Graduation Project
Academic Year 2025-2026
Midyear Short Documentation

Project Title: Real-Time Sign Translator App (**Beyady**)

Supervisors:
Dr. Mayar Ali
Eng. Hisham Ismail

Team Members:

1. Mohamed Khalifa Abbas (AI/ML Specialist)	**2201220**
2. Basmala Mohamed Salah (Mobile Developer/Flutter)	**2102315**
3. Abdallah Aboelhamd Abd ElGaber (Backend Developer/Laravel)	**2202828**
4. Rehab Ahmed Mohamed (Backend Developer/Laravel)	**2201040**
5. Reem Ezzat Ameen (Mobile Developer/Flutter)	**2200099**
6. Salsabel Momen Mahmoud (AI/ML Specialist)	**2201051**
7. Shahd Sherif Sharkawy (AI/ML Specialist)	**2200066**
8. Alaa Mahmoud Mohamed (UI/UX Developer)	**2201033**

Abstract. This documentation presents Beyady, a mobile application designed to provide real-time, bidirectional translation between Egyptian Arabic Sign Language (EASL/ESL) and text/voice, targeting Egypt's deaf and hard-of-hearing community (estimated at over 7.5 million individuals, including varying degrees of hearing impairment). Existing sign language technologies largely focus on non-Arabic variants (e.g., ASL), offer unidirectional translation, depend on specialized hardware or lighting conditions, or lack educational and conversational features, leading to ongoing social isolation, limited education, and employment barriers for EASL users.

The system utilizes MediaPipe for extracting 192-dimensional hand and upper-body landmark features, processed by a hybrid LSTM–GRU model achieving 92% accuracy on dynamic word-level sequences. Text-to-sign output is generated via a 3D animated avatar created in Blender and rendered in Unity. Built with Flutter (frontend) and Laravel (backend with MySQL/Firebase), the application is trained on a custom dataset of 11,338 video clips covering 574 signs in 17 semantic categories.

By integrating real-time translation (SignTranslate), bidirectional chat (SignChat), and structured learning modules (SignLearn) into one accessible platform, Beyady demonstrates a feasible, mobile-optimized solution for low-resource sign languages. The work highlights the potential of AI-driven assistive tools to enhance daily inclusion for the Egyptian deaf community.

1. Background

The domain of assistive technologies for individuals who are deaf or hard-of-hearing has advanced considerably through the integration of artificial intelligence, computer vision, and mobile platforms. These innovations enable real-time sign language recognition and translation, thereby fostering greater social participation and reducing communication barriers. Sign languages are fully-fledged visual-gestural systems with unique grammar and vocabulary, distinct from spoken languages and varying significantly across regions and cultures.

In Egypt, **Egyptian Arabic Sign Language (EASL/ESL)** serves as the primary means of communication for the deaf community. Recent estimates indicate approximately **500,000 to 1.2 million** primary users of Egyptian Sign Language, with broader hearing impairment affecting millions (e.g., around 4% of the population report hearing or speech impairments according to 2023 Ministry of Social Solidarity data, equating to roughly 4–5 million people when including varying degrees of loss). Despite this substantial population, digital tools remain scarce and predominantly support non-Arabic sign languages such as American Sign Language (ASL), leaving a critical gap in localized, accessible solutions.

The primary motivation for this project stems from the persistent exclusion experienced by deaf and hard-of-hearing individuals in Egypt. Daily interactions, educational opportunities, employment prospects, and access to public services are frequently hindered by the absence of effective, affordable communication aids. This contributes to social isolation, reduced self-esteem, and economic disadvantage for users, while limiting empathy and integration within the wider society.

Beneficiaries include:

- **Primary**: Deaf and hard-of-hearing individuals who gain independence in communication and learning.
- **Secondary**: Hearing family members, educators, colleagues, and service providers who can interact more naturally and inclusively.
- **Broader society**: Enhanced empathy, diversity awareness, and compliance with principles of universal accessibility and human rights.

The project employs the following **main techniques**:

- Computer vision via **MediaPipe** for precise extraction of hand, pose, and facial landmarks (yielding 192-dimensional feature vectors).
- Deep learning models hybrid **LSTM-GRU** architecture for temporal sequence modeling in dynamic signs (92% accuracy).
- 3D avatar animation created in **Blender** and rendered in real-time using **Unity**, integrated into the mobile interface via **flutter_unity_widget**.
- Cross-platform mobile development with **Flutter**, secure backend services via **Laravel** (with MySQL and Firebase), and Python-based AI pipelines trained on Google Colab.

The **main application** is a comprehensive mobile app named **Beyady** (meaning "with my hand" in Arabic), designed to deliver an inclusive ecosystem: bidirectional real-time translation (SignTranslate), conversational chat between deaf and hearing users (SignChat) and structured sign language lessons for hearing learners (SignLearn). By combining translation, education, and social connectivity in a single, user-friendly platform, Beyady aims to transform everyday communication and promote long-term societal inclusion for Egypt's deaf community.

2. Problem Definition

Deaf and hard-of-hearing individuals in Egypt face persistent and multifaceted communication challenges that significantly restrict their participation in everyday life, education, employment, and social interactions. Egyptian Arabic Sign Language (EASL/ESL), the natural language of the deaf community in Egypt, remains largely unsupported by modern digital tools, leaving an estimated 500,000–1.2 million primary sign language users and several million people with varying degrees of hearing loss without accessible, reliable, and affordable means of real-time communication with hearing individuals.

The core problems can be summarized as follows:

- **Lack of localized real-time translation tools**: Most commercially available or research-based sign language applications target American Sign Language (ASL) or other non-Arabic variants, offering no or minimal support for the distinct grammar, vocabulary, and cultural nuances of Egyptian Sign Language.
- **Predominantly unidirectional functionality**: Existing solutions typically provide only sign-to-text or text-to-sign conversion, rarely supporting true bidirectional interaction in real time, which is essential for natural conversations between deaf and hearing users.
- **Technical and environmental limitations**: Many systems suffer from sensitivity to lighting conditions, complex backgrounds, reliance on external hardware (e.g., specialized sensors or high-end GPUs), or restricted vocabularies (often 15–262 isolated signs), making them impractical for daily mobile use.
- **Absence of integrated educational components**: Few applications combine translation with structured learning modules, literacy support, or interactive practice tools, leaving both deaf individuals (who may need reading/writing reinforcement) and hearing individuals (who wish to learn sign language) without comprehensive resources.
- **Limited accessibility and inclusivity features**: Current tools frequently lack offline capabilities, privacy-focused processing, intuitive 3D avatar representations of signs, or support for conversational AI interaction, further marginalizing users in resource-constrained settings.

These barriers result in profound consequences: increased social isolation, reduced access to quality education, diminished employment opportunities, dependency on costly human interpreters (when available), and overall exclusion from full societal participation. The absence of a scalable, mobile-first, culturally relevant solution perpetuates inequality and hinders progress toward universal accessibility and digital inclusion in Egypt.

This project directly addresses these gaps by developing **Beyady**, a mobile application that delivers bidirectional real-time translation, integrated learning modules, conversational chat functionality, and AI-assisted interaction — all optimized for Egyptian Arabic Sign Language and designed to operate effectively on standard smartphones without external hardware dependencies.

3. Related Works

- The existing similar implementations to the idea of our project:

  Several sign language recognition systems have been developed using various approaches. We reviewed **six** representative systems:

  3.1 Eqab & Shanableh (2017) developed an Android app for Arabic Sign Language using Leap Motion Controller, achieving 98.7% accuracy on 15 isolated words with real-time bilateral translation. However, the system requires external hardware and has extremely limited vocabulary.

  3.2 Alkahtani et al. (2025) achieved 97% accuracy using VideoMAE transformer architecture on 587 Saudi Arabic Sign Language videos. While achieving state-of-the-art results, the system requires powerful GPU hardware and provides no user-facing application.

  3.3 Purwanto & Aryanto (2024) created a mobile Indonesian Sign Language app using YOLOv8n, achieving 98.6% mAP@50 on 38 gestures. The system is mobile-optimized but limited to static gestures and one-way translation.

  3.4 Navendu & Sahula (2024) used MediaPipe + LSTM-GRU for Indian Sign Language, achieving 89.5% accuracy on 262 words across 5,334 videos. Despite the largest dataset, the system is desktop-focused with no practical application.

  3.5 Al-Tarjuman (2024) offers an Arabic Sign Language mobile app using YOLOv8 with 98% mAP@50 on 38 gestures. While practical and mobile-ready, it supports only one-way translation without comprehensive educational features.

  3.6 Hand Talk (2012-Present) is a widely-used commercial app translating text/audio to American Sign Language (ASL), Brazilian Sign Language (Libras), and British Sign Language (BSL) using 3D avatars (Hugo and Maya). The app uses AI trained on 38,000+ words and has over 3 million users globally. Studies report translation accuracies of 83.3% (automatic evaluation) and 86.1% (human evaluation). While featuring educational content ("Hands Up" learning path) and 3D avatar translation, the system does not support Arabic Sign Language, relies on text/voice-to-sign translation only (no sign recognition), and exhibits translation errors with complex sentences, numbers, and dates.

- **Common Limitations**: Small vocabularies (15-262 gestures for Arabic systems), focus on non-Egyptian dialects or non-Arabic languages, recognition-only without comprehensive features (except Hand Talk which is translation-only), one-way translation or limited bidirectional capability, and limited mobile accessibility (except recent systems).

The main differences between them and our project:

Table 1: Comparative differences between existing systems and Beyady

Aspect	Existing Systems	Our Project
Language	Non-Arabic or Gulf dialects (UAE, Saudi)	First Egyptian Sign Language system
Dataset	15-6, 38,000 samples	Community-sourced Egyptian dataset
Vocabulary	15-262 gestures	Comprehensive (letters, numbers, words, phrases)
Features	Recognition/translation only OR translation-only (Hand Talk)	**3 integrated features:** SignTranslate, SignLearn, SignChat
Translation	80% one-way; Hand Talk: text→sign only (no sign recognition)	**Bilateral** with real-time 3D avatar (Blender + Unity)
Gestures	90% static/isolated only; Hand Talk: no sign recognition	Static + dynamic (MediaPipe + LSTM/GRU)
Platform	External hardware, GPU required, or mobile-only translation	**Mobile-only**, camera-based, **no external hardware**
Education	Basic dictionary or learning path (Hand Talk)	Structured curriculum validated by Faculty of Arts experts
Accessibility	Free some systems but limited features or not Arabic	Free, mobile-optimized, **Egyptian-specific**, accessible to all
Community	Commercial (Hand Talk)-academic-developed	Co-designed with Egyptian deaf community + special education experts

Key Differentiators:

1. **Egyptian Focus:** First system specifically for Egyptian Sign Language (7.5M users), not Gulf dialects, ASL, or other languages
2. **Complete Ecosystem:** Three integrated features (SignTranslate, SignLearn, SignChat) vs. single-purpose systems
3. **True Bilateral:** Real-time 3D avatar for Text→Sign **AND** sign recognition for Sign→Text (Hand Talk lacks sign recognition)
4. **Mobile Accessible:** Optimized for smartphones without external hardware or GPU
5. **Community-Driven:** Built with Egyptian deaf community and expert validation

In summary, existing systems demonstrate technical feasibility but fail to address Egyptian deaf community needs comprehensively. Hand Talk, while successful commercially for ASL/Libras with 3D avatars, does not support Arabic. Arabic systems lack comprehensive features or Egyptian dialect support. Our project uniquely combines Egyptian dialect, community partnership, true bilateral translation with 3D avatar, complete educational platform with real-time communication—representing a fundamental advancement beyond current solutions.

5. Project Specifications

This section outlines the detailed specifications of the **Beyady** mobile application, an application development project focused on real-time Egyptian Arabic Sign Language (EASL/ESL) translation and inclusive learning.

5.1 System Architecture

The system follows a **client-server architecture** optimized for mobile performance and scalability:

- **Frontend (Mobile Client)**: Developed using **Flutter** (Dart) for cross-platform support (Android and iOS). Handles user interface, camera input for sign capture, microphone/speech input, and real-time rendering of 3D avatars via **flutter_unity_widget** integration.
- **Backend Server**: Built with **Laravel** (PHP framework), using **MySQL** as the relational database for user data, chat history, learning progress, and animation metadata. Firebase provides real-time push notifications and supplementary chat synchronization.
- **AI Processing Layer**: Python-based modules (running on server or edge inference) utilize **MediaPipe** for landmark extraction, **OpenCV** for preprocessing and a hybrid **LSTM-GRU** model for dynamic sign sequence recognition. Models are trained on Google Colab .
- **External Integrations**: **Unity** + **Blender** for 3D avatar animation generation; pretrained speech-to-text (STT) and text-to-speech (TTS) services for voice handling.

Data flows from client → backend API → AI inference (if needed) → response (text, voice, or animated sign).

5.2 Stakeholders

- **Primary Users**: Deaf and hard-of-hearing individuals in Egypt – core beneficiaries who use translation for communication.
- **Secondary Users**: Hearing individuals (family members, educators, colleagues, service providers) – utilize SignTranslate, SignChat, and SignLearn to interact with and learn from deaf users.
- **Content Experts**: Faculty from the Special Education Department (Faculty of Arts) for ensuring sign language accuracy, pedagogical quality of learning modules, and cultural relevance of EASL content.
- **Broader Society**: Educational institutions, employers, and the general public who benefit from increased inclusivity and reduced communication barriers.

5.3Functional Requirements

The application must support the following core functionalities:

1. User authentication and profile management (email/OTP, Google Sign-In, secure sessions via JWT).
2. Real-time **Sign-to-Text/Voice** translation: Capture sign language via device camera → process landmarks → recognize via LSTM-GRU → output text or synthesized speech.
3. Real-time **Text/Voice-to-Sign** translation: Input text or voice → convert to sequence of signs → render using 3D avatar animation.
4. **SignChat**: Bidirectional real-time chat between deaf and hearing users (sign input → text display for hearing; text input → avatar sign display for deaf).
5. **SignLearn**: Progressive interactive lessons for hearing users (levels from letters/numbers to phrases), with 3D avatar demonstrations and quizzes.
6. User history, progress tracking, and offline caching for basic learning modules.

5.4Non-Functional Requirements

- **Performance**: Real-time processing latency < 1 second on mid-range smartphones; model inference optimized for mobile (edge or lightweight server calls).
- **Usability & Accessibility**: Intuitive, high-contrast UI; support for large text, dark mode, voice feedback; culturally sensitive avatar gestures.
- **Security**: JWT-based authentication, HTTPS encryption, data minimization, no unnecessary storage of camera feeds.
- **Reliability**: ≥ 92% recognition accuracy (validated on custom dataset); graceful degradation in poor lighting/network conditions.
- **Scalability**: Backend capable of handling up to 10,000 concurrent users; cloud-ready deployment.
- **Maintainability**: Modular code structure, Git version control, comprehensive API documentation.
- **Portability**: Android primary (iOS future).

Use Case Diagram

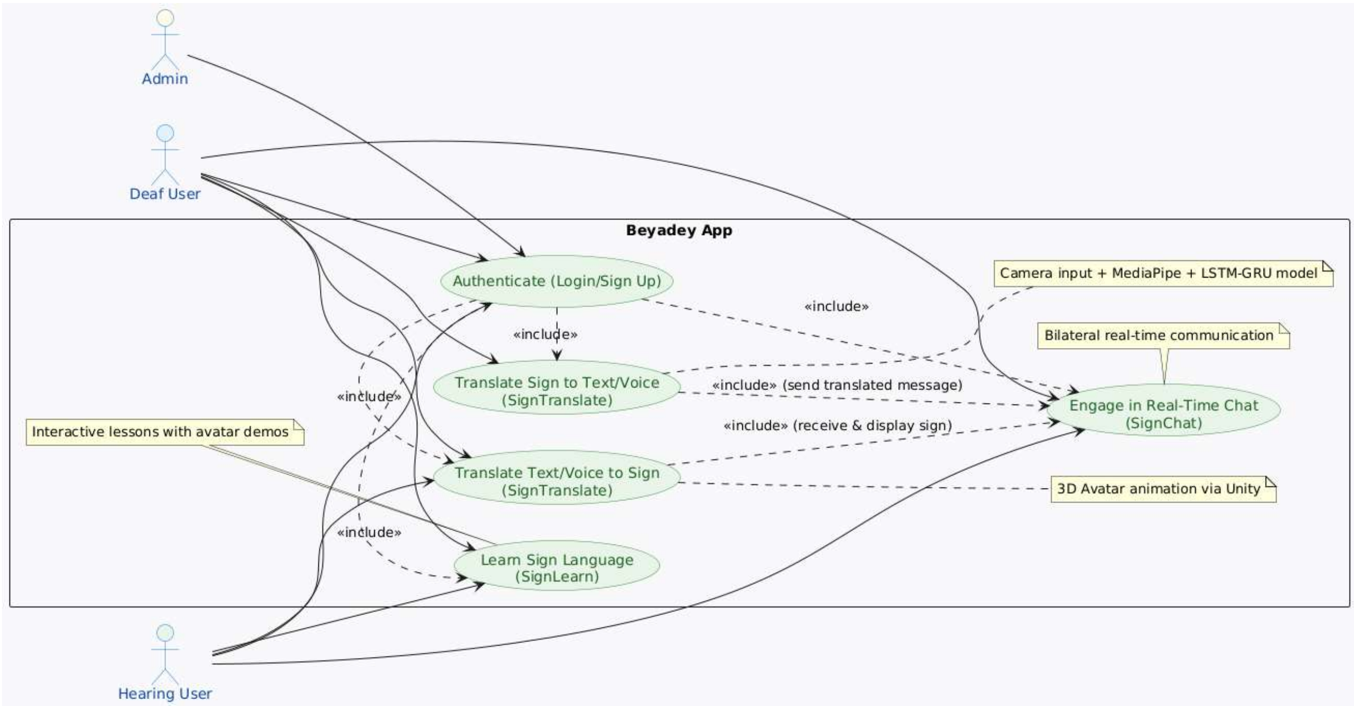

Fig. 1: Use Case Diagram of the Beyady application, illustrating primary actors (Deaf user, Hearing user, Admin) and core functionalities (SignTranslate, SignChat, SignLearn, authentication).

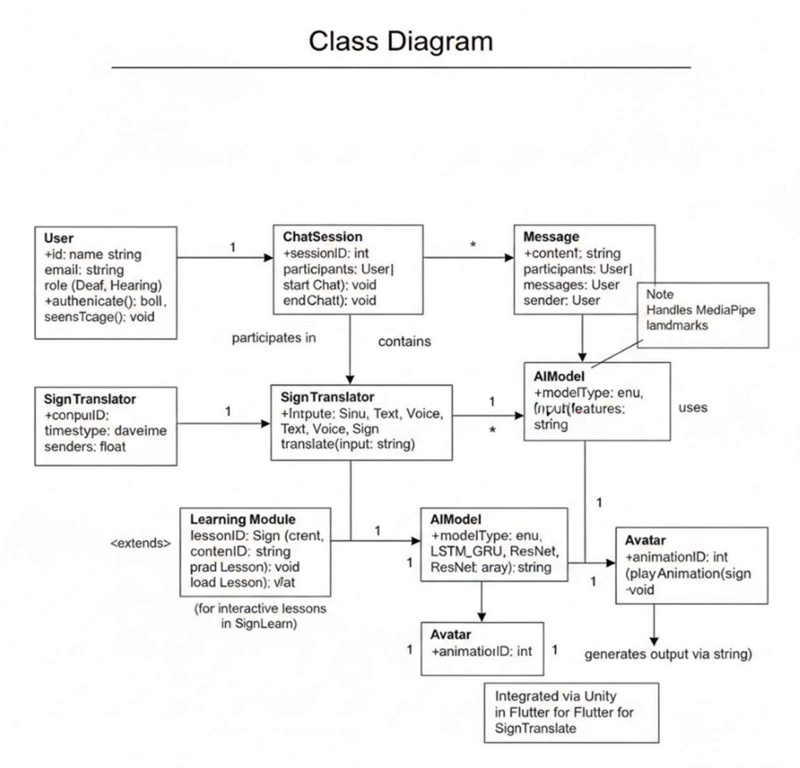

Fig. 2: Class Diagram of the Beyady application The diagram illustrates the main classes and their relationships, including User management, ChatSession for SignChat, SignTranslator for real-time bidirectional translation, Learning Module for SignLearn, AIModel (LSTM-GRU/ResNet variants), and Avatar for 3D sign rendering integrated via Unity in Flutter.

Sequence Diagram

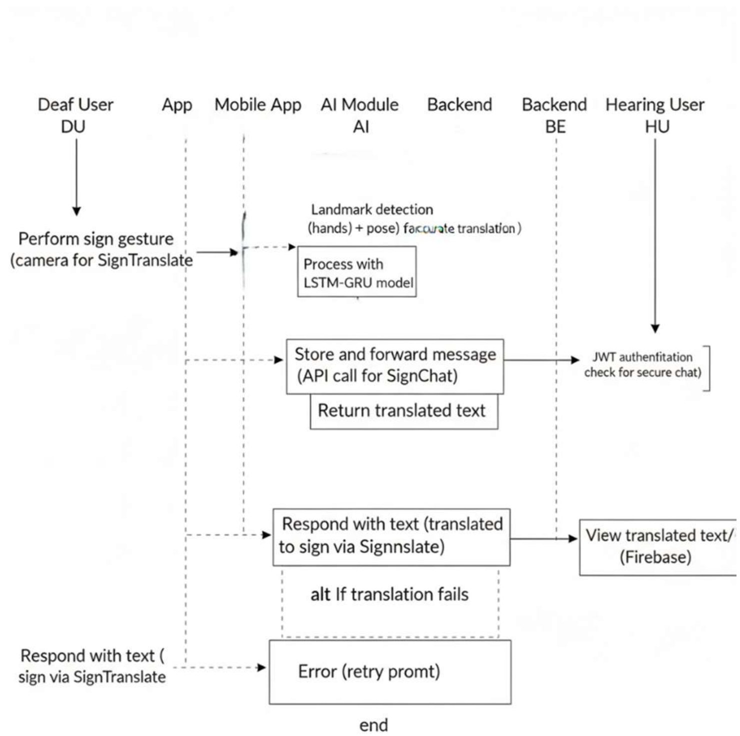

Fig. 3: Sequence Diagram of the Beyady application The diagram illustrates the real-time bidirectional translation and chat flow (SignChat / SignTranslate), showing interactions between the Deaf User, Mobile App, AI Module (MediaPipe landmark detection + LSTM-GRU processing), Backend (JWT authentication + API forwarding), and Hearing User, including message handling, text/sign response, and error recovery path (retry prompt on translation failure).

ERD Diagram

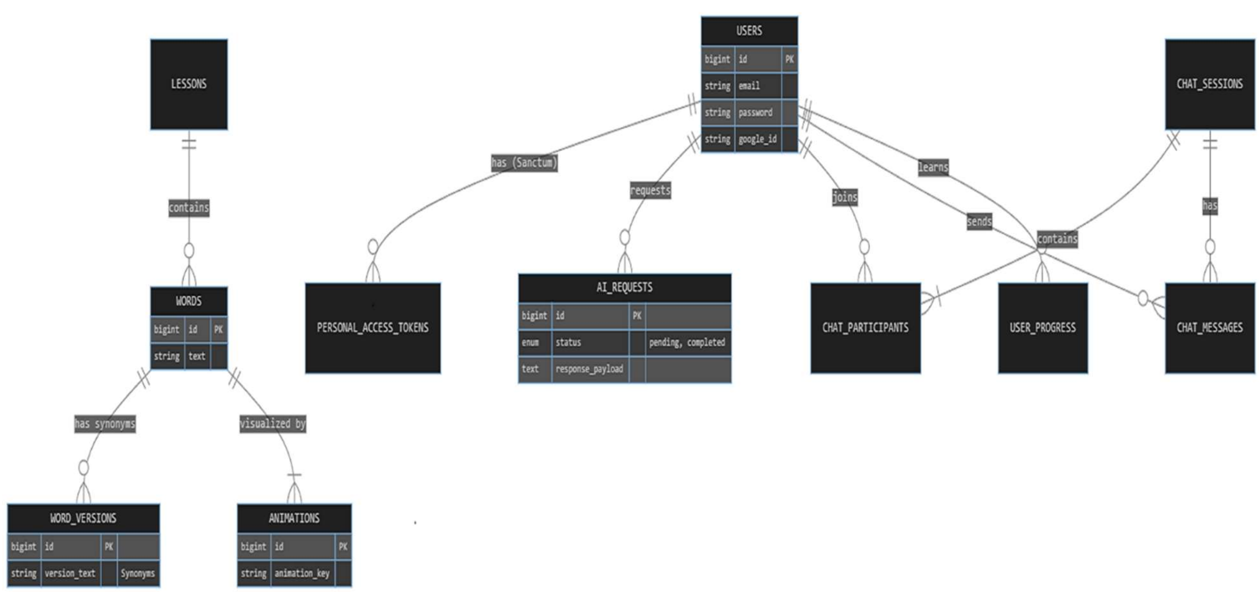

Fig. 4: ERD of the Beyady database schema The diagram illustrates the main entities (USERS, CHAT_SESSIONS, CHAT_MESSAGES, LESSONS, WORDS, WORD_VERSIONS, ANIMATIONS, AI_REQUESTS, PERSONAL_ACCESS_TOKENS, USER_PROGRESS, CHAT_PARTICIPANTS) and their relationships, including foreign keys, cardinalities, and attributes for authentication, chat history, learning progress, sign animations, and AI processing.

5.5 AI Components

The **Beyady** application incorporates advanced AI-driven components for accurate and efficient Egyptian Arabic Sign Language (EASL/ESL) recognition and generation. These components form the core intelligence of the real-time translation pipeline and are optimized for mobile deployment with minimal latency.

Feature List

The primary input features extracted from video frames or live camera feed include:

- **Hand landmarks** (MediaPipe Hands): 21 landmarks per hand × 2 hands = 42 points Each point provides x, y, z coordinates → 126 features
- **Upper-body pose landmarks** (MediaPipe Pose): 22 relevant upper-body keypoints (shoulders, elbows, wrists, hips, etc.) Each point provides x, y, z coordinates → 66 features
- **Total concatenated feature vector**: 192 dimensions per frame (126 hand + 66 pose)

These features capture both fine-grained finger movements (critical for letters and static signs) and contextual upper-body posture (essential for dynamic verbs, adjectives, and spatial grammar in EASL).

Feature Selection Process

Feature selection was performed to reduce noise, improve model generalization, and optimize inference speed on mobile devices:

1. **Initial extraction** — All 192 landmarks were collected using MediaPipe's holistic solution.
2. **Correlation analysis** — Pearson correlation matrix was computed across features to identify highly correlated landmarks (threshold > 0.85); redundant features (e.g., closely overlapping wrist/elbow points) were removed or averaged.
3. **Domain knowledge filtering** — Features with low relevance to EASL (e.g., lower-body landmarks irrelevant to most signs) were excluded based on consultation with Special Education experts.
4. **Final selection** — Retained 192-dimensional sequence of normalized (min-max scaled) coordinates per frame, augmented with temporal differences (Δx, Δy, Δz) between consecutive frames to enhance dynamic gesture modeling.

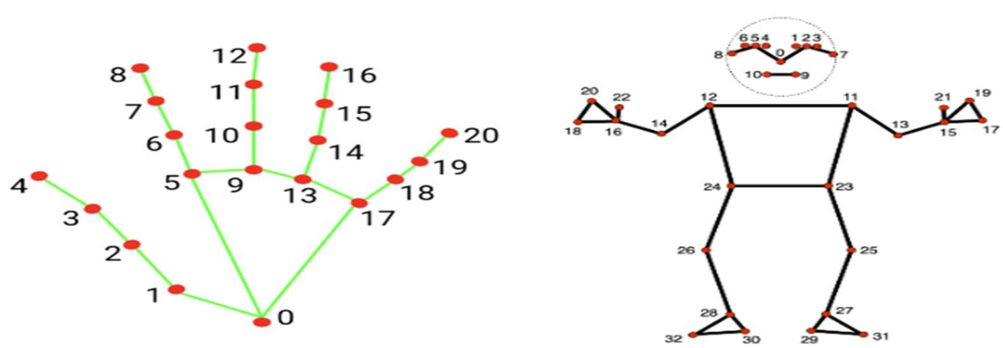

Fig. 5: MediaPipe landmark topology for hand (left) and pose (right) Illustrates the 21 hand keypoints (per hand) and 33 body keypoints used to extract 192-dimensional features for sign recognition.

Expected Training Process

The training pipeline follows industry-standard deep learning practices tailored for sequence classification:

1. **Data preprocessing**
    - Video frame extraction at 30 fps
    - Landmark normalization (relative to noise center)
    - Sequence padding/truncation to fixed length (max 60 frames)
    - Data augmentation Gaussian Noise , Scaling , Time Stretching and Frame Dropping .
2. **Model architecture**
    - **Dynamic sequences** (words, phrases): Hybrid LSTM-GRU model
        - Input: sequence of 192-dim feature vectors
        - Bidirectional LSTM (128 units) → GRU (128 units) → Dense layer → softmax
        - Dropout (0.3) and batch normalization applied
3. **Training setup**
    - Framework: Tensorflow
    - Optimizer: Adam (learning rate = 0.001)
    - Loss: Categorical Cross-Entropy
    - Scheduler: ReduceLROnPlateau (patience 5, factor 0.5)
    - Batch size: 32
    - Epochs: 50 (early stopping on validation loss)
    - Hardware: Google Colab Pro (T4/A100 GPU)
    - Train/validation/test split: 70/15/15 stratified by class
4. **Evaluation metrics**
    - Accuracy, Precision, Recall, F1-score (macro-averaged)
    - Confusion matrix analysis for class imbalance handling

Dataset Used

- **Primary dataset**: Custom-collected Egyptian Arabic Sign Language (EASL) corpus
    - **Size**: 11,338 short video clips
    - **Vocabulary**: 574 unique words/signs
    - **Categories**: 17 semantic groups
    - **Collection method**: Recorded in controlled studio setting with multiple native signers (varying age, gender, signing speed) and different camera angle using high-resolution smartphone cameras

This AI pipeline achieves **92%** validation accuracy on dynamic word-level sequences (LSTM-GRU), providing a robust foundation for real-time, mobile-optimized translation in the Beyady application.

Model Architecture

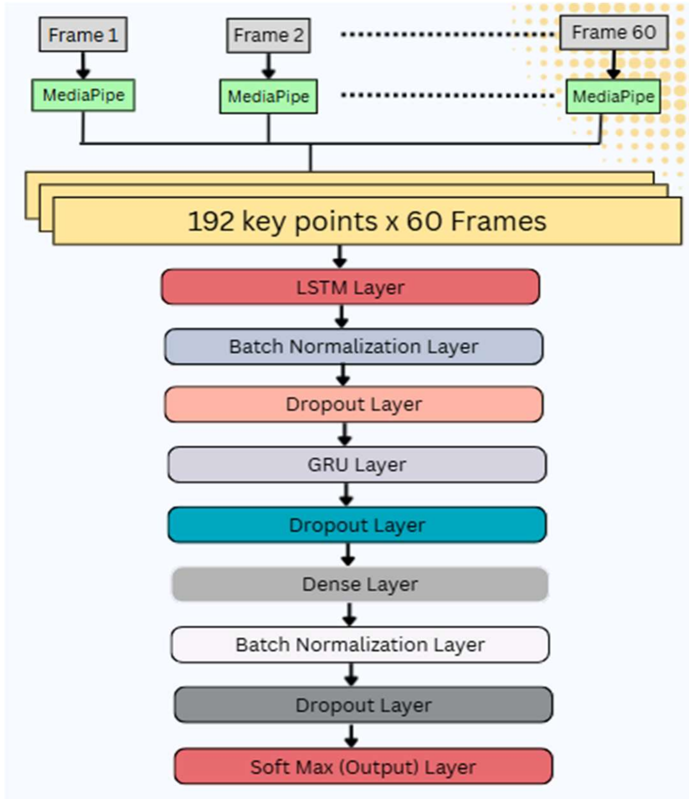

Fig. 6: Hybrid LSTM-GRU model architecture Depicts input sequence (192 keypoints × 60 frames) processed through LSTM, BatchNorm, Dropout, GRU, Dense, and Softmax layers for word-level sign recognition.

Table 2: Semantic categories covered in the EASL dataset

1	letters	**7**	Greetings	**13**	Vegetables
2	numbers	**8**	Animals	**14**	Clothing
3	verbs	**9**	Family	**15**	Colors
4	adjectives	**10**	Food	**16**	Jobs
5	pronouns	**11**	Drinks	**17**	Transportations
6	Time expressions	**12**	Fruits		

6. Work Plan

Table 3: Project Work Plan and Current Status

Task	Task Title	Description	Task Status
1	UI/UX Design & Implementation	Completed core UI screens including Authentication pages, Home page, and Text-to-Sign page, with consistent design and user flow.	Completed
2	Backend Setup & Database Design	Designed and implemented database schema and APIs; connected Flutter app with backend services for authentication and core features.	Completed
3	System Analysis & Architecture	Defined system requirements, overall architecture, and data flow between frontend, backend, AI models, and Unity avatar.	Completed
4	Dataset Collection & Preparation	Collected a large custom dataset for Egyptian Arabic Sign Language; dataset is usable but still expandable for higher accuracy.	In Progress
5	AI Model Development	AI team is actively training and improving sign recognition models; models are functional but not fully finalized yet.	Planned for next phase
6	Text to Sign Feature	Implemented initial Text-to-Sign translation pipeline; feature works smoothly and will be enhanced with more signs and animations.	Completed with improvements ongoing
7	Sign to Text Feature	Feature depends on completion of AI sign recognition model; implementation will start after model finalization.	Planned for next phase
8	Avatar & Animation Development	Created initial 3D avatar and basic animations; currently working on adding more sign animations using Blender and Unity.	Expected by the end of semester
9	Frontend Feature Integration	Integrated UI screens with backend APIs and Unity components to ensure smooth interaction between modules.	Completed
10	Sign Learn Feature	Feature not started yet; planned for the next phase to support structured sign language learning.	Expected during the semester
11	Integration & Testing	Full system integration and testing (unit & integration) to be completed after finishing AI models and animations.	Expected by the end of semester
12	Documentation & Finalization	Documentation will be finalized progressively during the second term alongside feature completion.	Expected by the end of semester

7. References

- Alkahtani, B. N., Alsraisri, N., & Almalki, A. J. (2025). *Achieving state-of-the-art accuracy in Arabic sign language recognition with VideoMAE and data augmentation*. Journal of Disability Research, 4, 1–11.

- Eqab, A., & Shanableh, T. (2017). *Android mobile app for real-time bilateral Arabic sign language translation using leap motion controller*. In 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA) (pp. 1–5). IEEE.

- Navendu, K., & Sahula, V. (2024). *Word level sign language recognition using MediaPipe and LSTM-GRU network*. In 2024 International Seminar on Intelligent Technology and Its Applications (iSeS). IEEE.

- Purwanto, A., & Aryanto, J. (2024). *Development of an Indonesian sign language translation application with hand gesture detection*. JSRET (Journal of Scientific, Research, Education, and Technology), 4(3), 2211–2228.

- Rahmawati, R., & Suryani, I. (2023). *Role of Hand Talk translator application for sign language translation*. Journal of Language and Literature Studies, 3(2), 78–89.

- Al-Tarjuman Project. (2024). *Arabic sign language translation application using YOLOv8*.