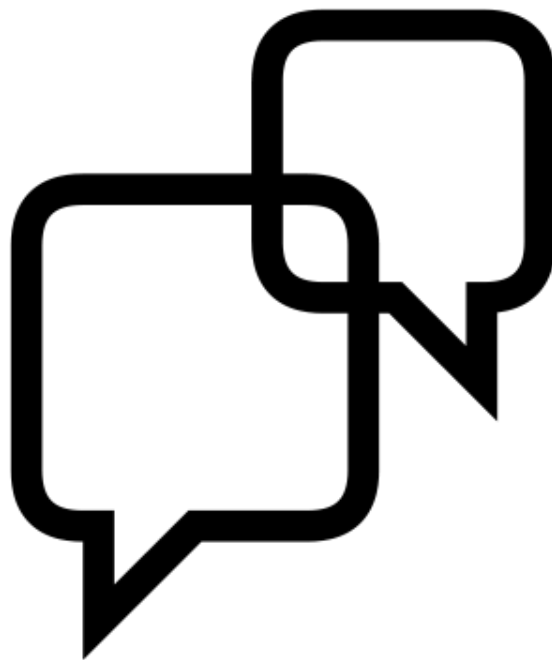


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
FALL 2017**



**TEAM C
LANGUAGE PRONOUNCIATION ASSISTING APP**

JOSUE C.
ALI S.
NOWREEN J.
XIWEN D.
KRISTEN R.

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	10.22.2017	GH	document creation

CONTENTS

1	Introduction	3
2	Architecture Overview	4
2.1	Client Layer	4
2.2	Server Layer	4
3	Subsystem Definitions & Data Flow	5
4	Client Layer Subsystems	6
4.1	Client Subsystem	6
5	Server Layer Subsystems	8
5.1	Server Subsystem	8
6	Hardware Interface Specifications	10
7	Installation	10
8	Appendix A	10

LIST OF FIGURES

1	A high-level data-flow diagram for our application	4
2	The data flow diagram for our app	5
3	Client subsystem diagram	6
4	Server subsystem diagram	8

LIST OF TABLES

2	Client Subsystem interfaces	7
3	Server Subsystem interfaces	9

1 INTRODUCTION

This section describes the purpose, use and intended user audience for the Language Pronunciation App. The Language Pronunciation App is an application that helps users improve their pronunciation of phonetically difficult words.

Users will be able to visualize their distance to the “perfect” phonetic pronunciation of a word.

2 ARCHITECTURE OVERVIEW

In order to minimize the burden of processing on the individual clients that utilize our system the application takes a lightweight-client approach to the traditional client-server application.

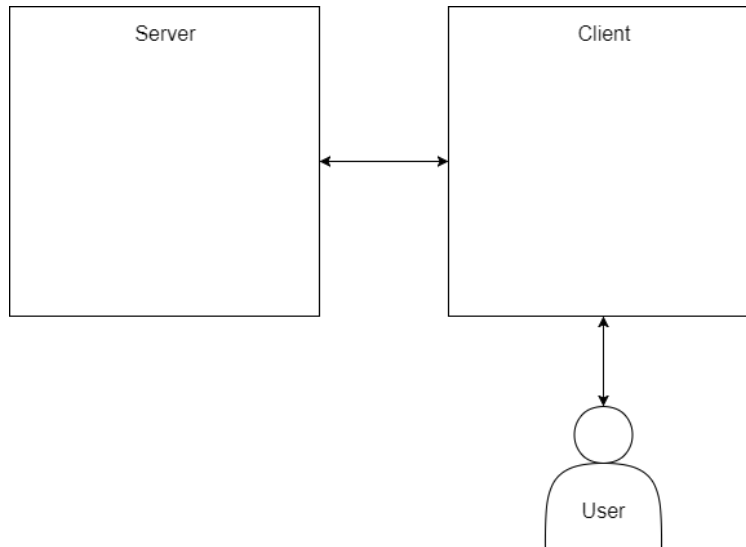


Figure 1: A high-level data-flow diagram for our application

2.1 CLIENT LAYER

The client will function as the HMI between our system and the user. It consists of a UI, and methods by which it may communicate with the server.

2.2 SERVER LAYER

The server will function as the work-horse for the application. It's purpose is to host the visualization function which will map an input of word, audio to a distance metric in the form of x, y co-ordinates.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

This section breaks down your layer abstraction to another level of detail. Here you graphically represent the logical subsystems that compose each layer and show the interactions/interfaces between those subsystems. A subsystem can be thought of as a programming unit that implements one of the major functions of the layer. It, therefore, has data elements that serve as source/sinks for other subsystems. The logical data elements that flow between subsystems need to be explicitly defined at this point, beginning with a data flow-like diagram based on the block diagram.

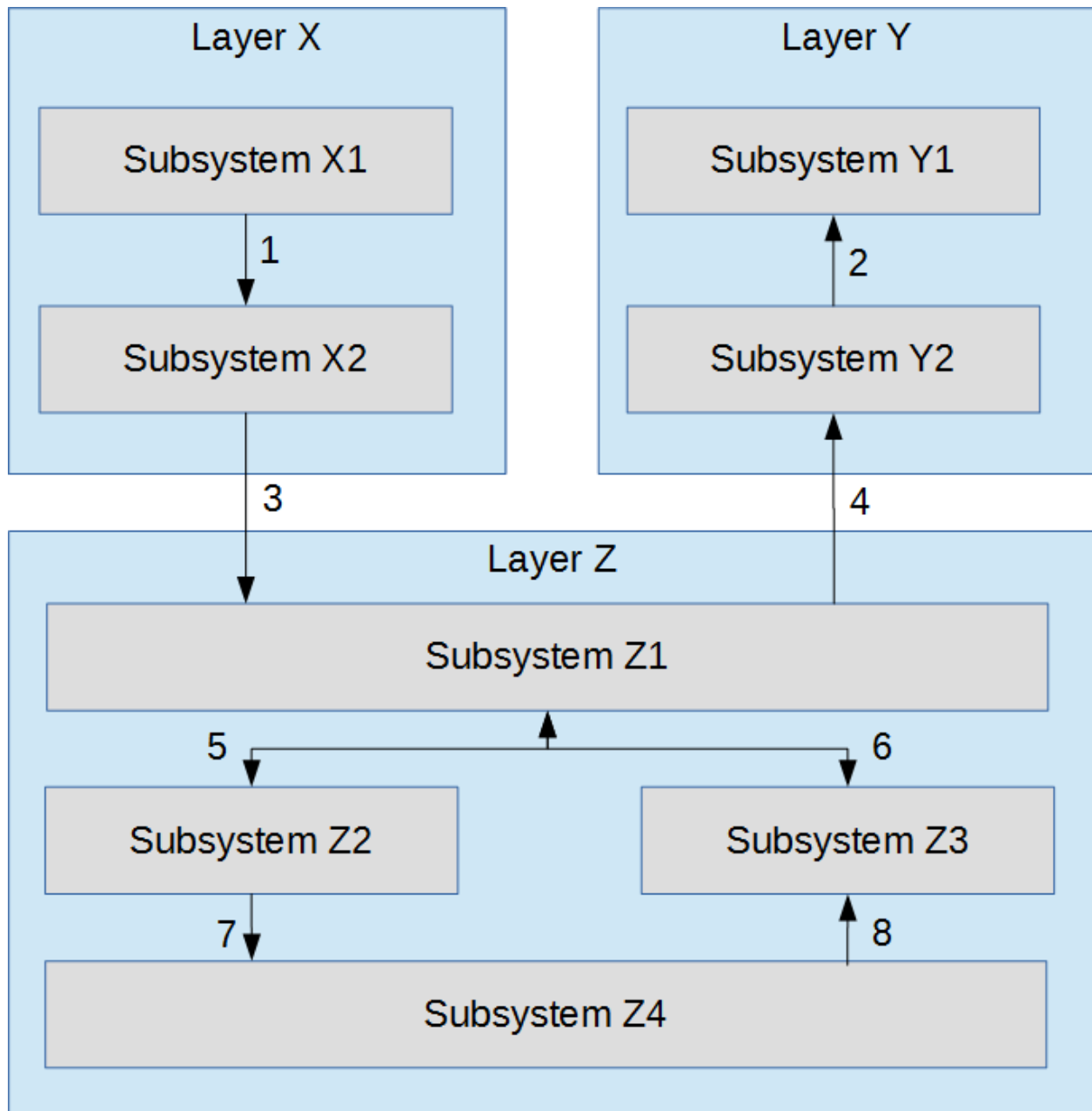


Figure 2: The data flow diagram for our app

4 CLIENT LAYER SUBSYSTEMS

4.1 CLIENT SUBSYSTEM

The client layer subsystem will display the UI of the application. It has a XY graph and a dot calculator. It has a button for record the voice of the users

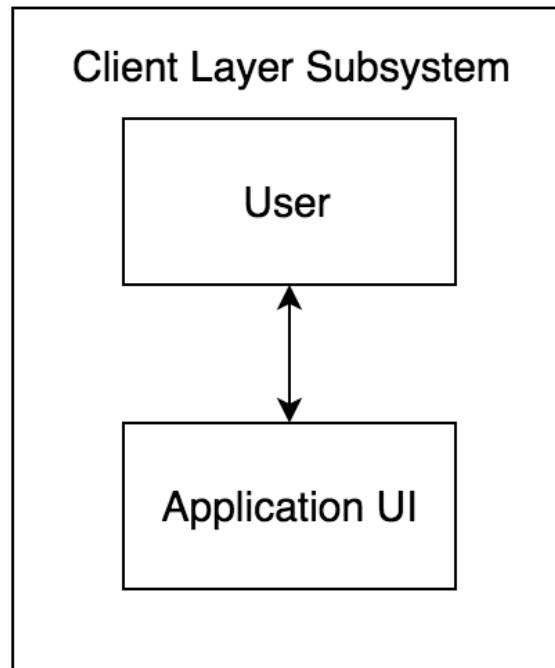


Figure 3: Client subsystem diagram

4.1.1 ASSUMPTIONS

The users press the record button, it will be recorded the sound and display the accurate dot.

4.1.2 RESPONSIBILITIES

The Client Layer Subsystem will send the user's voice to the application, and interaction with user. And it will display a dot on the application UI. The Client Layer should display all information to user. And receive user's request.

4.1.3 SUBSYSTEM INTERFACES

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 2: Client Subsystem interfaces

ID	Description	Inputs	Outputs
#1	XY Graph	N/A	Red dot
#2	Voice Record Button	User's sound	N/A

5 SERVER LAYER SUBSYSTEMS

5.1 SERVER SUBSYSTEM

The server layer subsystem will calculate the difference between the mean representative audio stored on the server and the input audio. It runs continuously and accepts POST requests containing audio and a target word.

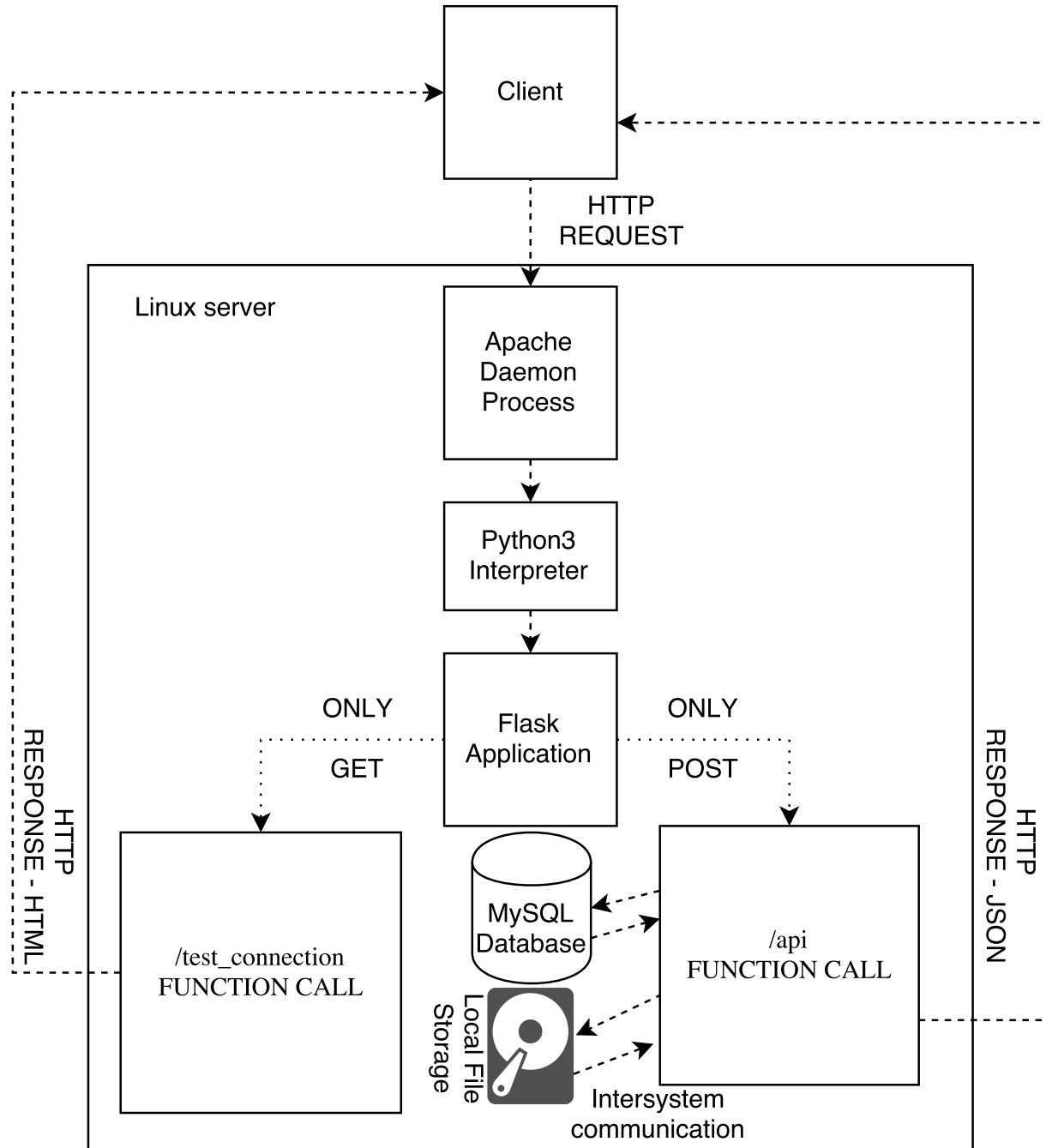


Figure 4: Server subsystem diagram

5.1.1 ASSUMPTIONS

The client will only send queries which the server can answer. At the heart of this assumption is the idea that POST requests should be sent in a specific format.

5.1.2 RESPONSIBILITIES

The server layer subsystem will be responsible for:

- Accepting and maintaining a POST request.
- Temporarily storing and removing when done the input audio file.
- Extracting features from each locally stored audio sample and aggregate into a single representative feature set.
- Extracting the feature set from the input audio sample and aggregate into a single representative feature set.
- Concatenate, normalize, and compare each representative feature set.
- Return a score for the difference.

5.1.3 SUBSYSTEM INTERFACES

ID	Description	Inputs	Outputs
#1	/test_connection/		HTML=HELLO WORLD
#2	/api/	(word,word.mp3)	JSON={word:word, score:difference_ - measure}

Table 3: Server Subsystem interfaces

6 HARDWARE INTERFACE SPECIFICATIONS

The Language Pronunciation App was designed in Android Studio. The app requires an Android smartphone or an Android emulator to run.

The app requires the device it is used on to have a microphone to record sound. Speakers are also recommended.

7 INSTALLATION

To install the app, download it from Google Play.

8 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.