Software Requirements Specification (SRS)

For

SignToLearn – ASL Alphabet Learning Application

Wentworth Institute of Technology
Department of Computer Science and Networking
Software Design and Development – COMP566

Jared Siraco, Garrett Lister and Chou Yang

Table of Contents

Introduction

1.1 Purpose of this Document 3
1.2 Scope of the Development Project 3
1.3 Definitions, Acronyms and Abbreviations 3
1.3.1 Acronyms 3
1.3.2 Definitions 3
1.3.3 Abbreviations 4
1.4 References 4
1.5 Overview of Document 4

General Description

2.1 User Characteristics 4
2.2 Product Perspective 4
2.3 Overview of Functional Requirements 4
2.4 Overview of Data Requirements 5
2.5 General Constraints, Assumptions, Dependencies, Guidelines 5
2.5.1 Constraints 5
2.5.2 Assumptions 5
2.5.3 Dependencies 5
2.5.4 Guidelines 5
2.6 User View of Product Use 6

Specific Requirements

3.1 External Interface Requirements 7
3.2 Detailed Description of Functional Requirements 7
3.3 Performance Requirements 13
3.4 Quality Attributes 13
3.4.1 Security 13
3.4.2 Availability 13
3.4.3 Reliability 13
3.4.4 Maintainability 13

Other Requirements 14

Use Cases 14

Appendix 15

6.1 Interaction Diagram 15
6.2 Component Diagram 15

1. INTRODUCTION

1.1 Purpose of this Document

This SRS describes the purpose of the SignToLearn application. SignToLearn is a standalone application that works with the Microsoft Kinect camera. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints and interface. This document is primarily intended to be a reference for developing the first version of the system for the development team.

1.2 Scope of the Development Project

SignToLearn is an application working with the Microsoft Kinect camera to teach users the American Sign Language (ASL) alphabet. This will be accomplished by first going through a training step where SignToLearn will instruct through video demonstrations how to sign each letter and ask that that action be repeated several times in order to collect necessary data. After the required training step the testing portion will be enabled where they will have to sign letters as instructed by SignToLearn at defined intervals. Data captured from the Microsoft Kinect's camera would check the users' accuracy.

1.3 Definitions, Acronyms, and Abbreviations

<u>1.3.1 Acronyms</u>

ASL: American Sign Language

SDK: Software Development Kit

SRS: Software Requirements Specification

SQL: Structured Query Language

UI: User Interface

<u>1.3.2 Definitions</u>

K-means Algorithm: Groups data points into K clusters based on how similar they are (K is the number of clusters)

Microsoft Kinect: A motion sensing input device that captures data in two dimensional video and three dimensional coordinates.

Kinect near mode: A function exclusive to the Kinect for windows version that allows the better capture of objects between 600 and 800 millimeters away from the camera.

1.3.3 Abbreviations

3D: Three Dimensional

1.4 References

Candescent, S. (2013, January 4). Candescent nui. Retrieved from http://candescentnui.codeplex.com/documentation

Get started building kinect for windows apps and experiences. (n.d.). Retrieved from http://lifeprint.com/asl101/topics/wallpaper1.htm

Vicars, W. (n.d.). *Sign language*. Retrieved from http://lifeprint.com/asl101/topics/wallpaper1.htm

1.5 Overview of Document

This SRS will review the hardware, software, interface and user requirements as well as the constraints and dependencies of the SignToLearn application.

2. GENERAL DESCRIPTION

Using Microsoft's Kinect hardware, SignToLearn will allow users to learn the ASL alphabet like never before. This is not a passive learning application. Using the user's own input, the Kinect will be able to guide the user into performing the ASL alphabet. This sets SignToLearn apart from any other ASL teaching utility on the market.

2.1 User Characteristics

As a teaching utility using data captured from the Kinect camera to teach and test the user on the ASL alphabet, the user will need to be physically able to perform the letters of the ASL alphabet. Users who cannot physically perform the signs for various reasons would not be able to use SignToLearn properly.

2.2 Product Perspective

SignToLearn is a stand-alone application which interfaces with the user through a graphical user interface and also through a natural user interface using a Microsoft Kinect, a motion sensing input device which features a z-pixel depth sensor and is based on a webcam design. This application uses the Microsoft Kinect to capture 3 dimensional models as well as detect defined gestures as input.

2.3 Overview of Functional Requirements

First the user will need to be able to create and load profiles to store their configuration and test data in. Next the user's hand will need to be calibrated so that the application can read the user's signs as accurately as possible. This will be done by having the user hold their hand at a specified distance to determine the users finger placement and minimize background interference. After calibration, the application will show the user how to sign each letter and have them sign it several times. Data will be gathered from this and recorded to help the application recognize gestures later on. Once the user is ready the application will display words for users to spell in ASL, the user will then be asked to spell each letter of the displayed word at defined intervals. If the user signs a letter incorrectly they will be asked to sign that letter again at the next interval until it is done correctly.

2.4 Overview of Data Requirements

Gesture tracking and configuration data will be gathered through the Kinect device and used during runtime within the application. Users should be able to save the configuration data and access it through created profiles. Data from the learning phase will be saved to each user profile as well

2.5 General Constraints, Assumptions, Dependencies, Guidelines

2.5.1 Constraints

This product must be used in a clutter free space.

The user must stand in front of the Kinect with their hand between 600 and 800 millimeters away from the Kinect.

2.5.2 Assumptions

N/A

2.5.3 Dependencies

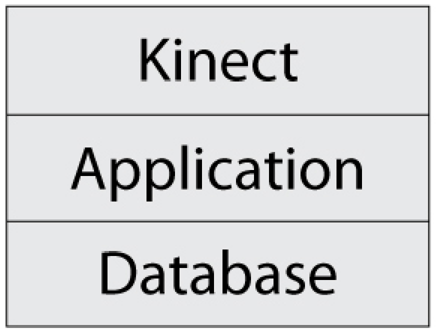
This product will only work on windows systems enabled to work with Kinect hardware.

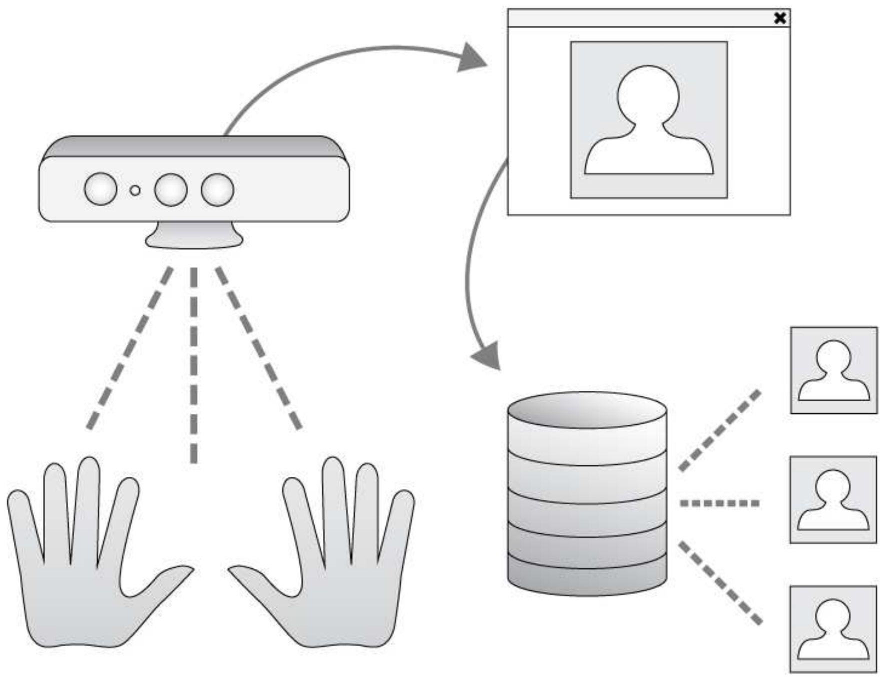
This product needs a Microsoft Kinect for Windows (Near mode enabled)

2.5.4 Guidelines

This product must be able to calibrate for many different hand sizes and shapes.

2.6 User View of Product Use





The user begins by ether creating a profile, loading a profile, or choosing to sign in as a guest. They are then directed to the application landing page where they have two choices, ether to start training or testing. If the user has not completed training, the testing option will not be available. The training option begins by the user holding their hand close to the Kinect to properly configure the application for their hand. Then the user is instructed on how to sign each letter and to repeat the sign several times. Once training is complete, the testing option is made available to the user. Once the user chooses the testing option the user will be displayed letters to sign. After this the user will see each letter displayed and will be directed to make the sign for that letter. If the user gets the letter correct the next letter will be displayed, otherwise the user will be shown a demonstration of the letter to be signed and directed to try to sign the letter again. After the word is completed, another word will be displayed to complete until the user desires to end their session and save their progress if signed in under a profile.

3. SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

This application needs access to the Kinect SDK version 1.8 to be able to interact with the Kinect camera. A SQL database is also needed to be able to save and load the potentially large amounts of user data efficiently.

3.2 Detailed Description of Functional Requirements

Component	Purpose	Input(s)	Processing	Output(s)
Load profile button	Provides a list of saved profiles	Profile table in the SQL database	Queries the database, then displays users if they are available	A drop down menu of users to choose from
Guest profile	Allows users to log in without creating a user	N/A	Runs like a normal user with the save button disabled	N/A
Create profile button	Link from the main page to the profile creation page	User mouse click	Sends the user to the profile creation page	N/A
Create profile name text box	Place to enter real name	Full name	Holds text typed by user	N/A
Create profile username text box	Place to enter username	Username	Holds text typed by user	N/A

Component	Purpose	Input(s)	Processing	Output(s)
Create profile submit button	Allows user to submit the new profile to the database	Mouse click	Starts create profile function	N/A
Create profile function	Allows users to create new profiles	Username, name	Adds a new profile to the database, checks for duplicate username, rejects and redirects user back to the creation page. If not the profile is created and the user is sent to the main page	Invalid username / name message, or successful creation message
Load profile function	Gives users access to their saved profiles	User selection from drop down menu	Queries the database, and grabs saved data from training and testing portions of the application	User profile data
SQL Database	Holds profile information, recorded gestures, and test data	User training, testing, requests and creation	SQL statements sent from the main body of code manipulate and read the database	Stored and retrieved data
Training start button	Starts the training sequence	User mouse click	Loads the training portion of the application	Training process signal to start

Component	Purpose	Input(s)	Processing	Output(s)
Testing start button	Starts the testing sequence	User mouse click, training database table	Checks if the user has completed training, if not the button is disabled, else the button begins the testing application	Shows disabled or enabled, sends signal to start testing process
Save profile button	Button that allows the user to save their profile data at anytime	user mouse click, current profile	Starts saving process	Confirmation that the user saved their profile
Clear profile button	Button to allow the user to reset their profile	User mouse click, current profile	Sends the current profile and signal to delete to the clear function	Confirmation of profile being cleared, UI reset

Component	Purpose	Input(s)	Processing	Output(s)
Clear profile function	Clears the designated profile of associated data	Profile to clear	Sends delete statements to the SQL database, erasing all of the data associated with the current profile	Cleared profile
Save profile function	Allows users to store current data	Temporary data collected during the current session	Copies data changes made during the current session to the profile data	Save confirmation, data changes to the profile
Hand recognition	Identifies the position of the users hand	Kinect read in, users hand	Finds the users palm using the Kinect SDK, and tracks its center-point	3D coordinates of the center-point of the users hand
Hand contour recognition	Identifies the shape of the users hand	Kinect read in, users hand, hand location	Using the location of the hand looks for the palm, fingers, and thumb by separating it from the background based on its distance from the Kinect	Set of points shaping the hand

Component	Purpose	Input(s)	Processing	Output(s)
Hand square dimensions	Identifies the height and width of the hand	Set of points from the hand contour recognition	Finds the maximum and minimum x and y values, creates a square based on this	A square with the height and width of the user's hand
Fingertip Recognition	Identifies user's finger tips	Set of points from the hand contour recognition	Uses a K-means Algorithm on the set of the hands coordinates to find the location of the users fingertips	Number of fingers held up, coordinates of fingers
User ready to sign detection	From the user holding their hand open, palm forward in front of Kinect this will determine that the user is ready to sign	Fingertip detection	Once the fingertip detection recognizes five fingers this signals that the user is ready to sign	Signal that the user is ready to sign
Area calculation	Finds the area of the hand	Set of points from the hand contour recognition	Uses points on the hand to calculate its area	Area of the hand in pixels

Component	Purpose	Input(s)	Processing	Output(s)
Percentage of square area	Calculates the percentage of the area of the hand's square dimensions it takes up	Area of the hand, Square dimensions of the hand	Area of the hand divided by the area of its square dimensions	Percentage of the square dimensions the hand occupies
Hand Sign training	Teaches user how to sign, and use the application	Signal to start training, hand recognition	Shows user how to sign certain letters through video demonstration	N/A
Hand configuration	Collect data on how a user preforms hand signs to compare with during testing	Captured hand signs from the training sessions, Percentage of square area, square dimensions after the user has signed	Takes the percentage of the square area, the height and width of the hand while signing, and stores them in the selected profile.	Hand sign data stored to the SQL database
Hand sign testing	Tells the user what letter to sign, and if it was done correctly	Hand sign data from the training sessions, Captured sign during testing, Percentage of square area, square dimensions after the user has signed	Compares the user's attempt at signing a letter they were instructed to and compares it to the average of signs made during training to confirm accuracy	Message stating if it was an accurate or inaccurate sign
Exit button	Allows users to exit at anytime	User mouse click	Sends signal to exit application	Exit signal

N/A

Component	Purpose	Input(s)	Processing	Output(s)
Exit function	Exits the application	Profile save status	Checks if the user has saved their profile if they haven't a pop-up will appear and ask if they would like to save	Pop-up, exited application

3.3 Performance Requirements

The application should not require any network connection and should only have one user at a time. The application needs to be able to recognize hand gestures quickly and accurately so that the user can sign at their own pace. Each user on a system will have their own profile containing files with their configuration and recorded gestures.

3.4 Quality Attributes

3.4.1 Security

N/A

3.4.2 Availability

As user profiles are created in the application, the amount of data stored will increase. In the event where the disk space is filled, the user will need to remove profiles or move the installation to another disk.

3.4.3 Reliability

The data captured of the user's hands must be reliable and consistent in order for the application to compare the hand signs the user makes during testing to the hand signs made during training. Reliability in this area can be increase by adding more data from training to compare against.

3.4.4 Maintainability

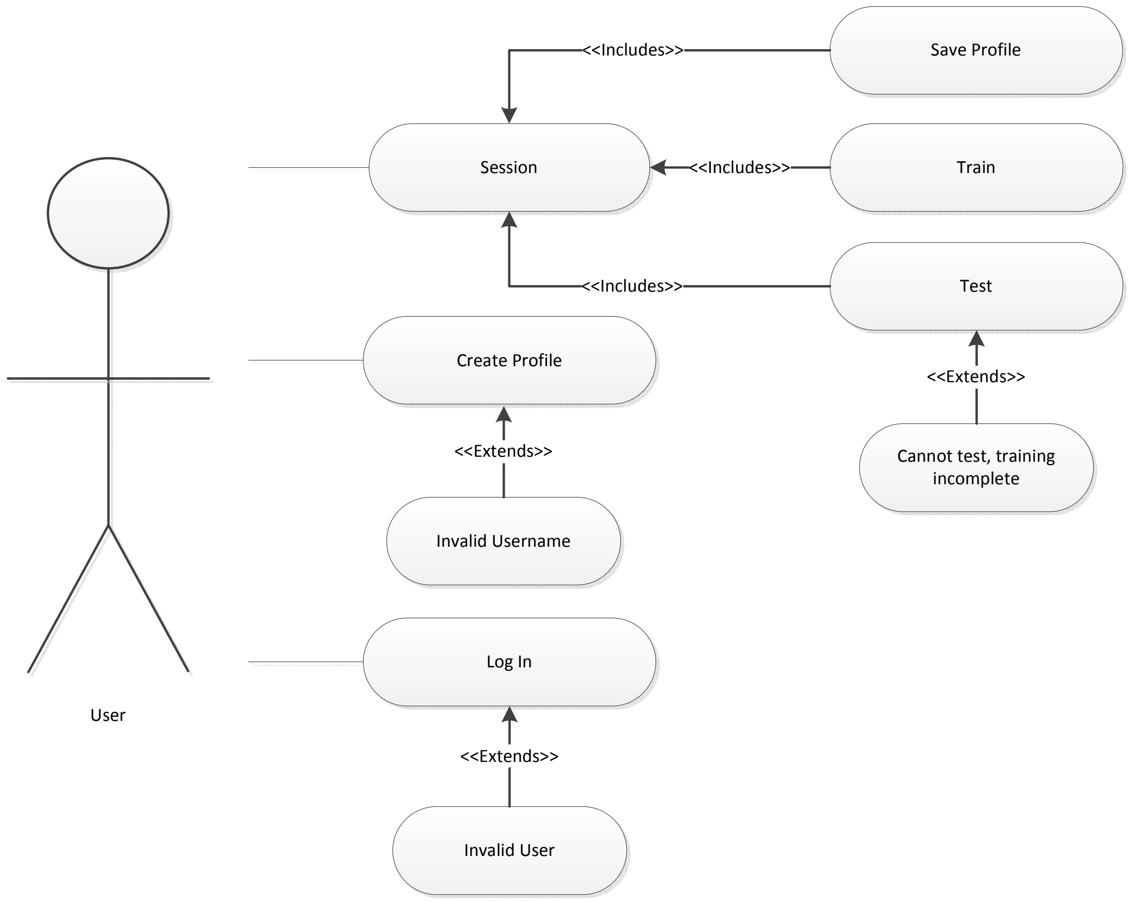
N/A

4 Other Requirements

N/A

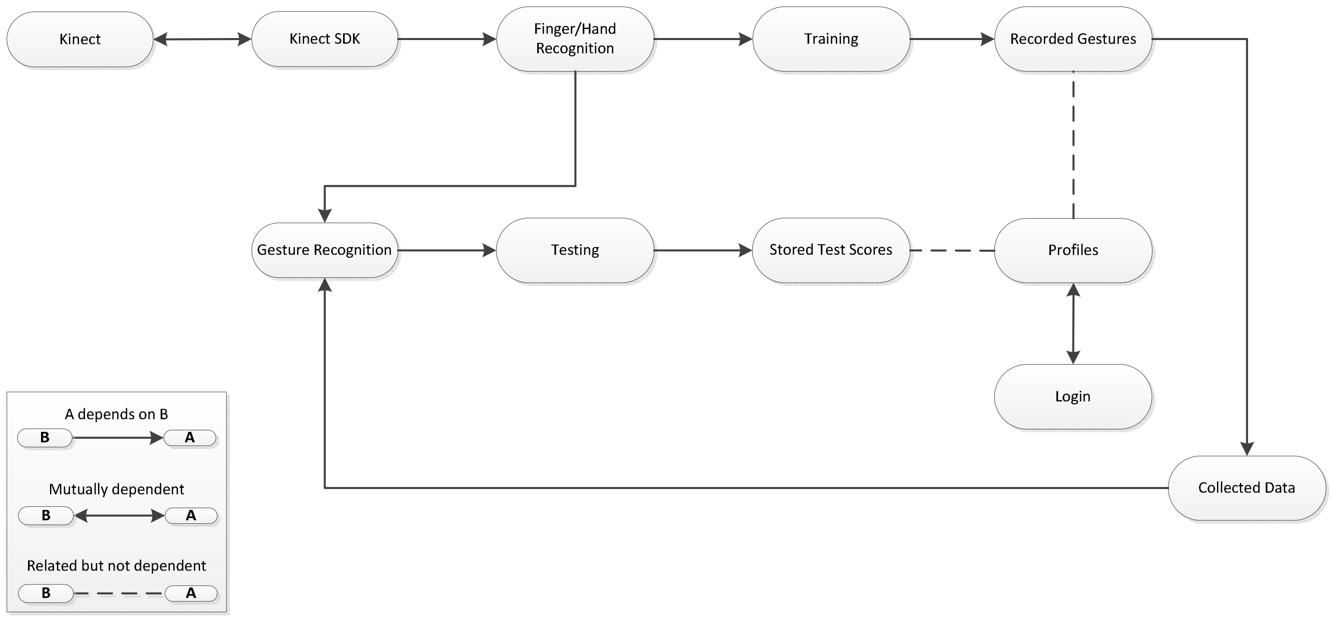
5 Use Cases

The user will be able to log in, create a profile and start a session where they can train, test their skills and save their progress. When loading a profile, if the username does not exist, they are notified that the username entered is invalid. When creating a profile, if the username contains non alphanumeric characters, or if the username already exists, the user will be notified. A user can also log in as a guest. In this case, any progress the user makes during testing cannot be saved.

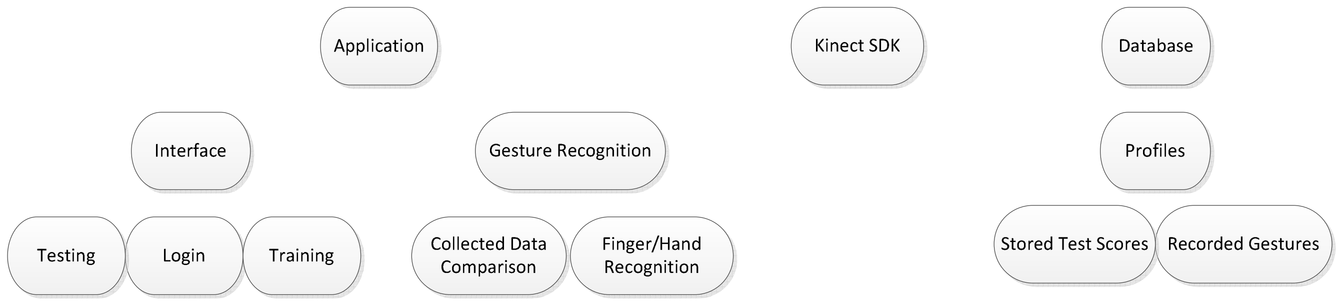


6 Appendix

6.1 Interaction Diagram



6. 2 Component Diagram


Application
Kinect SDK
Database
Interface
Gesture Recognition
Profiles
Testing
Login
Training
Collected Data Comparison
Finger/Hand Recognition
Stored Test Scores
Recorded Gestures