Data Design

# Data Objects and Resultant Data Structures

Signs are stored in two different data structures. The data structure “letter” holds absolute information about a sign such as the number of fingers shown, the percentage area of the hand, and the closest point of the hand. This data structure is used to recognize a sign during training before the user is calibrated. Functions on the data access layer retrieve letter data from the database to compare with the input from the Kinect. The second data structure “sign” holds information specific to an individual user and inherits all of the basic information from “letter”. During training, user specific calibrated information is stored in the database. This data contains the area of a user’s hand. The sign data structure is used to recognize signs more accurately.

# Database Structure

//chou needs to update the ERD to move stuff into alphabet

<insert ERD here>

The alphabet table is where information is stored for the “letter” data structure. It stores non changing information used to recognize signs. The primary key is letter so it only has one entry per letter in the alphabet.

The sign table is where calibrated sign recognition information is stored. It has a compound primary key using letter and username to hold one entry per letter per user. Username is a foreign key to the user table and letter is a foreign key to the alphabet table.

The user table holds user information such as their username, first name, last name, and current training progress. The primary key is username to keep them unique throughout the application. Training progress is a foreign key to the alphabet table and stores the letter that the user is calibrated up to.

The scores table stores score information for a specific user session. It stores the total score, number of signs passed, number of signs failed, and the session ID. The session ID is a unique primary key used to differentiate sessions. Username is a foreign key to the user table used to store session information per user.