SignSync

Demo 1

Apollo projects & Gendac

Introduction

- Working model with:
  1. Sign-to-text translation for ASL alphabet
  2. Speech-to-text translation (complete)
  3. Text-to-sign translation
  4. Platform and device compatibility
  5. Unit testing
  6. CI/CD

- Use cases 1-3 forms the foundation of the core translation requirements of the translating system

Translation process

- Speech-to-sign ⏳

  Speech-to-text ✅

  ↓

  Text-to-sign ✅

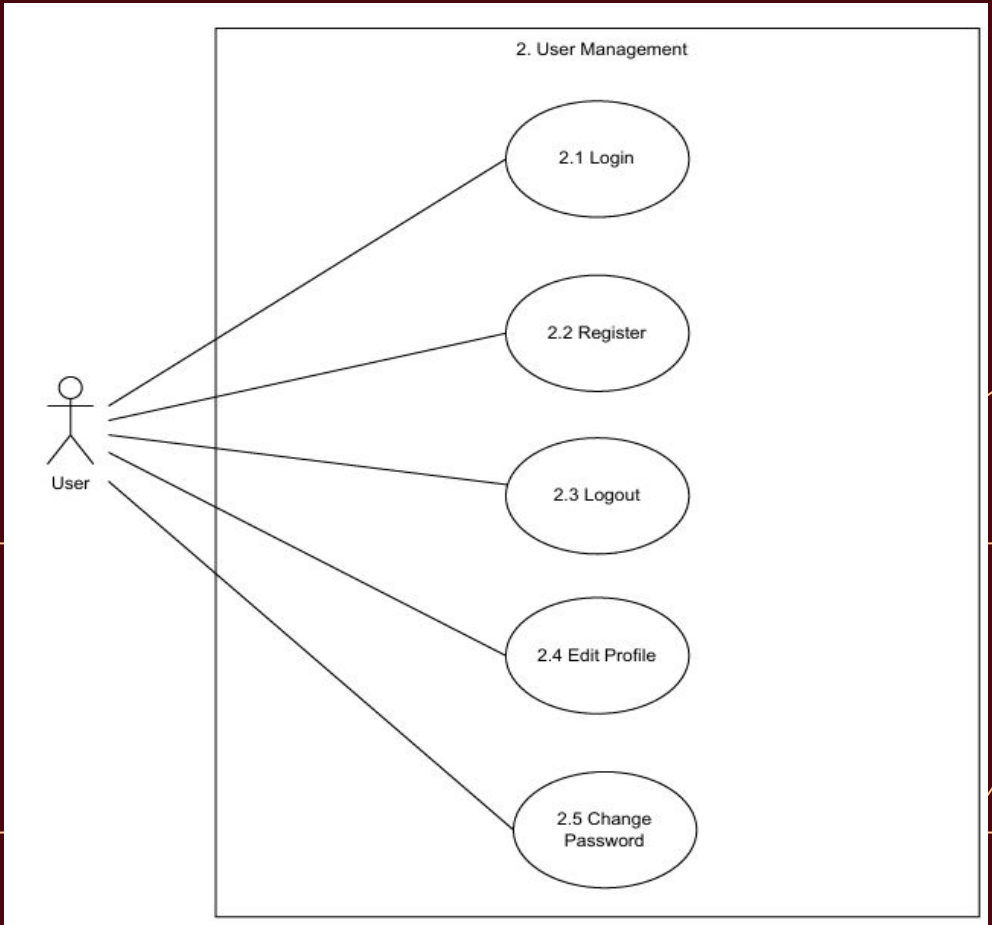
- Sign-to-speech ⏳

  Sign-to-text √

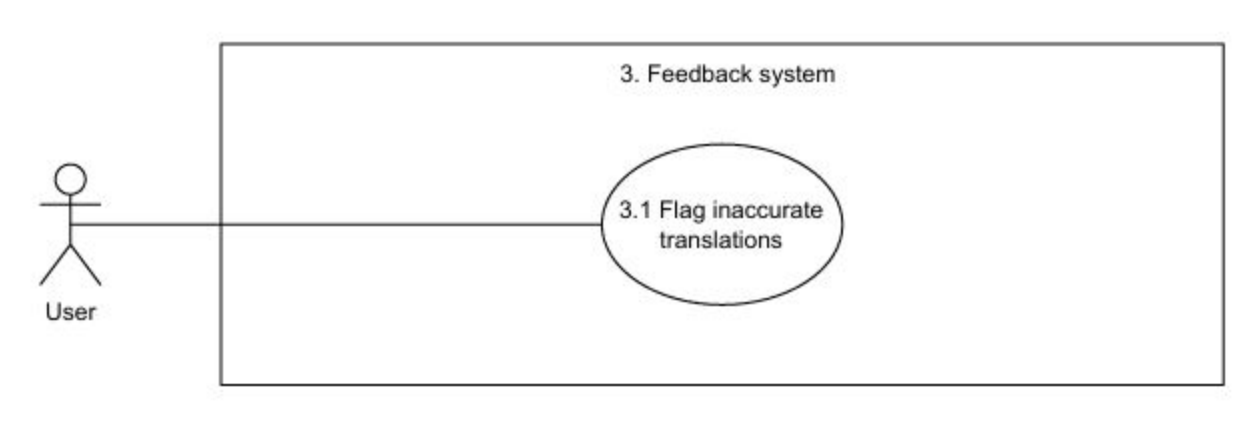
  ↓

  Text-to-speech ⏳

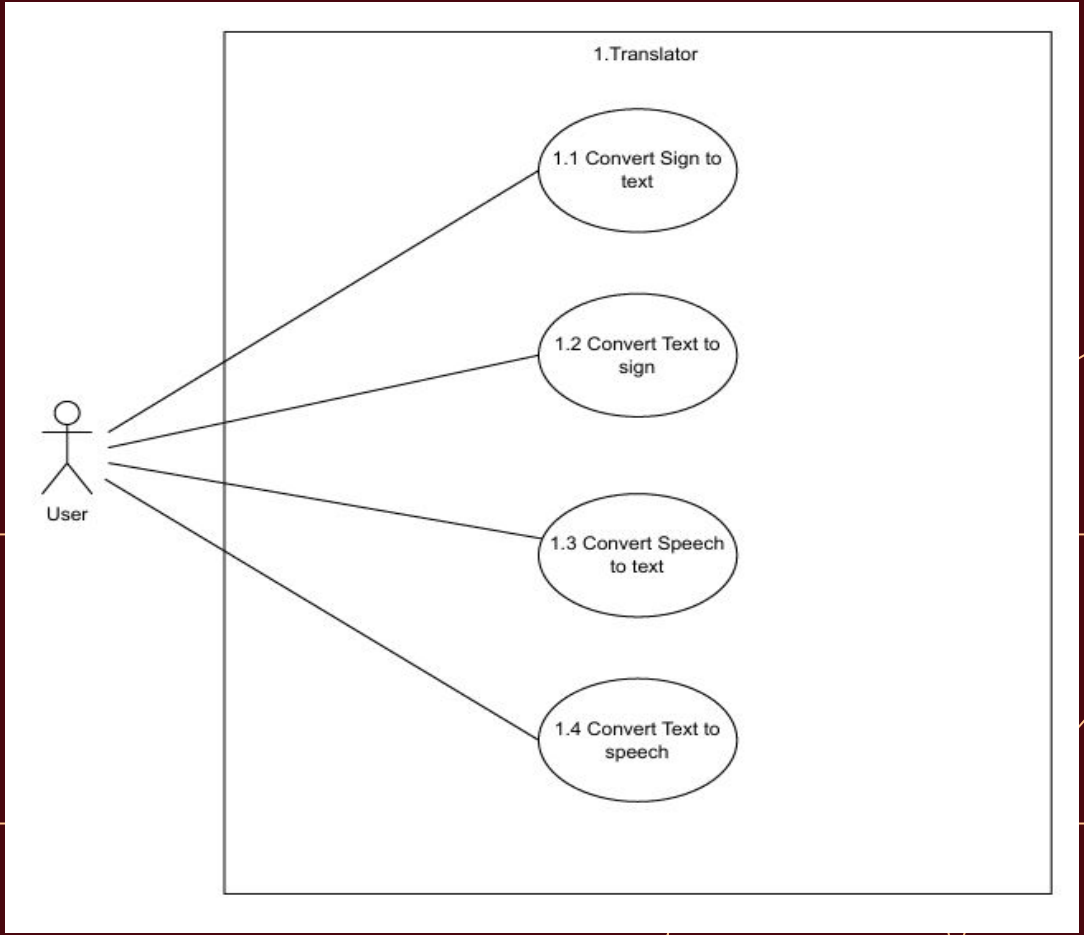
Use Case Diagrams continued



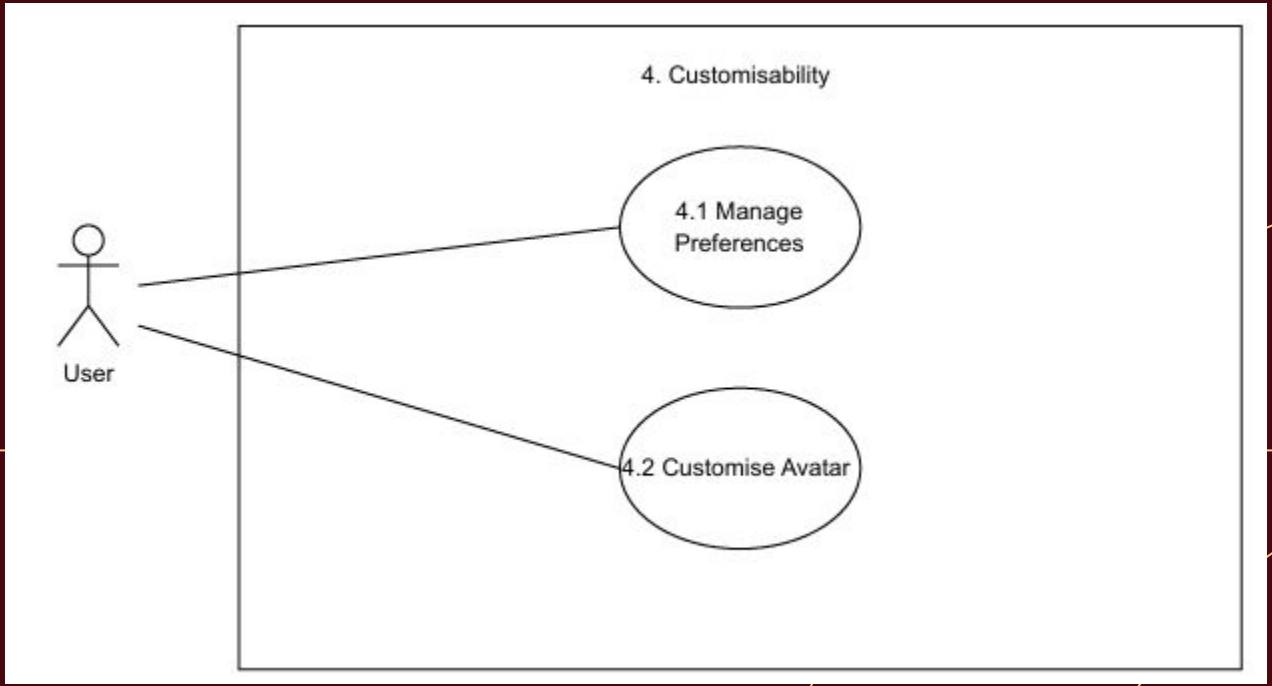
Use Case Diagrams continued



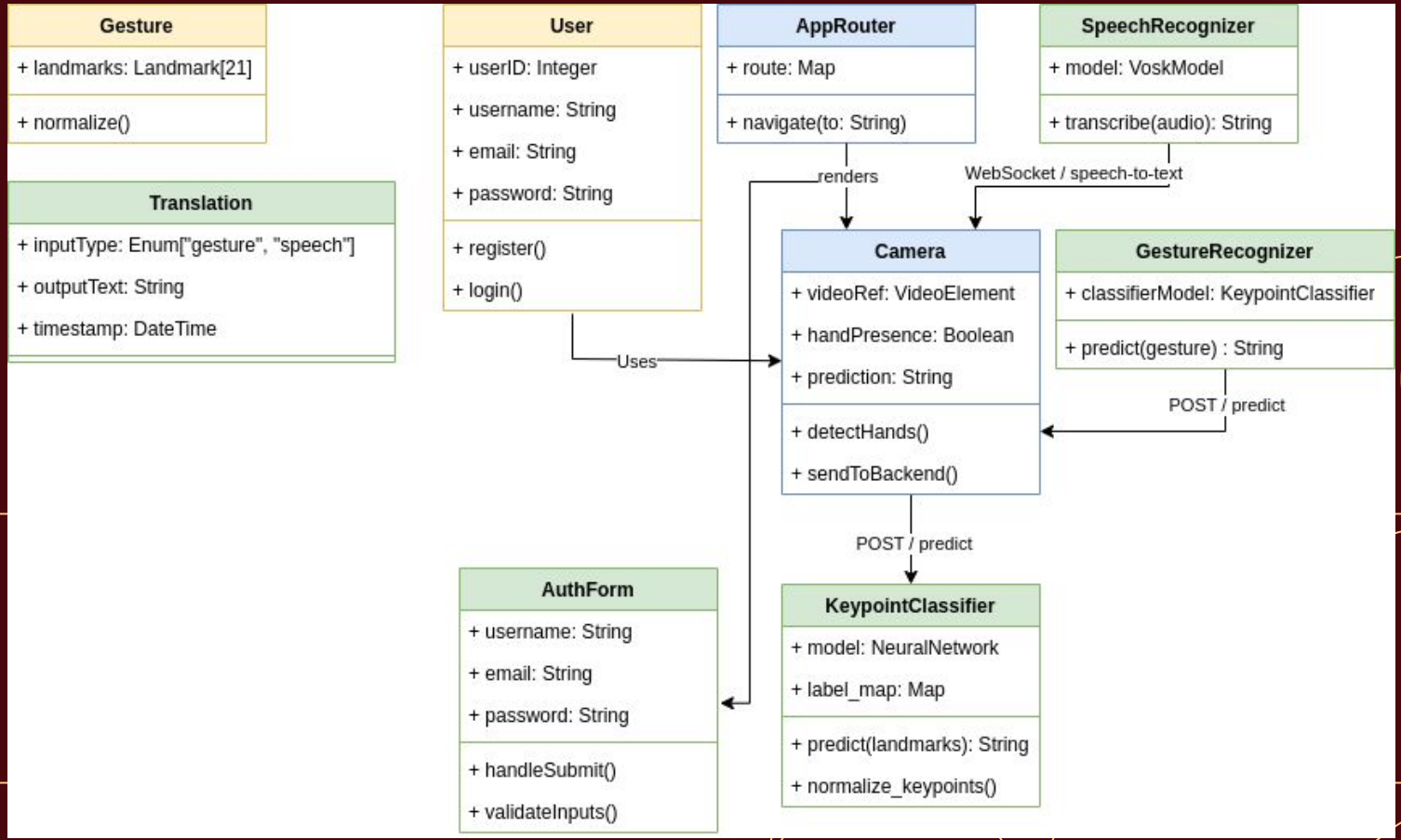
Use Case Diagrams



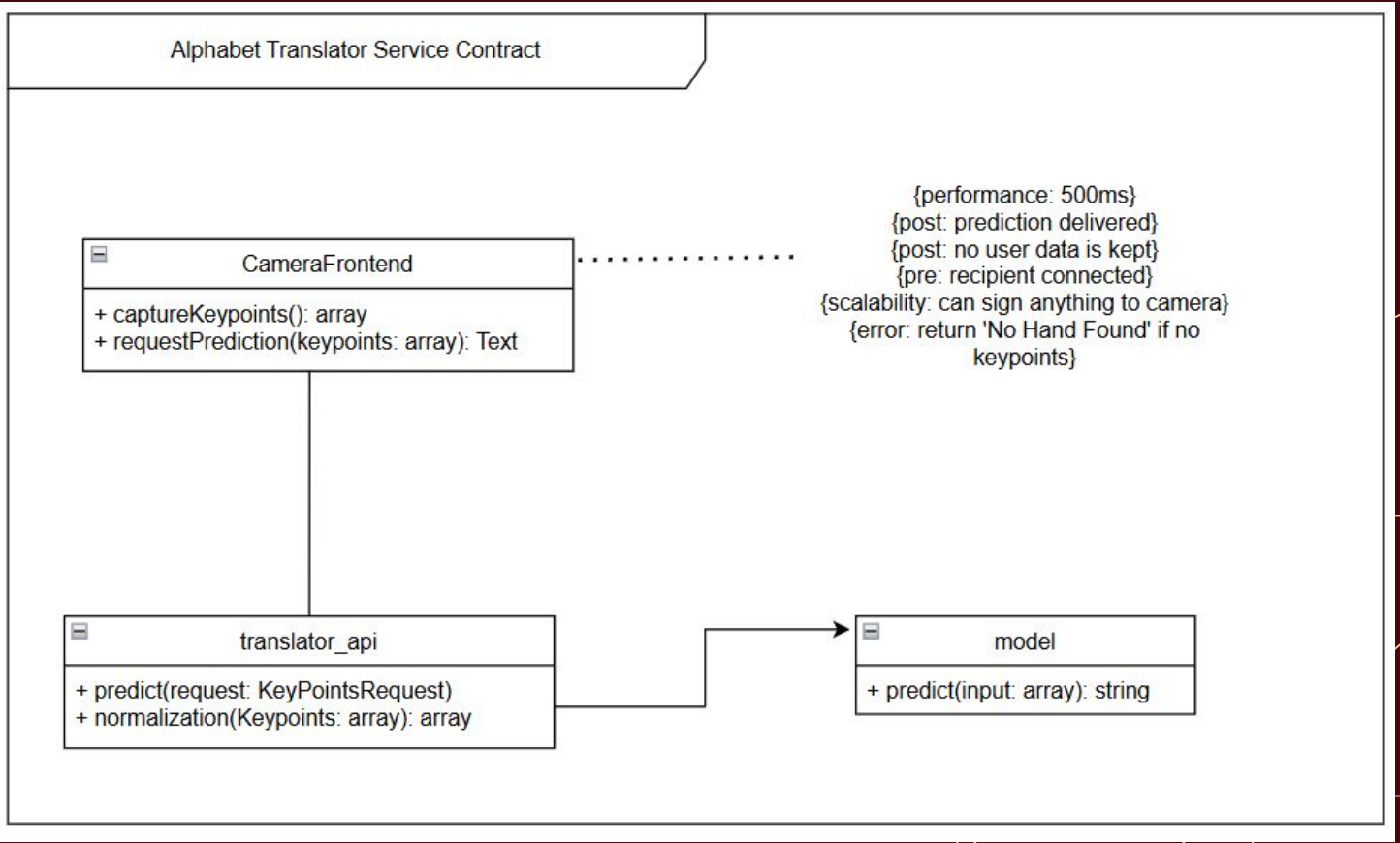
Use Case Diagrams continued



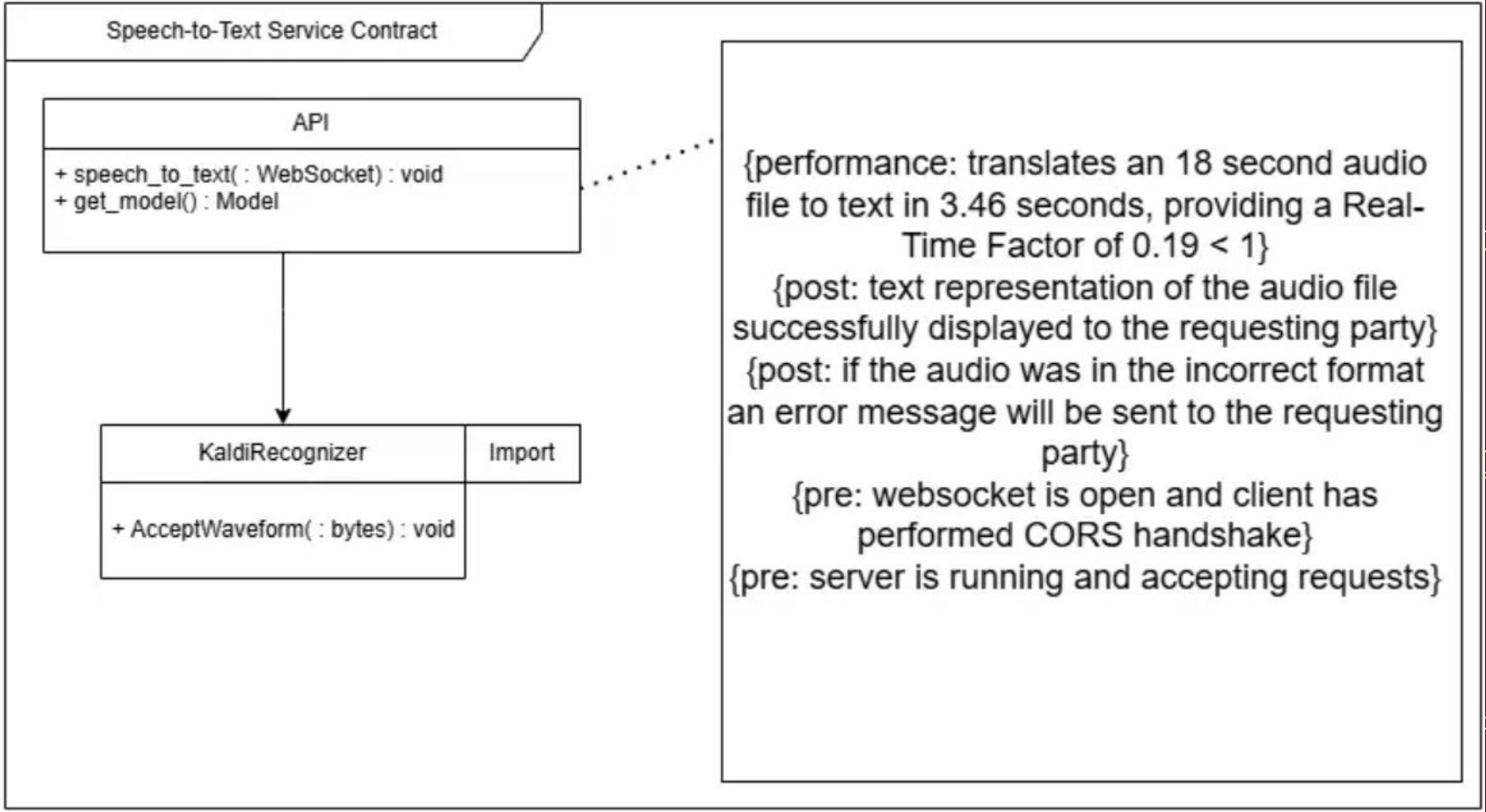
Domain Model



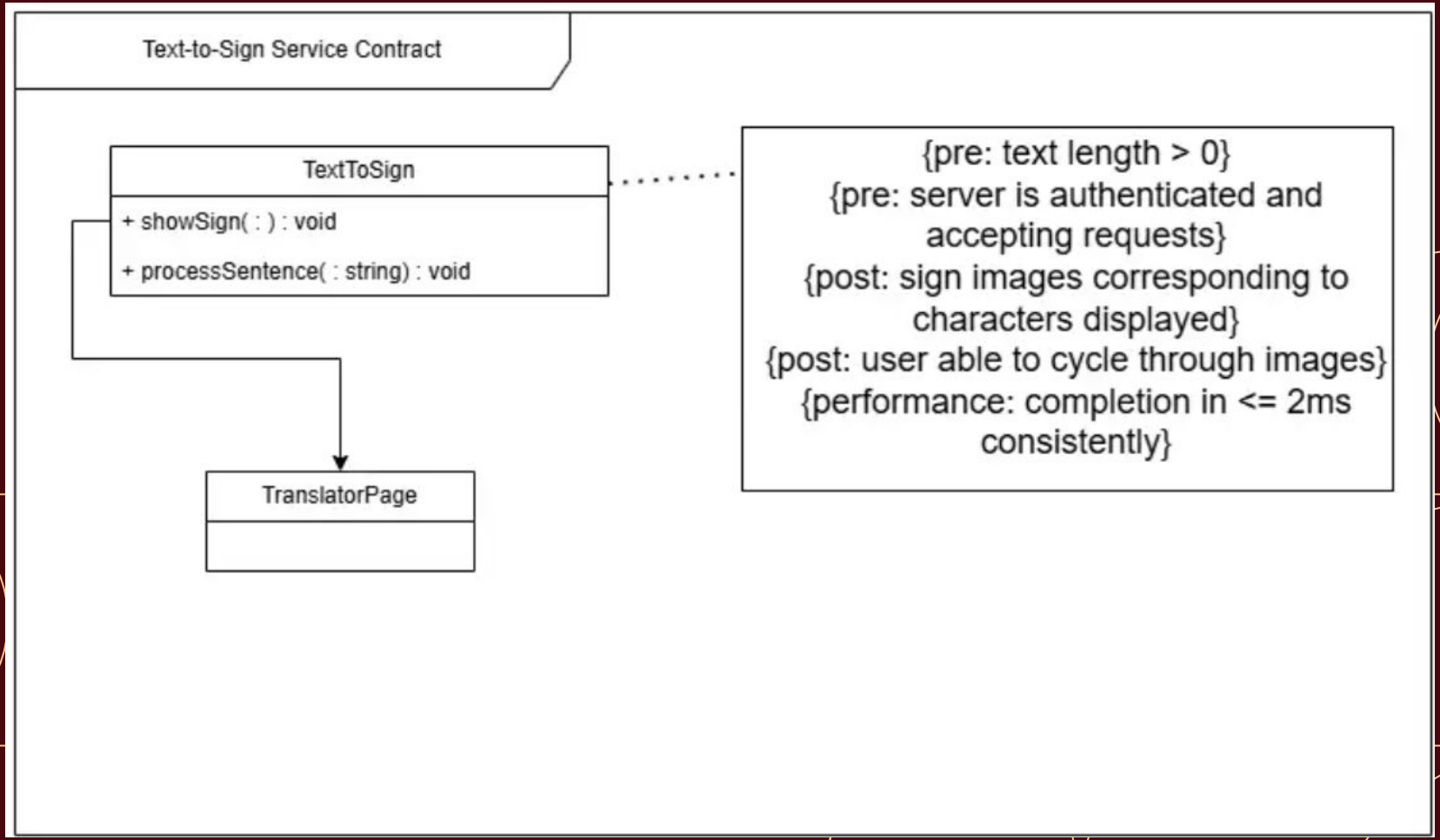
Service Contracts: Sign-to-text



Service Contracts: Speech-to-text



Service Contracts: Text-to-sign



Micro-service Architecture

- Scalability
  => can scale important services independently

- Incremental development fits well with Agile-Scrum methodology

- Allows each team member to develop within preferred framework

- Independent services allows to split the translation system into independent parts, each with clear responsibilities
  => ease of debugging, less code per module
  => independent updates
  => simplifies complex system
  => fault isolation

Technology Stack

- Front-end (languages)
  => JavaScript
  => HTML
  => CSS
- Front-end (frameworks)
  => React
  => Tailwind

- Back-end
  => Python
  => MongoDB
  => NodeJS
  => ExpressJS

SRS Review

- User Stories
- Functional Requirements
- Architectural Requirements
  => Quality Requirements - performance, scalability, modularity
  => Architectural Patterns - microservices
  => Design Patterns
  => Constraints
  => Technology Requirements

User Stories

1. **Deaf or Hard-of-Hearing Users:**
   - View spoken language translated into sign animations.
   - Use sign language to communicate back via webcam input.

2. **Hearing Users:**
   - Speak naturally and have their speech translated into signs.
   - Read or hear signed responses translated to text or audio.

3. **Administrators/Researchers**
   - Monitor system performance.
   - Gather feedback data for AI retraining.
   - Manage user access and customization settings.

Functional Requirements

R1: text-to-Sign Translator
- R1.1: Capture text input from user.
- R1.2: Translate English text to Sign gloss.
- R1.3: Search word definition for appropriate sign.
- R1.4: Display sign through avatar.

R2: Sign-to-Text Translator
- R2.1: Capture webcam input and extract hand keypoints
- R2.2: Classify sign gesture sequences using trained AI model
- R2.3: Convert recognized signs to sign gloss
- R2.4: Convert sign gloss to English

R3: Feedback and AI Improvement System
- R3.1: Allow users to flag inaccurate translations
- R3.2: Log flagged data
- R3.3: Retrain AI models periodically using collected data

R4: User Interface
- R4.1: Display live avatar animations based on translation output
- R4.2: Show translated text and/or play voice feedback
- R4.3: Offer accessibility options (high-contrast mode, font scaling, voice personas)

R5: User Management and Settings
- R5.1: Authenticate users
- R5.2: Store and retrieve user preferences and settings
- R5.3: Allow users to login
- R5.4: Allow users to register

R6: Speech to Text
- R6.1: Capture user speech
- R6.2: Convert speech to text
- R6.3: Display text on screen

R7: Text to Speech
- R7.1: Capture text input by user
- R7.2: Convert text to Speech
- R7.3: Play speech for user to hear

Live Demo

Unit Tests

- alphabetTranslator-service
- speechToText-service
- register
- login

Unit Tests: Speech-to-text and Text-to-speech

```
(base) PS C:\Users\wesse\OneDrive\Desktop\CS\2025\Sign-Sync\Sign-Sync\backend\tests> pytest speechTests.py
========================================== test session starts ==========================================
platform win32 -- Python 3.12.4, pytest-8.2.0, pluggy-1.6.0
rootdir: C:\Users\wesse\OneDrive\Desktop\CS\2025\Sign-Sync\Sign-Sync\backend\tests
plugins: anyio-4.9.0, asyncio-0.23.6
asyncio: mode=Mode.STRICT
collected 4 items

speechTests.py ....                                                                        [100%]

========================================== 4 passed in 6.72s ==========================================
(base) PS C:\Users\wesse\OneDrive\Desktop\CS\2025\Sign-Sync\Sign-Sync\backend\tests> pytest speechTests.py
========================================== test session starts ==========================================
platform win32 -- Python 3.12.4, pytest-8.2.0, pluggy-1.6.0
rootdir: C:\Users\wesse\OneDrive\Desktop\CS\2025\Sign-Sync\Sign-Sync\backend\tests
plugins: anyio-4.9.0, asyncio-0.23.6
asyncio: mode=Mode.STRICT
collected 4 items

speechTests.py ....                                                                        [100%]

========================================== 4 passed in 5.58s ==========================================
```

Development Plan

- Model for gesture recognition
  => Long Short Term Memory Recurrent Neural Network (LSTM RNN)
  => for classification of sequences

- Model for Text-to-Sign translation
  => Small to Large Language model
  => for translation of english grammar / sign language grammar to signs

- ~~Optionally a service to convert normal text into sign language grammar~~
  => sign language grammar is different
  => e.g. “I go shop” vs “I am going to the shop”
  => could boost LLM training efficiency,

Development plan continued

- Choosing a cloud service provider
- Deploying microservices to the cloud platform
- Avatar functionality
  => with ability to “speak”
  => with ability to show signs
- AI and Machine learning enhancements
  => refine translation accuracy via
    - User feedback mechanism
    - Machine learning techniques (back propagation)
- UI customizability
  => avatars, font size, color contrast

CI/CD

Thank you