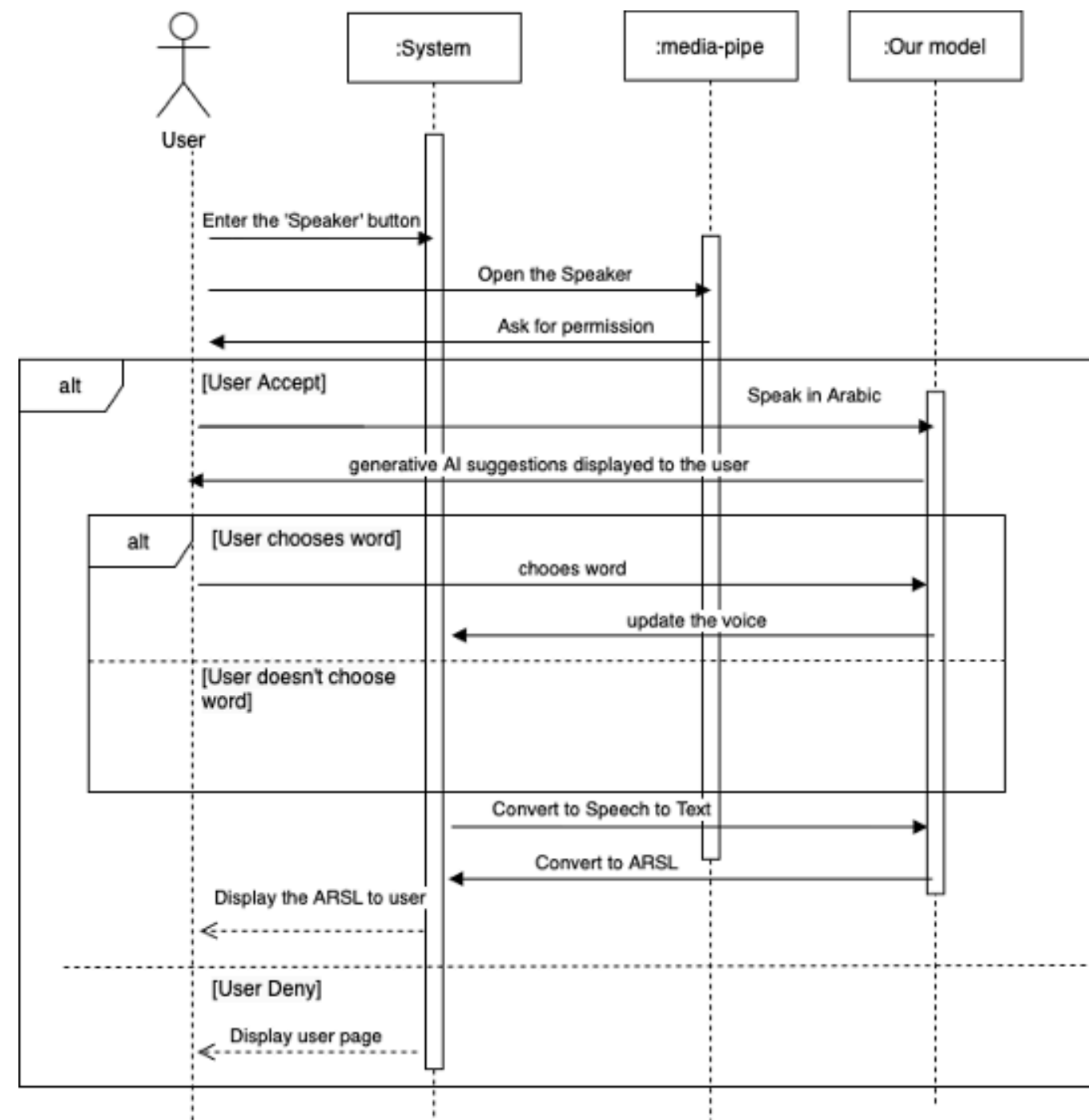
REQUIREMENT ANALYSIS - USE CASES



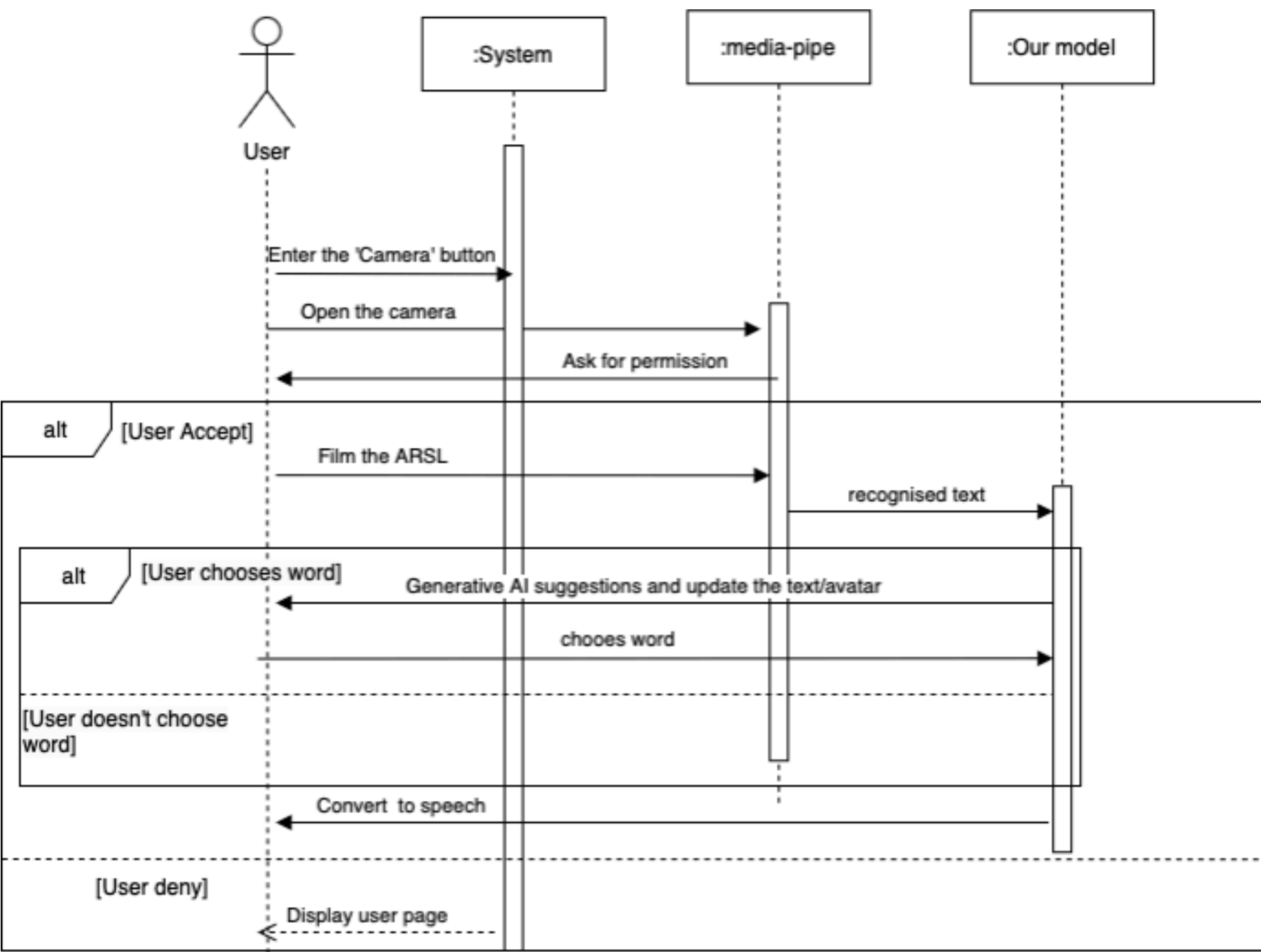
REQUIREMENT ANALYSIS - USE CASES



REQUIREMENT ANALYSIS - SEQUENCE DIAGRAMS



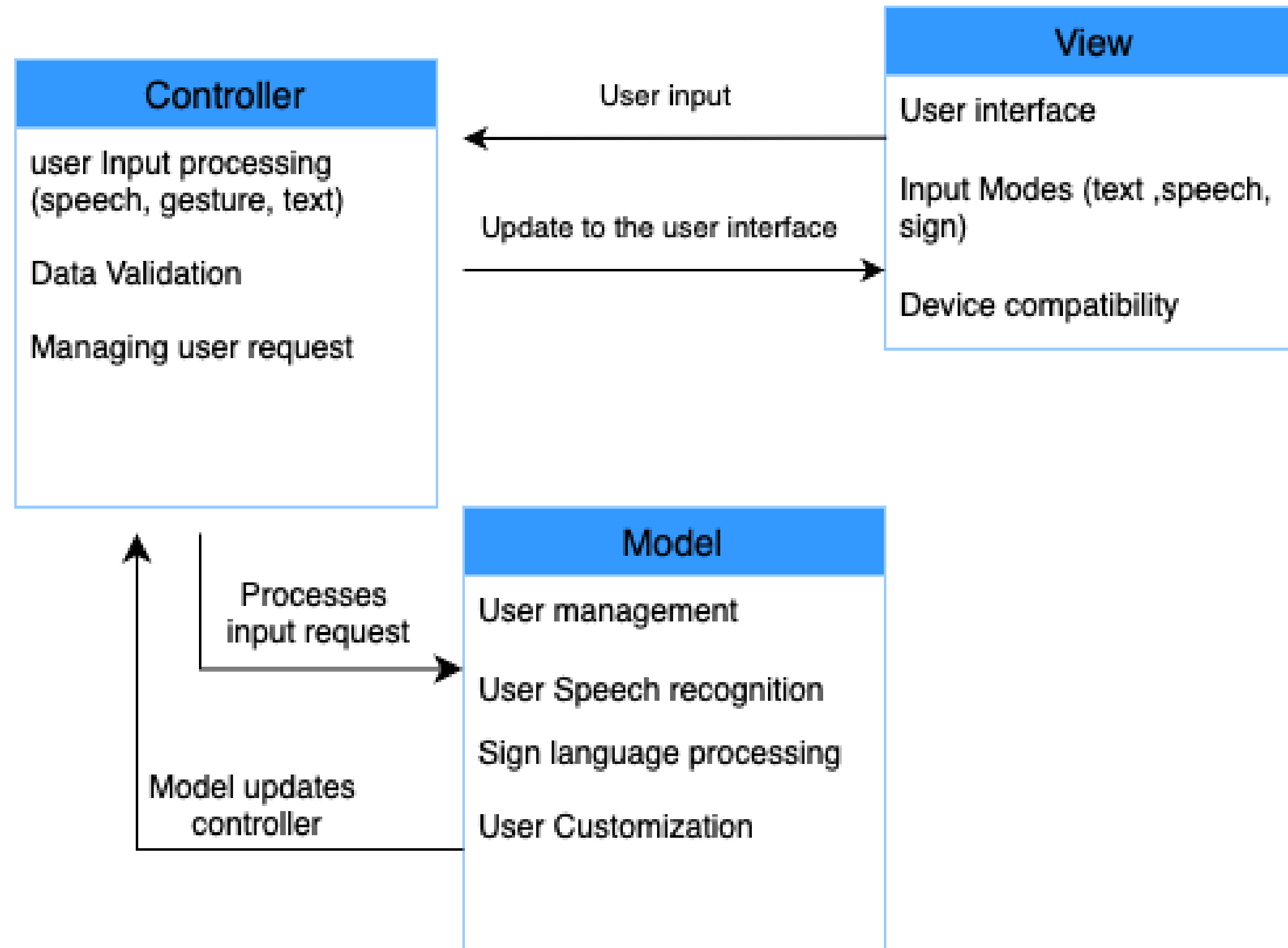
SPEECH TO ARABIC SIGN



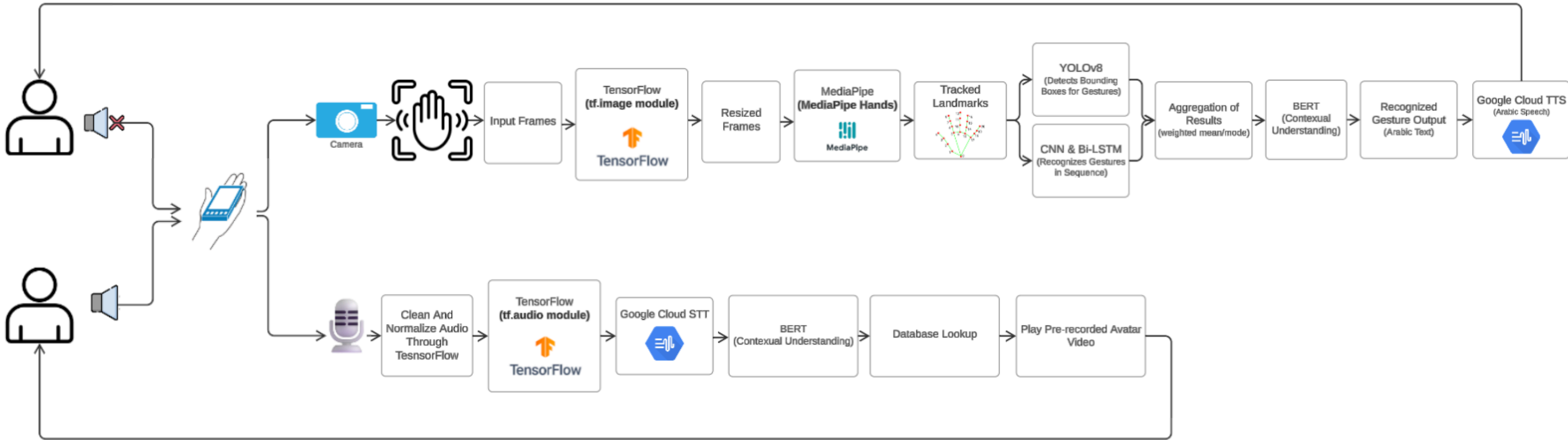
SIGN TO SPEECH

SYSTEM DESIGN

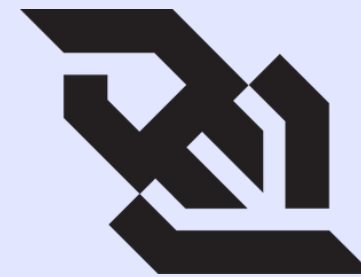
4.1 SYSTEM ARCHITECTURE



SYSTEM ANALYSIS

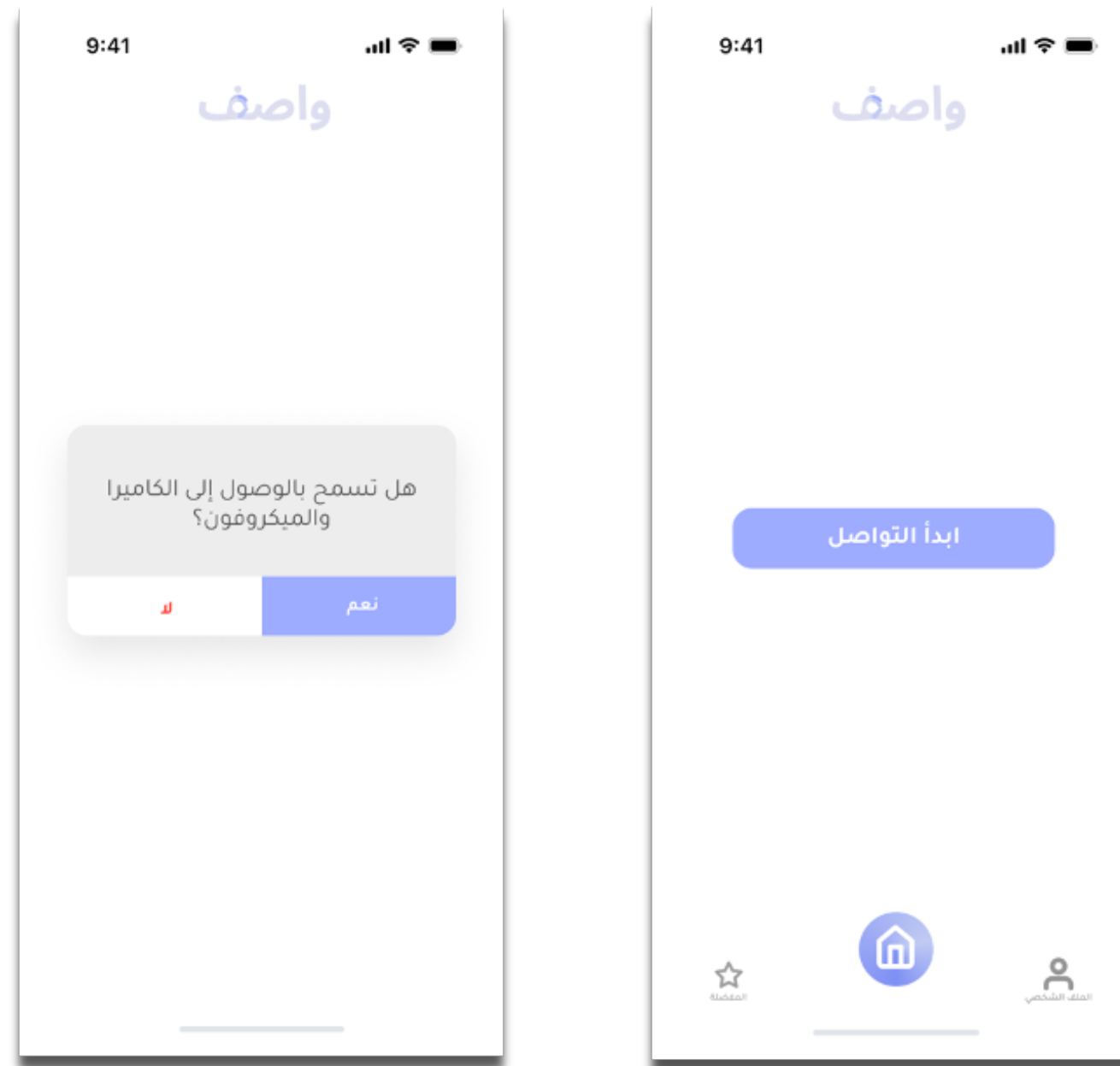


4.2 IMPLEMENTATION REQUIREMENTS



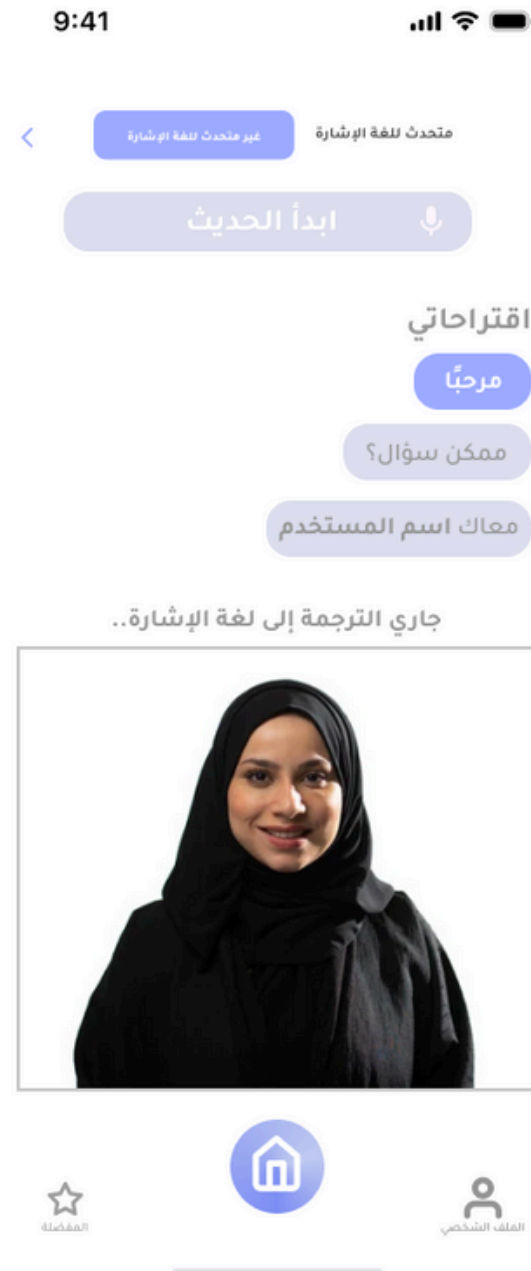
4.3

USER INTERFACE DESIGN



Permissions and Communication Settings

USER INTERFACE DESIGN

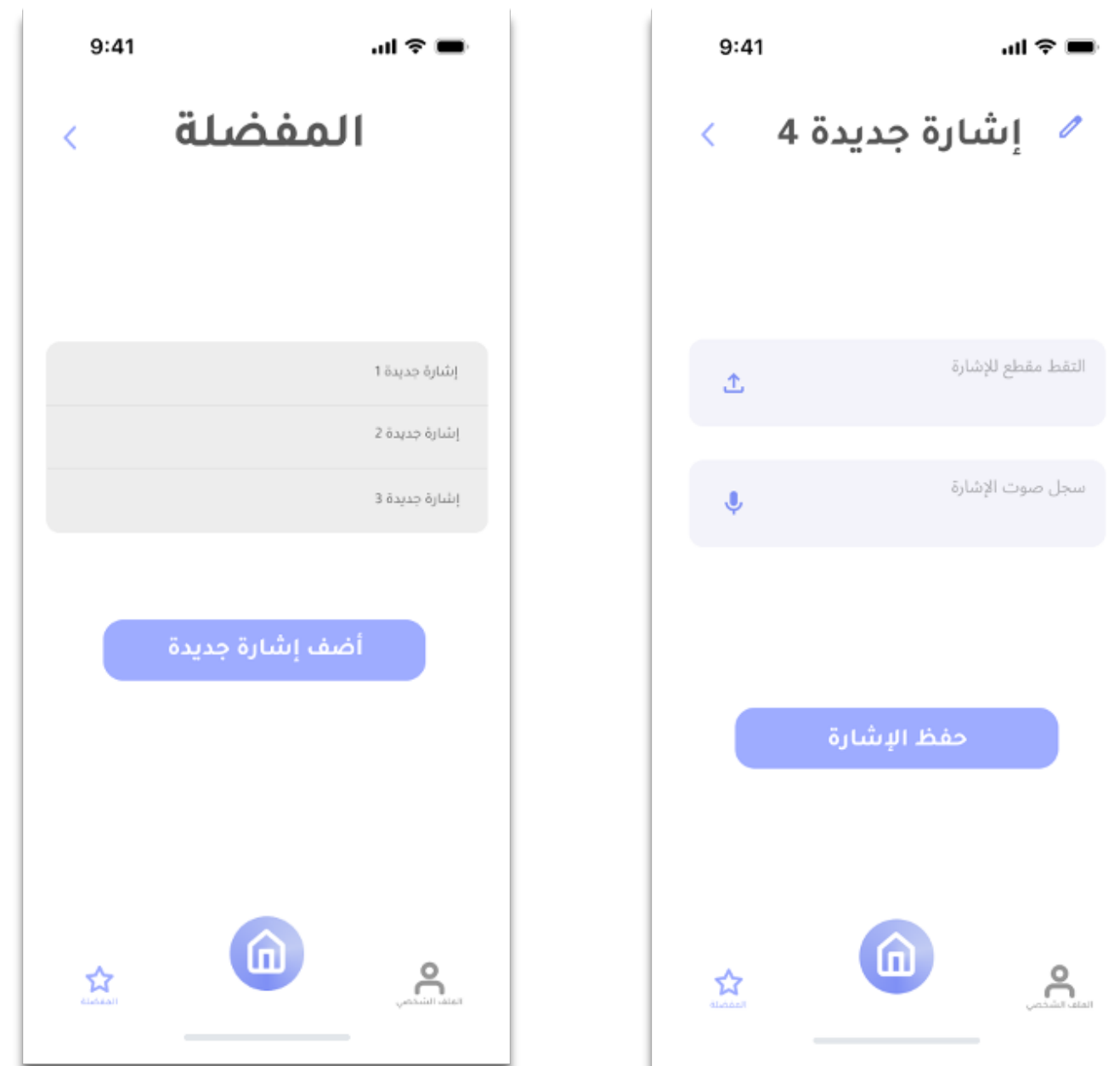


Communication Options for Non-Sign Language Users



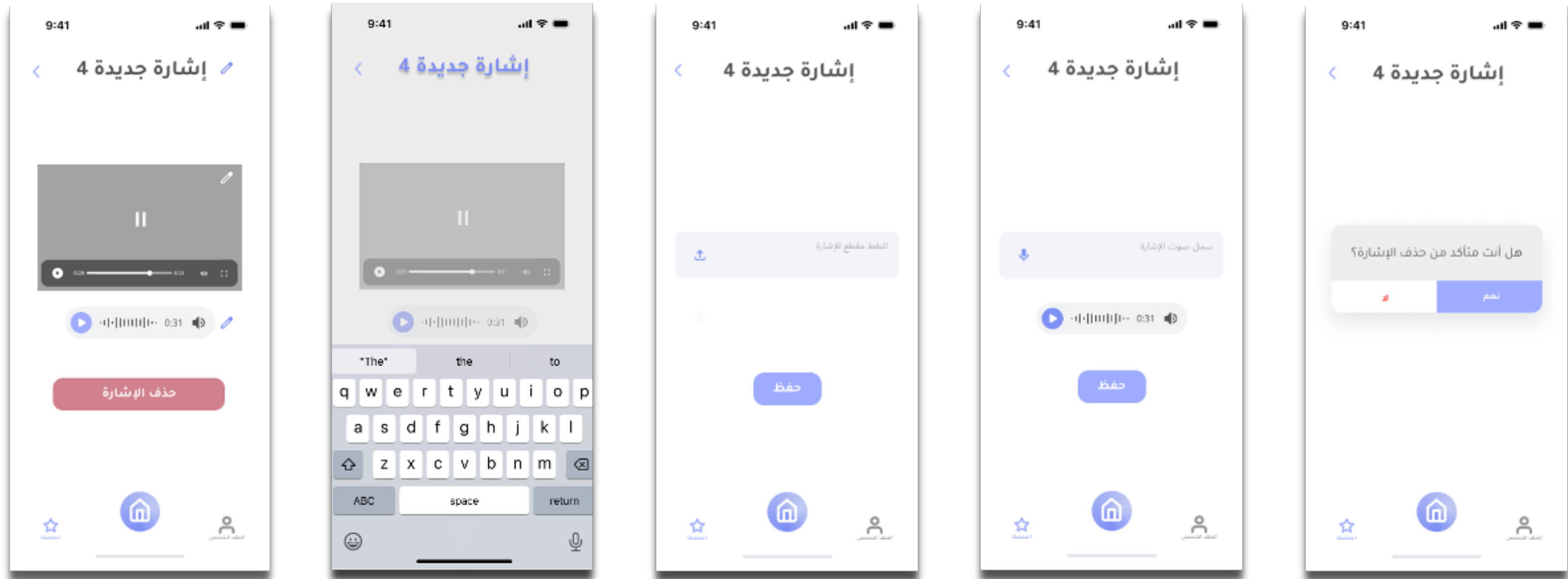
Communication Options for Sign Language Users

USER INTERFACE DESIGN



MANAGING AND ADDING CUSTOM SIGN GESTURES

USER INTERFACE DESIGN



VIEWING, MANAGING AND DELETING CUSTOM SIGN GESTURES

USER INTERFACE DESIGN



Font size adjustment

IMPLEMENTATION

5.1 IMPLEMENTATION

The system was created using the KARSL-502 dataset, which has over 3 million images!

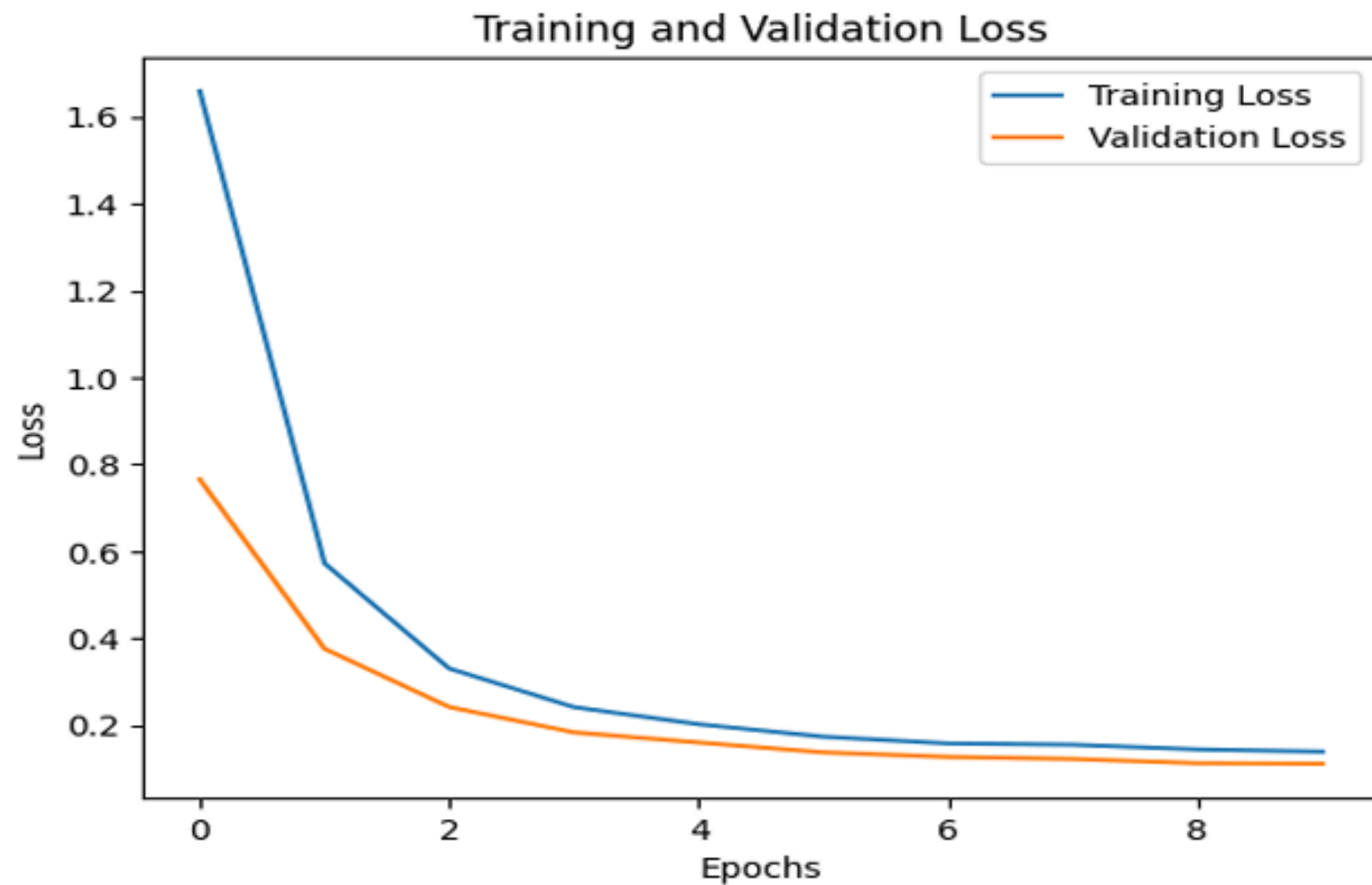


FIGURE 65: TRAINING AND VALIDATION LOSS

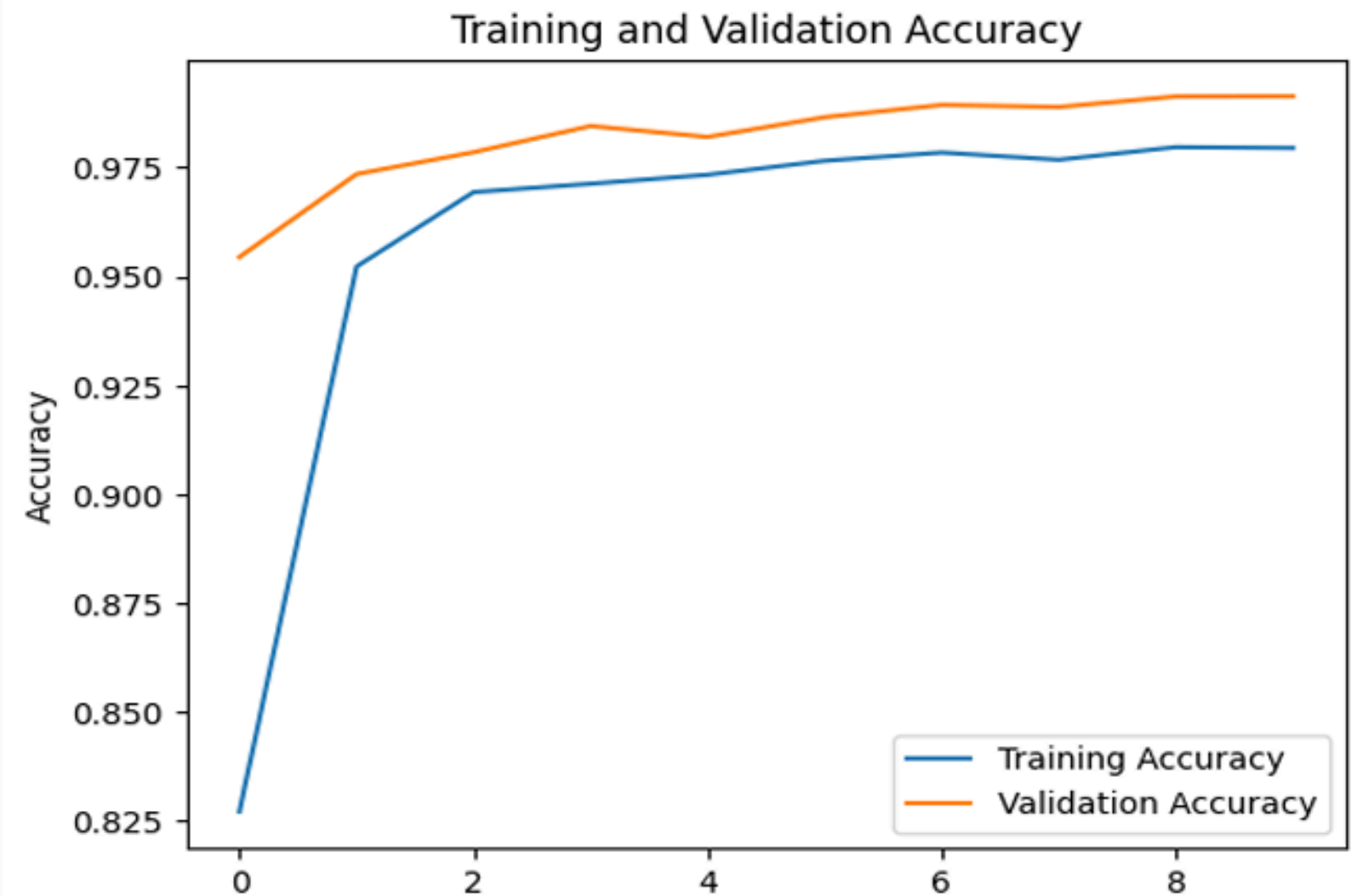


FIGURE 64: TRAINING AND VALIDATION ACCURACY

5.1

IMPLEMENTATION

Classification Report:				
	precision	recall	f1-score	support
0	0.98	0.99	0.99	150
1	0.97	0.98	0.97	150
2	0.97	0.99	0.98	150
3	0.96	0.97	0.96	150
4	0.98	0.98	0.98	150
...				
250	0.97	0.96	0.97	150
300	0.99	0.97	0.98	150
350	0.97	0.99	0.98	150
400	1.00	0.99	1.00	150
450	0.97	0.97	0.97	150
500	1.00	0.99	0.99	150
accuracy			0.98	75300
macro avg	0.98	0.98	0.98	75300
weighted avg	0.98	0.98	0.98	75300

FIGURE 66:CLASSIFICATION REPORT OF THE MODEL TRAINING

5.1

IMPLEMENTATION

Real-Time Gesture Detection and Voice Output

```
model = tf.keras.models.load_model("sign_language_model_final.keras")
labels = ["أحبك" , "شكراً" , "مرحباً" , "لا" , "نعم"]

# Video capture and MediaPipe initialization
cap = cv2.VideoCapture(0)
hands = mp.solutions.hands.Hands()
face = mp.solutions.face_detection.FaceDetection()
```

FIGURE 71: VIDEO CAPTURE AND MEDIAPIPE INITIALIZATION

5.1

IMPLEMENTATION

Our tries on the AI-Driven Avatar using OpenAI **Sora**



5.1

IMPLEMENTATION

Speech-to-Sign Language Avatar System



FIGURE 73: SIGN LANGUAGE AVATAR DISPLAYING A RECOGNIZED GESTURE

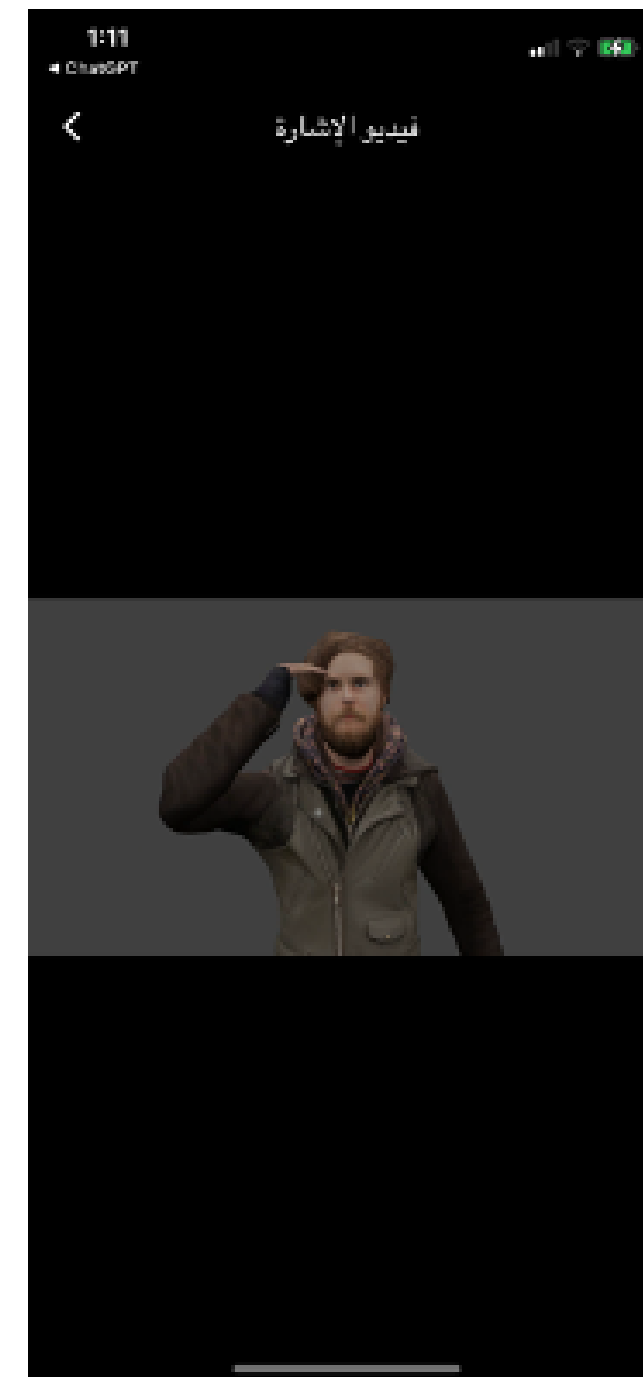
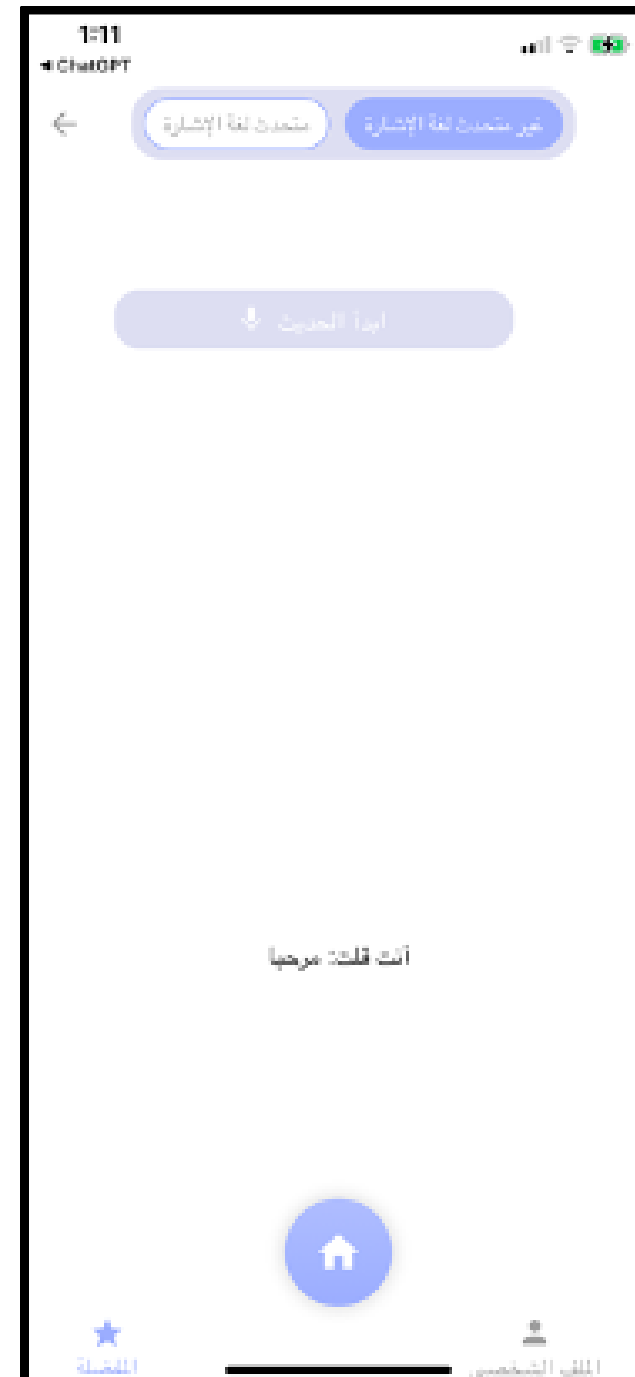
5.2

I/O SCREENS



5.2

I/O SCREENS



TESTING

6.1

TEST PLAN

To ensure reliability & real-world usability, **WASIF** application underwent **five key** testing phases:

Unit Testing

Verified components like gesture classification, TTS conversion, and avatar playback.

Functional Testing

Assessed full workflows: camera input → speech recognition → gesture prediction → avatar → voice feedback.

Integration Testing

Smooth coordination between the FastAPI backend, Flutter frontend, and on-device media modules.

Compatibility Testing


Checked consistent performance and layout across multiple iOS device models.

Acceptance Testing

Involved real users from deaf/mute and hearing communities to validate bidirectional communication success.


6.2

TEST CASES

Test Case	Description	Screenshot	Status
1.1 USER REGISTRATION AND EMAIL VERIFICATION.	The system registers the user with a unique email and password, sends a verification email, and activates the account once the email is verified.		Pass


6.2

TEST CASES

Test Case	Description	Screenshot	Status
1.2 USER LOGIN AND PREFERENCES SAVING.	The system validates the user's login credentials and saves their preferences for future personalized access.		Pass


6.2

TEST CASES

Test Case	Description	Screenshot	Status
1.3 SPEECH RECOGNITION AND DISPLAY.	The system uses AI to recognize the user's speech and displays the corresponding video		Pass


6.2

TEST CASES

Test Case	Description	Screenshot	Status
1.4 GESTURE RECOGNITION AND TRANSLATION.	The system captures the user's gestures via video, translating them into synthesized speech instantly.		Pass


6.2

TEST CASES

Test Case	Description	Screenshot	Status
1.5 FONT SIZE ADJUSTMENT AND DISPLAY CONFIGURATION.	The system allows the user to configure display settings like font size and display mode for improved accessibility.		Pass

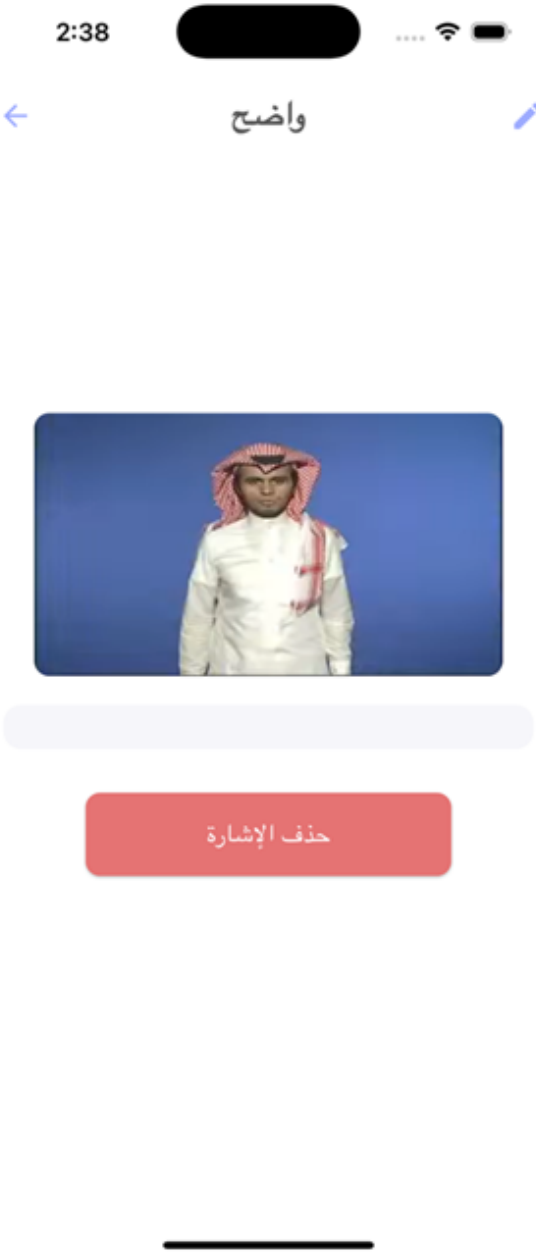
6.2

TEST CASES

Test Case	Description	Screenshot	Status
1.6 SWITCHING BETWEEN MODES.	Users can select their preferred communication method, that is speech or sign language tailored to their needs, this triggers the Update Interface function, adjusting the app's layout for a seamless experience.	 A screenshot of a mobile application interface for ChatGPT. At the top, the status bar shows the time 1:11 and battery level at 59%. Below the status bar, there's a header with a back arrow and the text 'ChatGPT'. The main content area features two large, rounded rectangular buttons: one labeled 'متحدث لغة الإشارة' (Sign Language Speaker) and another labeled 'غير متحدث لغة الإشارة' (Non-Sign Language Speaker). Below these buttons is a large, light blue button with a microphone icon and the text 'أبدأ الحديث' (Start Talking). At the bottom of the screen, there's a navigation bar with three icons: a star labeled 'المفضلة' (Favorites), a house icon, and a person icon labeled 'الملف الشخصي' (Profile). The text 'أنت قلت: مرحبا' (You said: Hello) is visible above the navigation bar.	Pass

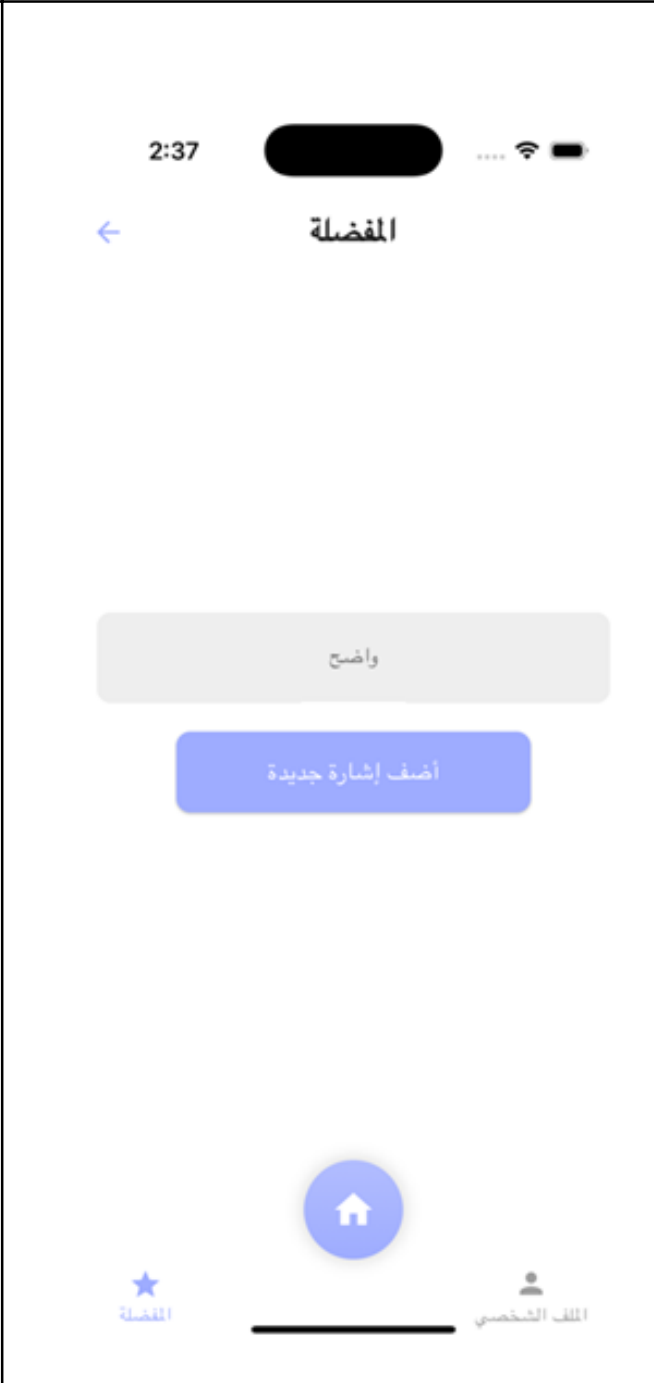
6.2

TEST CASES

Test Case	Description	Screenshot	Status
1.7 MODIFYING CUSTOM SIGN LANGUAGE GESTURES.	Users can create and save custom gestures, to modify a gesture, they can edit and update Gestures, if a gesture is no longer needed, it can be deleted. The Display Gesture feature allows users to see the details of specific gestures for better understanding.		Pass

6.2

TEST CASES

Test Case	Description	Screenshot	Status
1.8 BASIC FUNCTIONALITIE	Users can operate basic features. They can create, edit, or even view custom sign language gestures. The app stores sign data locally and adjusts the interface to display the sign .		Pass

6.3

TEST RESULTS

THE OVERALL TESTING PHASE DEMONSTRATED THAT **WASIF** SYSTEM IS **STABLE**, **ACCURATE**, AND **SUITABLE** FOR REAL-TIME COMMUNICATION IN MOBILE ENVIRONMENTS.

Test Type	Key Findings
Model Testing	Achieved 97.96% accuracy, F1 score of 0.96 on gesture classification model.
Functionality Testing	All system features passed including gesture recognition, voice output, and avatar video rendering.
Integration Testing	Smooth integration between FastAPI, Flutter, camera, and audio components.
Compatibility Testing	The app worked effectively across various devices including iPhone 15 Pro Max and Android 12+.
Acceptance Testing	Real users confirmed the system's effectiveness for communication and accessibility.

CONCLUSION

7.1 EVALUATION

WASIF bridges the communication gap between the deaf/mute and hearing communities through real-time, two-way translation.

It supports gesture-to-speech using MediaPipe and a neural network with 97%+ accuracy, and speech-to-sign using an avatar offline.

Built with Flutter, it runs on iOS and Android. Limitations include lighting sensitivity and word-level avatar output, but future improvements like AI avatars and sentence-level translation are planned.

7.2 FUTURE WORK

The next phase of development will include:



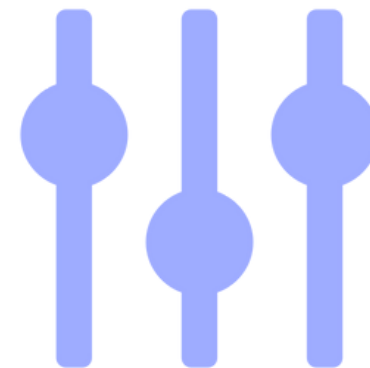
AI-driven animation



**Expanding
the vocabulary**



**saving personalized
signs across devices**



customizable avatars



**Enhancing speech
accuracy for dialects**

With these improvements, WASIF aims to become a scalable, intelligent tool that adapts to the diverse needs of its users.

واصف

جامعة الأميرة
نورة بنت عبدالرحمن
كلية علوم الحاسب والمعلومات



THANK YOU

FOR BEING A PART OF THIS JOURNEY

