Speech to ASL Application:
Design Document

Table of Contents

1 Introduction **2**

2 Design Goals **4**

3 System Behavior **4**

4 Logical View **4**

4.1 High-Level Design (Architecture) 5

4.2 Mid-Level Design 5

4.3 Detailed Class Design 6

5 User Classes and Characteristics **6**

6 Physical View **6**

1 Introduction

This document describes the architecture and design for the Speech to ASL application being developed for customers that want to translate verbal English to ASL. This document also identifies the Speech to ASL major system components and describes its static attributes and dynamic patterns of interaction.

Speech to ASL is an android app that allows the user to translate his words to ASL(American sign language). The user can write a sentence or speak a sentence to the app. The app will then play a series of videos translating the words in ASL. The Speech to ASL application will be developed using Android Studio and will be open source on github.

The purpose of this document is to describe the architecture and design of the Speech to ASL application in a way that addresses the interests and concerns of all major stakeholders. For this application the major stakeholders are:

- Users and the customer – they want assurances that the architecture will provide for system functionality and exhibit desirable non-functional quality requirements such as usability, reliability, etc.
- Developers – they want an architecture that will minimize complexity and development effort.
- Project Manager – the project manager is responsible for assigning tasks and coordinating development work. He or she wants an architecture that divides the system into components of roughly equal size and complexity that can be developed simultaneously with minimal dependencies. For this to happen, the modules need well-defined interfaces. Also, because most individuals specialize in a particular skill or technology, modules should be designed around specific expertise. For example, all UI logic might be encapsulated in one module. Another might have all business logic.

• Maintenance Programmers – they want assurance that the system will be easy to evolve and maintain on into the future.

The architecture and design for a software system is complex and individual stakeholders often have specialized interests. There is no one diagram or model that can easily express a system's architecture and design. For this reason, software architecture and design is often presented in terms of multiple views or perspectives [IEEE Std. 1471]. Here the architecture of the Speech to ASL application is described from 4 different perspectives [1995 Krutchen]:

1. Logical View – major components, their attributes and operations. This view also includes relationships between components and their interactions. When doing OO design, class diagrams and sequence diagrams are often used to express the logical view.
2. Process View – the threads of control and processes used to execute the operations identified in the logical view.
3. Development View – how system modules map to development organization.
4. Use Case View – the use case view is used to both motivate and validate design activity. At the start of design the requirements define the functional objectives for the design. Use cases are also used to validate suggested designs. It should be possible to walk through a use case scenario and follow the interaction between high-level components. The components should have all the necessary behavior to conceptually execute a use case.

2 Design Goals

The design priorities for the Speech to ASL application are:

- The design should minimize complexity and development effort.
- The design should strive for optimality.
- The design should follow coding standards.
- The design should be clear to all developers.
- This design should focus more on performance rather than usability.

3 System Behavior

The architecture description presented here starts with a review of the expected system behavior in order to set the stage for the architecture description that follows.

Behavior List:

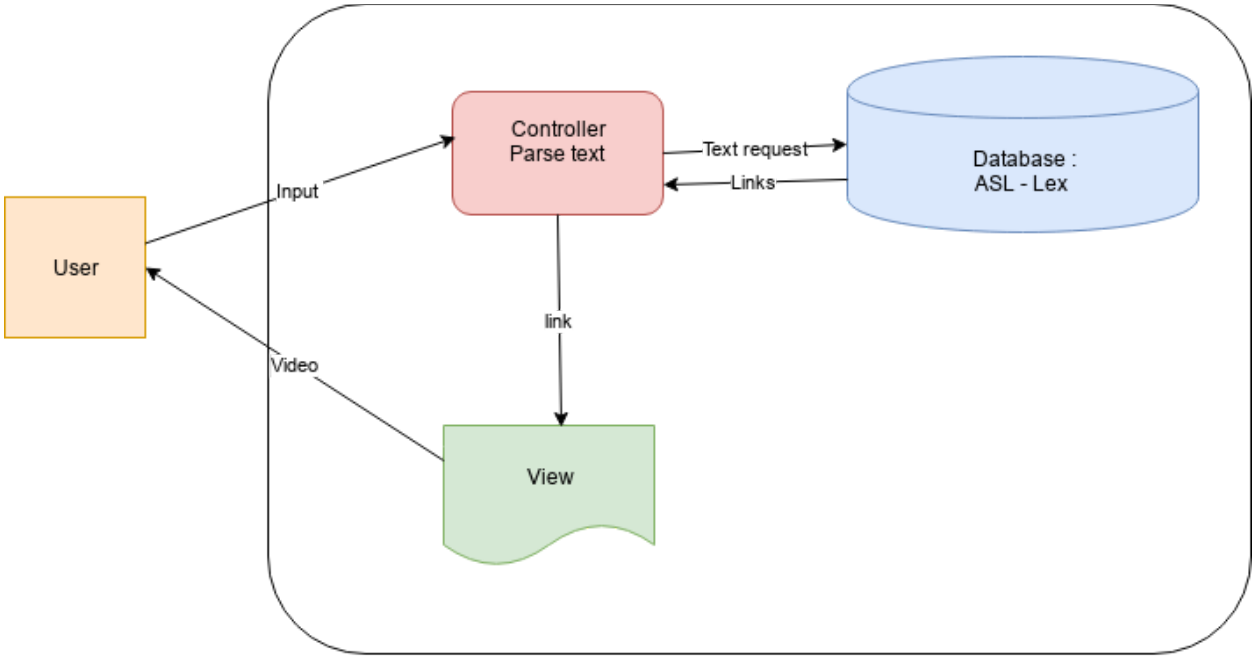
- When user selects the application, the application opens
- When user selects speech, the application goes into speech mode
- When user selects text, the application goes into text mode
- When the user selects the record button, the app translates the speech to text
- When the user de-selects the record button, the app stops translating
- When the text is collected, the app translates it into videos and images
- When the user chooses words, the app highlights and translates these words
- When the user selects a new interface, the app changes interface

4 Logical View

The logical view describes the main functional components of the system. This includes modules, the static relationships between modules, and their dynamic patterns of interaction.

In this section the modules of the system are first expressed in terms of high level components (architecture) and progressively refined into more detailed components and eventually classes with specific attributes and operations.

4.1 High-Level Design (Architecture)

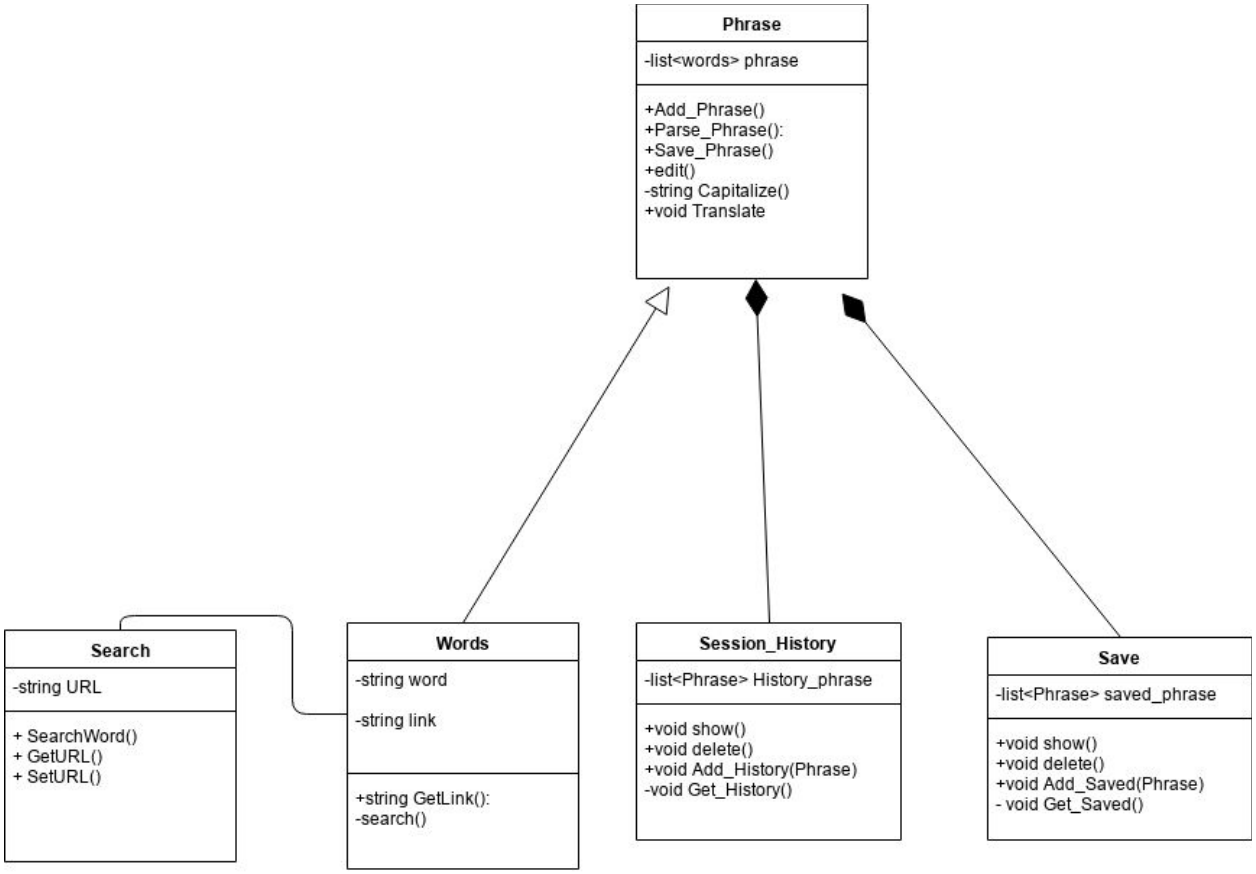


The Speech to ASL App will take user's input text or speech and output a text phrase and an American Sign Language (ASL) video translation taken from the ASL-Lex database. These translations will then be stored into a Phrase History list.

4.2 Mid-Level Design

We will have a phrase class that will store the phrases the user inputs either through text or voice. From there the phrase class will parse the phrase into a list of word objects. These word objects will work with the search class in order to search the words in the database and then retrieve a video link that will be played. We also have a Session_History class that will store all the phrases the user has been searching and when the user closes the app the history will clear itself. And finally we have a saved class where the user can save certain phrases and have them to be used later, even after exiting the application.

4.3 Detailed Class Design



<For a few key classes you might want to show associations, attributes and methods.>

5 User Classes and Characteristics

The following are sample scenarios for the Speech to ASL application.

Scenario 1: Non-ASL speaker wants to communicate with ASL speaker

- Description: A Non-ASL speaker, Jane, has a deaf friend that only speaks in ASL. Jane has trouble communicating with her friend because she knows very little about ASL. Jane would like a way to speak to her deaf friend quickly at any location.

Scenario 2: Non-ASL speaker wants to learn ASL

- Description: A Non-ASL speaker, Tom, is taking an introduction to ASL class. Tom wants to practice his ASL skills, but sometimes gets stuck while trying to speak it fluently. Tom would like a way to translate English into ASL when he gets stuck.

6 Physical View

Major components will be physically deployed on Android devices that support Android 7.0 Nougat.