UNIVERSITY OF PLYMOUTH

PUSL3190 Computing Individual Project

Project Proposal

Speech-to-Sign Language Translation Mobile Application

Supervisor: Mr. Saravanapavan Nasiketha

Name: **WGHM Fernando**
Plymouth Index Number: **10898762**
Degree Program: **BSc(Hons) in Computer Science**

Table of Contents

Introduction....4

Chapter 1: Problem Statement....5

Chapter 2: Project Description....5

Chapter 3: Research Gap....6

Chapter 4: Requirements Analysis....7

- 4.1 Functional Requirements....7
  - 4.1.1 Speech Recognition Module....7
  - 4.1.2 Translation Processing....7
  - 4.1.3 Animation Output....7
  - 4.1.4 User Interface....7
- 4.2 Non-Functional Requirements....7
  - 4.2.1 Performance....7
  - 4.2.2 Reliability....7
  - 4.2.3 Security....8
  - 4.2.4 Usability....8
- 4.3 Technical Requirements....8
  - 4.3.1 Device Requirements....8
  - 4.3.2 Development Requirements....8
- 4.4 Constraints....8
  - 4.4.1 Technical Constraints....8
  - 4.4.2 Environmental Constraints....8

Chapter 5: Finance....9

- 5.1 Development Tools and Software....9
  - 5.1.1 Cloud Services (Monthly)....9
- 5.2 Testing Requirements....9
  - 5.2.1 Device Testing....9
- 5.3 API and Service Costs (Monthly)....9
  - 5.3.1 Development APIs....9
  - 5.3.2 Storage Services....9
- 5.4 Operational Costs (Monthly)....10
  - 5.4.1 Internet and Utilities....10
  - 5.4.2 Documentation....10

5.5 Cost Optimization Strategies........10
5.5.1 Resource Management........10
5.5.2 Development Efficiency........10
5.6 Detailed Cost Breakdown........10
5.6.1 Development Tools Cost (One-Time)........10
5.6.2 Monthly Service Costs........10
5.7 Financial Plan Summary........10
Chapter 6: External Organizations........10
6.1 Potential Community Partnerships........10
6.1.1 Sri Lanka Central Federation of the Deaf........11
6.1.2 Sri Lanka School for the Deaf........11
Bibliography........12

Introduction

Every day, over 330,000 Sri Lankans Deaf or hard of hearing face obstacles that few of us ever even think about. A simple exchange—a meal order, a doctor's visit, or a course of study—is an obstacle to be overcome in a world not yet adequately equipped to accommodate their needs. The underlying issue is not one of hearing but of access and inclusion.

There is an urgent shortage of certified sign language interpreters in Sri Lanka today, and there are only six of them who are employed to serve the whole country. While foreign apps give support for American or British Sign Language, our own Sri Lankan Sign Language (SLSL) with its rich cultural background and regional variations has very little support online.

There is a silver lining, though. As of early 2025, Sri Lanka boasts more than 29 million active mobile connections, ahead of its own population. The wholesale adoption of smartphones is a rare opportunity to leverage technology for inclusion.

Welcome to Signosi: A revolutionary Android app capable of converting spoken Sinhala into real-time Sri Lankan Sign Language animations. With the application of advanced speech recognition and thoughtfully designed animations that respect the intricacies of SLSL, Signosi is poised to empower the Deaf community to facilitate simpler interactions in their daily life.

This proposal outlines our vision, the technical challenges ahead, and how we aim to make communication accessible to all Sri Lankans. With Signosi, we're not merely building an app; we're building a more inclusive society where each voice is heard and understood.

Chapter 1: Problem Statement

The deaf and hard-of-hearing community in Sri Lanka encounters enormous communication barriers due to an acute shortage of interpreters for the Sinhala sign language, with only a handful of certified interpreters available across the country. The humanitarian crisis in this regard prevents day-to-day communication, education, and work opportunities from being accessible to many persons in the Deaf community

Chapter 2: Project Description

The Speech-to-Sign Language Translation Application is a novel effort towards breaking down communications barriers between Deaf and non-Deaf communities in Sri Lanka. Compared with global solutions that mainly target either British Sign Language (BSL) or American Sign Language (ASL), this application is specifically tailored towards Sri Lankan Sign Language (SLSL), hence holding cultural and linguistic relevance.

At its core is a state-of-the-art speak platform optimized both in Tamil as well as in Sinhala that can decode regional forms. Along with noise cancellation, it makes conversation seamless in urban areas as much as in rural areas.

The translation algorithm extends beyond literal word-for-word interpretation by looking at context at a sentence level as well as considering culture. This helps in avoiding misunderstanding and ensures that correctly interpreted SLSL is communicated.

To make interpretation in signs seamless, the application employs 2D animated avatars in terms of SLSL. The animations consider key gestures, facial expressions, and body language to enhance conversations in a more expressive and natural manner. The animations are optimized with mid-range smartphone users in mind, with a design that does not waste resources.

The user interface is simple and accessible, with both Tamil as well as Sinhala supported. The high contrast display functionality, adjustable text sizes, as well as quick buttons on frequently accessed phrases, makes it accessible regardless of age as well as ability.

Security and confidentiality are a high priority. All data is locally encrypted on disk with a mirrored copy in a customizable configuration in the cloud. There is minimal collection with complete users' confidentiality, making it suitable in highly confidential contexts as in healthcare as in lawyer client communications.

This app also features a context-based, location-based, historically context-based suggestion feature that makes frequently accessed words in context available. Medical words are preferred in a hospital, as educational words are preferred in schools.

Developed with ongoing consultation from the Deaf community in Sri Lanka, regular testing is conducted on the application to enhance translation quality, animation quality, and overall usability. Upcoming updates will add regional dialect coverage, compatibility with wearable devices, and domain-specific words in areas such as medicine and law.

Technically, the application is developed on Flutter with cross platforming, TensorFlow Lite with offline speak, local database with SQLite. An order to display signs smoothly with a small amount of memory.

By prioritizing cutting-edge technology, accessibility , as well as relevance, Speech-to-Sign Language Translation Application will be a key driver in improving inclusion as well as improving communications in Deaf communities in Sri Lanka.

Chapter 3: Research Gap

A comprehensive review of research that has already been conducted in Sinhala speech-to-sign language translation indicates glaring loopholes in the technology that currently exists. While some attempts have been made, viable and user-friendly solutions for everyday application are mostly unexplored. Karunarathna et al. (2022), in their "Sinhala to Sri Lankan Sign Language Machine Translation System" paper presented at the 21st International Conference on Advances in ICT for Emerging Regions (ICTer), noted that existing models do not support real-time processing and proper grammatical translation of Sinhala to Sri Lankan Sign Language (SLSL).

Their research identifies significant structural divergences of Sinhala syntax from SLSL, making direct translation ineffective with resulting inaccurate and misleading signs.

Bandara et al. (2019), in their paper "Speech Recognition for Sinhala Language: A Review" in the International Journal of Computer Applications, thoroughly address Sinhala speech recognition challenges that are specific to the language. Key points are handling unique phonetic features, separating close phonemes, and accommodating regional pronunciation variations. These typical challenges have a direct bearing on the performance and reliability of any Sinhala-to-sign language system with speech input.

A significant work by Dias et al. (2020), titled "Real-time Sinhala Sign Language Translation System using Image Processing" and presented at the International Conference on Image Processing and Robotics, is targeted towards image-to-text translation. But they refer to considerable shortcomings in current datasets used for SLSL, which they note are inadequate for successful machine learning training.

Additionally, research "Mobile Assistive Technologies for the Deaf Community in Sri Lanka" by Wijayaratne and Perera (2021) published in the Journal of ICT for Development also suggests that culturally and language-specific mobile solutions are needed. They share that there is not currently available the mobile applications to represent SLSL's distinctive linguistic and cultural features and as a result there is low use among Sri Lanka's Deaf community.

From the above defined work, someresearch gaps emerge:

1. Insufficiency of end-to-end Sinhala speech recognition systems specifically tailored for sign language translation.
2. Insufficient integration of proper SLSL grammatical structures into translation systems.
3. Deficit of high-quality large-scale SLSL datasets perfectly aligned to machine learning.
4. Urgent need for context-aware, mobile-oriented solutions crafted specifically keeping Sri Lanka in perspective.

My project aims to bridge these gaps by developing an app that is able to integrate English speech recognition with accurate translation into SLSL, written keeping the real-world mobile computing limitations and native Sri Lankan setting in mind.

Chapter 4: Requirements Analysis

4.1 Functional Requirements

4.1.1 Speech Recognition Module

The system will be able to capture and process both Tamil and Sinhala speech inputs effectively. Visual cues for speech capture and pausing or resuming recording as needed will be made available to users. The system will also allow users to manually edit recognized text and enter text manually as an alternative. It will also retain common phrases for convenience's sake.

4.1.2 Translation Processing

The system will also translate the speech automatically into text upon capture and proceed to transform this text into the right sign language animations. It will enable basic offline translation, grant users a simple way of agreeing with translations, and encompass repeat translation facilities. The system will translate common phrases and expressions, and it will offer error correction capabilities for accuracy improvement.

4.1.3 Animation Output

Animations will be presented in 2D, with ease of showing sign language movement in a fluid sequence. The system will support required hand movement and minimal facial expression to make reading easy. Users can replay animations, control their speed, and utilize fingerspelling where needed. Common animations will be cached in-memory for faster retrieval.

4.1.4 User Interface

The interface will be supported in both Tamil and Sinhala languages, offer high contrast display settings, and support adjustable text sizes. Emergency sentences will be readily accessible through shortcuts. A beginner's tutorial and extensive help documentation will also be offered to help users navigate and make proper use of the system.

4.2 Non-Functional Requirements

4.2.1 Performance

The system will provide basic offline capabilities, boot in three seconds, and allow for continuous operation without pause. It will be efficient in handling disruption of processing, drain battery highly optimally, and remain responsive even on low-end hardware.

4.2.2 Reliability

If a system crash happens, it will automatically recover with the user's data successfully retrieved. Changes in network conditions will be handled smoothly, display clear error messages, and automatically save work in progress. It should also provide automatic update as well as data backup options.

4.2.3 Security

The system will secure user data with secure local storage and permission requests within authorized access. Users will also have simple privacy settings and delete data options. User privacy will be guaranteed in data transmission via secure communication protocols.

4.2.4 Usability

The system will be accessible without requiring special training, easy to use intuitively, have one-hand usability and easy-to-use navigation, and support simple feedback by users. It will provide a consistent interface with the main features offered for instant access, alongside supporting basic accessibility.

4.3 Technical Requirements

4.3.1 Device Requirements

Minimum required features are Android 4.1 and later, 2GB of RAM, and 32GB of storage. The device should include support for a standard microphone, touch screen, internet connection for updates, and minimum graphical processing support.

4.3.2 Development Requirements

Key development technologies will be the Flutter framework, TensorFlow Lite for machine learning support, SQLite for database support, minimum animation support, efficient local storage, network handling support, and an efficient error logging system.

4.4 Constraints

4.4.1 Technical Constraints

The system should be able to function effectively in low bandwidth, manage device constraints effectively, accommodate various screen sizes, and manage various hardware configurations. It should be able to manage memory and storage efficiently, have its battery life optimized, and manage background operations seamlessly.

4.4.2 Environmental Constraints

The system will function optimally in a variety of lighting conditions and will efficiently handle background noise. It will always operate in normal weather conditions, equally responding to both indoor and outdoor environments. Fluctuations in networks will be handled with ease to ensure even usage across environments.

Chapter 5: Finance

5.1 Development Tools and Software

5.1.1 Cloud Services (Monthly)

Initially, the project will utilize Firebase's free tier for basic database operations, potentially upgrading to the Spark Plan if necessary at no additional cost. Google's Speech-to-Text API offers 60 free minutes monthly, with an estimated additional monthly expense of approximately Rs. 3,200 ($10) beyond the free tier.

5.2 Testing Requirements

5.2.1 Device Testing

Testing will primarily occur on personal Android devices already available to team members. Additional tests will be conducted on laboratory equipment at universities when needed, along with testing on volunteer end-user devices to ensure compatibility and performance.

5.3 API and Service Costs (Monthly)

5.3.1 Development APIs

Initially, free tiers of speech recognition services will be utilized, with estimated monthly costs scaling up to Rs. 3,200 post-deployment. Usage will be monitored for optimization, and offline processing methods will serve as a fallback to minimize expenses.

5.3.2 Storage Services

Cloud storage will be relied upon free tiers at first, though an approximated Rs. 1,600 ($5) per month is anticipated for animation storage needs. Local caching and asset compression methods will be employed to further reduce the expense.

5.4 Operational Costs (Monthly)

5.4.1 Internet and Utilities

Development internet access is provided by the university at no cost, while mobile data for testing purposes is budgeted at Rs. 1,500 per month. Appropriate mobile data packages will be selected based on specific testing cycles, along with data reduction strategies.

5.4.2 Documentation

Online documentation tools and project management solutions such as GitHub and Trello will be utilized at no cost.

5.5 Cost Optimization Strategies

5.5.1 Resource Management

The project will maximize usage of available free service tiers, optimize API calls for cost-efficiency, and utilize university resources where possible. Efficient data storage practices will be adopted to further minimize expenses.

5.5.2 Development Efficiency

Open-source tools and libraries will be prioritized. Efficient testing and structured development processes will be maintained, coupled with regular code optimization to reduce resource demands.

5.6 Detailed Cost Breakdown

5.6.1 Development Tools Cost (One-Time)

All necessary IDE, CLI tools, Git, and project management software will incur no initial costs, totaling Rs. 0 for one-time development setup.

5.6.2 Monthly Service Costs

Monthly costs include Rs. 3,200 for API services, Rs. 1,600 for cloud storage, Rs. 1,500 for mobile data, plus a 15% contingency buffer of Rs. 945, resulting in total monthly expenses of Rs. 7,245.

5.7 Financial Plan Summary

The total project budget of Rs. 57,960 accounts for all necessary development costs and includes strategic contingency buffers. This financially prudent approach allows for flexible adjustments while ensuring essential resources remain readily accessible throughout project execution.

Chapter 6: External Organizations

6.1 Potential Community Partnerships

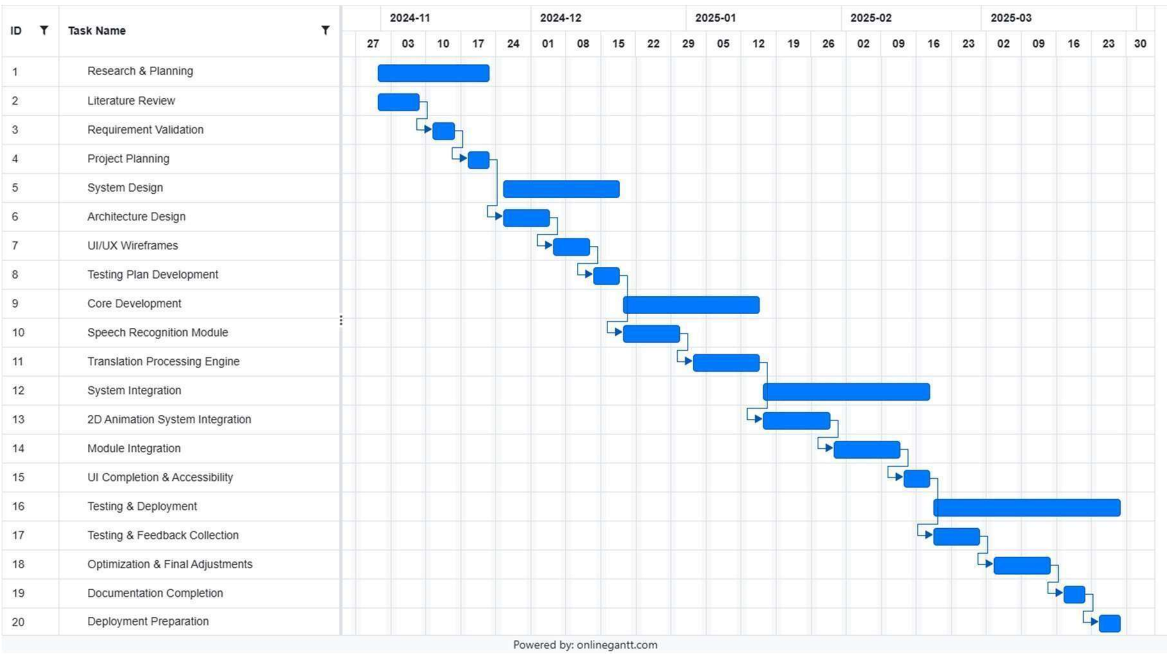
6.1.1 Sri Lanka Central Federation of the Deaf

The project proposes a cooperative relationship with the Sri Lanka Central Federation of the Deaf in the project's most important areas, such as verifying the accuracy of sign language translations, user testing with community members, and obtaining cultural and linguistic input. The partnership would also provide valuable resources related to Sri Lankan Sign Language (SLSL) and give feedback on overall user interface design.

6.1.2 Sri Lanka School for the Deaf

Collaboration with the Sri Lanka School for the Deaf would involve exposing students to actual test environments and gaining feedback on the learning value of the application. The school can provide actual use and user experience feedback to improve the usefulness and applicability of the application.

Chapter 7: Project Timeline



Bibliography

Adithya, R., Sowmya, V. and Soman, K.P., 2020. 'A comprehensive survey on features and methods for sign language identification'. *Artificial Intelligence Review*, 53(6), pp.4159-4226.

Aldaej, R., Alfowzan, L., Alhashem, R., Alhoshan, W. and Alzahrani, B., 2023. 'Mobile-Based Sign Language Recognition: A Systematic Review and Meta-Analysis'. *Sensors*, 23(5), p.2589.

Bragg, D., Shiver, B., Bloch, N., Shiver, S., Morris, M. R. and Vogler, C., 2021. 'Mobile app accessibility: Where are we and what needs to be done?'. *ACM CHI Conference on Human Factors in Computing Systems*, pp.1-15.

Camgöz, N.C., Koller, O., Hadfield, S. and Bowden, R., 2020. 'Sign language transformers: Joint end-to-end sign language recognition and translation'. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.10023-10033.

Jayasekara, P., Madhushanka, B.G. and Ranasinghe, R.A.M.C., 2021. 'Towards Sign Language Recognition in Sinhala: A Novel Framework Based on Deep Learning', *International Conference on Advances in ICT for Emerging Regions*, pp.145-150.

Li, D., Rodriguez, C., Yu, X. and Li, H., 2020. 'Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison'. *IEEE Winter Conference on Applications of Computer Vision*, pp.1459-1469.

Nanayakkara, H.C. and Perera, A.S., 2020. 'A bilingual electronic dictionary for translation between Sinhala and Sri Lankan sign language'. *International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp.276-281.

Stoll, S., Camgöz, N.C., Hadfield, S. and Bowden, R., 2020. 'Text2Sign: Towards sign language production using neural machine translation and generative adversarial networks'. *International Journal of Computer Vision*, 128(4), pp.891-908.

World Federation of the Deaf, 2023. *Sign Language Rights Report*. [online] Available at: <https://wfdeaf.org/> [Accessed 1 November 2024].

World Health Organization, 2023. *World Report on Hearing*. [online] Geneva: WHO. Available at: <https://www.who.int/publications/i/item/world-report-on-hearing> [Accessed 1 November 2024].

Yin, K., Moryossef, A., Hochgesang, J., Goldberg, Y. and Alikhani, M., 2021. 'Including signed languages in natural language processing'. *Association for Computational Linguistics*, pp.73477360.

Zhou, M., Ng, R.W., Li, L. and Xiao, X., 2022. 'Speech to sign language translation: A survey'. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30, pp.1971-1986.