Abstract

This research focused on how Natural User Interfaces cater for handicap people. NUI is the capability to interact with computers in humanistic ways. NUI is powered by touch, gestures, sound and senses. If we look at the human set, it includes normal people as well as handicap people. So, lots of NUI stuff available in the world that cannot use by these special need people.

In the sense of justifying above said problem, dumb or deaf people spends a depressing lifestyle with their situation. Because, they have the mobility unlike blind people or people with physical disabilities. But, they cannot communicate with each other easily. So, they have to face uncomfortable situations in their day today life.

Communication is one of the best parts of our day to day life. Ordinary people communicate with each other easily. Hearing impaired people use special language called as sign language. But ordinary people can't understand sign language.so they face different kind of difficulties when they communicate each other's. Also when sign languages different from county to country. That mean American Sign Language different than British. British sign language different than Sri Lanka.so they have faced infinite difficulties when they communicate each other's. Also they have to limited sign symbols for express their ideas.

For that worldwide Organizations proposed different kind of solutions to fulfill these issues. But still they tried to build complete system. To overcome these barriers they used image processing techniques and used hardware techniques for overcome these barriers. Among these solutions few ones are support to Sinhala sign language.

Researchers and organizations trying to identify sign language. But, I'm focusing to create a prototype which identify sentences and also which can used to implement any sign language. My System provides a very accurate solution to overcome these barriers which are occurred using sign language to communicate. Which capture gestures from the LeapMotion device and convert it to sentences. Also it gives the ability to users to define their own sign language. The accuracy of the system is near more than 95% when we consider the sign symbol identification process.

Acknowledgement

I would like to take this opportunity to pay my gratitude to Dr. Ruvan Weerasinghe, Dean of Informatics Institute of Technology (IIT), for his advices and valuable guidance throughout the project as my supervisor. I also thank the course leader - MSc program, Mr. Guhanathan Poravi and Mr. Mohamed Cassim for been supportive, which enabled me to make my project successful one.

I also thank the principle of deaf school Ratmalana, Mr Christoper, and staff of deaf school Rathmalana, for the valuable advices, which enabled me to make my project successful one.

Table of Contents

Abstract.....i
Acknowledgement.....ii
Table of Contents.....iii
List of Figures.....vii
List of Tables.....ix
List of Abbreviations.....x
Chapter 1: Introduction.....11
1.1 Chapter Overview.....12
1.2 Introduction.....12
1.3 Background.....13
1.4 Aim and Objectives.....14
1.4.1 Aim.....14
1.4.2 Objectives.....14
1.5 Feature of the Prototype.....14
1.6 Communication Model for Hearing Impaired People.....14
1.7 Resource Requirements.....15
1.8 Project Document Structure.....15
1.9 Chapter Summary.....16
Chapter 2: Literature Review.....17
2.1 Chapter Overview.....18
2.2 Introduction.....18
2.3 Natural User Interfaces.....18
2.3.1 Touch Screen.....18
2.3.2 Gestures.....19
2.3.3 Speech Recognition.....20
2.3.4 Gaze Tracking.....21
2.3.5 Brain-Machine Interfaces.....22
2.3.6 Skinput.....23
2.3.7 SixthSense.....24
2.4 Devices Available in the Market.....25
2.4.1 Kinect Device.....25
2.4.2 LeapMotion Device.....26
2.4.3 Myo Arm Band.....27

2.5 Related Works....28
2.5.1 Sign Language Translation – "The sound of singing"....28
2.5.2 Sign Language to Speech Translation System Using PIC Microcontroller....28
2.5.3 Enable Talk System....29
2.5.4 Mimix Speech to sign language real-time translation....31
2.6 Comparison between Existing Systems....33
2.7 Chapter Summary....33
Chapter 3: Project management....34
3.1 Chapter Overview....35
3.2 Project Management Methodology....35
3.2.1 Time Allocation....35
3.2.2 Constraints and Dependencies....36
3.2.3 Risks and Mitigation Plan....36
3.3 Development Methodology....37
3.4 Research Methodology....40
3.5 Chapter Summary....40
Chapter 4: Requirements Specification....41
4.1 Chapter Overview....41
4.2 Requirement Elicitation Process....42
4.2.1 End User Questionnaire....44
4.2.2 Formal Interviews....48
4.2.3 Finding from Requirement Elicitation....49
4.3 Stakeholders....50
4.4 Use Case Diagram....52
4.5 Functional Requirements....56
4.6 Non Functional Requirements....57
4.7 Chapter Summary....57
Chapter 5: System Architecture and Design....58
5.1 Chapter Overview....59
5.2 High Level Design....59
5.2.1 USL System....59
5.2.2 High Level Architecture....60
5.3 System Design....62

5.4 Domain Model of USL System.....63
5.5 Design and Architecture Optimization.....64
5.6 Dependency Diagram.....64
5.7 Entity Relationship Diagram..... 65
5.8 Design Goals.....65
5.9 Chapter Summary.....66
Chapter 6: Implementation.....67
6.1 Chapter Overview.....68
6.2 Technologies.....68
6.3 Implementation of Rest Service.....69
6.4 Implementation of Admin Module.....71
6.4 Implementation of Dashboard Module.....73
6.4.1 Implementation of a View.....73
6.4.2 Implementation of the Logger.....76
6.4.3 Implementation of the Routing 77
6.5 Implementation of Gesture Definer 77
6.6 USL System Implementation.....80
6.7 Chapter Summary.....81
Chapter 7: Testing.....82
7.1 Chapter Overview.....83
7.2 Objectives of Testing.....83
7.3 Testing Criteria.....83
7.4 Functional Requirements Testing.....83
7.5 Module and Integration Testing.....84
7.6 Non Functional Requirements Testing.....85
7.6.1 Accuracy Testing.....85
7.6.2 Performance Testing.....86
7.7 Limitations of the Testing Process.....87
7.8 Chapter Summary.....87
Chapter 8: Evaluation.....88
8.1 Chapter Overview.....89
8.2 Evaluation Criteria.....89
8.3 Evaluators.....89

8.4 Evaluation Methodology……90
8.5 Evolution Survey Questions……90
8.6 Evaluation Survey Findings……91
8.6.1 Overall Concept……91
8.6.2 Scope & depth of the project……91
8.6.3 System Design, Architecture and Implementation……92
8.6.4 Usability Accuracy and Performance……93
8.7 Chapter Summary……93
Chapter 9: Conclusion……94
9.1 Chapter Overview……95
9.2 Achievements of Aim and Objectives……95
9.3 Knowledge from Course Modules……96
9.4 Use of Existing Skills……96
9.5 Learning Outcomes……97
9.6 Problems Faced……97
9.7 Limitations of the Research……97
9.8 Future Enhancements……98
9.9 Summary……99
REFERENCES……a
Appendix A – Admin Module Screen Shots……f
Appendix B – Dashboard Module Screen Shots……k
Appendix C – Test Cases……m
Appendix D – Entity Data Model Diagram……p

List of Figures

Figure 2.1 - Touch Screen 19
Figure 2.2 - Gestures.....20
Figure 2.3 - Speech Recognition.....21
Figure 2.4 - Gaze Tracking.....22
Figure 2.5 - Brain-Machine Interfaces.....23
Figure 2.6 - Skinput.....24
Figure 2.7 – SixthSense.....25
Figure 2.8 - Kinect Device.....26
Figure 2.9 - LeapMotion Device.....27
Figure 2.10 - Myo Arm Band.....28
Figure 2.11 - Sign Language Translation.....29
Figure 2.12 - Enable Talk System 1.....30
Figure 2.13 - Enable Talk System 2.....31
Figure 2.14 - Mimix Speech System.....32
Figure 4.1 - Questionnaire success rate.....44
Figure 4.2 - Age Group Representation.....45
Figure 4.3 - Impaired People Seen Frequency.....45
Figure 4.4 - Likeliness of taking to a hearing impaired people.....46
Figure 4.5 - Literacy of Sign Language.....46
Figure 4.6 - Smart Phone User Access Preference.....47
Figure 4.7 - Preference Admin Site Access.....47
Figure 4.8 - Devices use to access admin site.....48
Figure 4.9 - Onion Diagram for the Proposed System.....50
Figure 4.10 - Use Case Diagram for USL System 1.....52
Figure 4.11 - Use Case Diagram for USL System 2.....54
Figure 5.1 – High Level Diagram.....60
Figure 5.2 – High Level Architecture.....61
Figure 5.3 Domain Model.....63
Figure 5.4 – Dependency Diagram.....64
Figure 5.5 – Entity Relationship Diagram.....65
Figure 6.1 – User Rest service end point.....69
Figure 6.2 – User Logic Layer Implementation.....69
Figure 6.3 – User Logic Layer Interface.....70
Figure 6.4 – User Data Layer Implementation.....70
Figure 6.5 – User Data Layer Interface.....70
Figure 6.6 – Json of User Response.....71
Figure 6.7 – User Controller and Associated Views.....72
Figure 6.8 – User Controller End Points.....73
Figure 6.9 - Client File Structure.....74

Figure 6.10 – Admin Controller……74
Figure 6.11 – Common Activate Function……75
Figure 6.12 – User View……75
Figure 6.13 – Common Logger……76
Figure 6.14 – Route Handler……77
Figure 6.15 – Customize LeapTrainer Gesture Definer……78
Figure 6.16 – Gesture Definer Module……78
Figure 6.17 – Gesture Definer save Interface……79
Figure 6.18 – USL System Customize Gestures Code……80
Figure 6.19 – USL System Interface……81
Figure 7.1 - Accuracy Rate of Gesture Recognition for Common Languages……85
Figure 7.2 - Accuracy Rate of Gesture Recognition for User Language……86
Figure 8.1- Achievements of Nonfunctional Requirements……93
Figure 9.1 – Future USL System……98

List of Tables

Table 1.1 - Software and Hardware Requirements.....15
Table 2.1 - Comparison between Existing Systems.....33
Table 3.1 - Time Allocation.....35
Table 3.2 - Identified Risks for USL System.....37
Table 3.3 - Development Methodology.....37
Table 4.1 - Evaluation of Requirement Elicitation Method.....42
Table 4.2 - Summary of Findings.....49
Table 4.3 - Stakeholders and Roles.....51
Table 4.4 - Use Case Description for User Login.....52
Table 4.5 - Use Case Description for Register User.....53
Table 4.6 - Use Case Description for Add Gesture.....55
Table 4.7 - Use Case Description for Map Gesture to Language.....55
Table 4.8 - Requirement Prioritization Levels.....56
Table 4.9 - Functional Requirements of the System.....56
Table 4.10 - Non Functional Requirements.....57
Table 5.1 - Domain Model Description.....63
Table 7.1 - Tested Functional Requirements.....83
Table 7.2 - Summary of Module and Integration Testing.....84
Table 7.3 - Performance Testing of USL System Using Google Chrome 86
Table 8.1 - Evaluation Criteria.....89
Table 8.2 - Evaluator Groups.....89
Table 8.3 - Evaluation Survey Questions.....90
Table 8.4 - Summary of Evaluation Feedback on Project Concept.....91
Table 8.5 - Summary of Evaluation Feedback on Project Scope and Depth.....91
Table 8.6 - Summary of Evaluation Feedback on Design, Architecture and Implementation.....92
Table 9.1 - Objectives achievements.....95
Table 9.2 - Knowledge from Course Modules.....96

List of Abbreviations

Abbreviation	Definition
USL	Universal Sign Language
NUI	Natural User Interface
ASP	Active Server Pages
UML	Unified Modeling Language
SAPI	Speech Application Programming Interface
API	Application Programming Interface
OAuth	Open Authentication
ORM	Object Relational Mapping
REST	Representational State Transfer
CRUD	Create, Read, Update and Delete
IIS	Internet Information Services
JSON	JavaScript Object Notation
MVC	Model View Controller
UI	User Interface

Chapter 1: Introduction

Content

- Chapter Overview
- Introduction
- Background
- Aim and Objectives
- Feature of the Prototype
- Communication Model for Hearing Impaired People
- Resource Requirements
- Project Document Structure
- Chapter Summary

1.1 Chapter Overview

The purpose of this chapter is to provide an overview of the project. It starts with a brief introduction and then moves to background of the problem. Also defines the aim and objectives of the project to overcome the limitations of sign language systems. Furthermore the high level features of the system are described and the chapter concludes with an overview of the chapters contained in this report.

1.2 Introduction

Natural User Interface (NUI) is the capability to interact with computers in humanistic ways. NUI is powered by touch, gestures, sound and senses. Gestures are the motions of the body which is used to communicate with others. And to make this communication to happen both sender and receiver must know the same set of information for a particular gesture. Today, we are talking about NUI and we develop various things for NUI.

Look at things that are available in the NUI world.

- Touch screen interfaces let users interact with controls and applications more intuitively than a cursor-based interface because it is more direct.
- Gesture recognition systems track user motions and translate those movements to instructions.
- Speech recognition allows users to interact with a system through spoken commands.
- Gaze-tracking interfaces allow users to guide a system through eye movements.
- Brain-machine interfaces read neural signals and use programs to translate those signals into action.

Also, new devices and technologies comes to NUI world every day which are more advance and more humanistic.

- Skinput is an input technology that uses bio-acoustic sensing to localize finger taps on the skin.
- SixthSense is a gesture-based wearable computer system.

People are sick of mouse and keyboard. They want their machine to be a person who close to them. But, most of the time users adapt to designer's choices. For an example, touch based smartphones or tablets comes with designer's ways or choice as NUIs, but users adapt to it. Even the zooming (We zoom using our fingers. But is it the way we used to read small letters of a newspaper or a book. What we did was move our eyes towards the paper or the book)

All these amazing technologies now we have with us, But, I suggest let's think backwards. Person to person is different, we have our own way of actions. The way I move my hand is not the same way that others do. Designers design some NUI things and give it to users saying that these are natural use interfaces that you can use. It may be the way that some users prefer, but some users may prefer totally different thing. For an example, some users build up large databases in spreadsheet programs because they find those are easier to use. Researcher pointing out using examples that people think and express in ways that cannot always be predicted by designers.

New technologies and devices like Kinect, Google Glass, Leap Motion, Corning Gorrilla Glass, Myo band, Eye Tribe, Realsense camera, Oculus devices and etc. There are all these amazing things in to connect with NUI and do amazing things. If we take a look at human set, it is not only the people who have every part of their body. There are squint people, cripple people, blind people, dumb people, deaf people and etc. So, if there are NUIs it can be used by all

these peoples, it will be great. Also, every human not wanted to do the same task like everyone. Each and every man has their own way. So, we are trying to reach humans using NUI, but still we have a question, is this users want?

Human are mainly connected with each other by using the Communication. Communication is an essential part of day today life. Normal people can communicate with each other easily using any language in the world if they are capable of that language. But, for the hearing impaired people communication is a difficult task since they are limited to the sign language. Using sign language they can communicate only with the people who know the sign language. This is a huge barrier to them to express their ideas, to achieve good positions in their lives and due to this matter they are unaware of important news in the world.

Therefore if there is a method which provides a proper way to communicate with hearing impaired people and normal people it will be a great solution to reduce this communication barrier. At such times the need of a system which converts sign language into audio and audio into sign language is needed. So that; those people will be able to communicate with everyone easily.

The hearing impaired people face difficulties when they are trying to communicate with normal people. This is same for the normal people when they communicate with hearing impaired people. The reason is deaf and hard of hearing people cannot hear any voice and normal people do not know the sign language. Therefore their communication is limited among themselves.

The solution come up with is to develop a system which captures the symbols or gestures and convert into convert into an audio. So it helps normal people to understand what they say in sign language.

1.3 Background

Nearly one person among ten people is hearing impaired people. It is one of the major portions in Sri Lanka society. According to survey which has done by the health ministry of Sri Lanka. According to the survey 9% of people are hearing impaired people among 1,800,000 Sri Lankans. It is significant number when compare to total population. Most of hearing impaired people uses Sinhala sign language for their communication. Hearing impaired people face difficulties when they are trying to communicate with ordinary people, because of hearing impaired people cannot hear anything from outside of the world. Not only that but also normal do not have proper knowledge about the sign language. This gap between normal people and the hearing impaired people increase the distance between normal people and hearing impaired people.

Now a day's world is so caring about the hearing impaired people. Most of the hearing impaired people have facilities to learn the sign languages such as American Sign Language, Sinhala sign language and Tamil sign language. It is important to care and provide guidance the child to get familiar with the sign symbols from their childhood after identifying a child as a hearing impaired by born. Otherwise, that child will face lot more troubles while communicating with others. Therefore a special application is required to deliver solutions for the following listed issues.

- To decrease the communication gap among hearing and normal people.
- To identify sentences from hearing impaired people.

- To use sign language applications in the market to identify Sinhala sign language.

To full fill these problems I proposed is a system which identify gestures and convert it to voice. My application provides a solution to reduce the communication gap between normal and hearing impaired people.

1.4 Aim and Objectives

This section mainly focuses about objectives of this project. Mainly project states short term aims and long term objectives.

1.4.1 Aim

The Prime aim is build a system which can reduce the communication gap between ordinary people and hearing impaired people. In this system, it captures the sign symbol and it converts as an audio. Also implement the system in application where users can define gestures and maps those gestures to words.

1.4.2 Objectives

- Carryout a study on Natural User Interfaces available today.
- Research and critically analyses existing NUI application today in the market
- Perform a survey among young crowd what they what as NUIs tomorrow
- Perform a survey among disable people to find out the easy ways for them to interact with computers
- Investigate NUIs for disable people and their capabilities and drawbacks
- Develop the solution as a framework so that others can use it
- Test the prototype using disable people and evaluate their feedback
- Use framework in a prototype to prove the concept
- Identify further enhancements

1.5 Feature of the Prototype

User performs some gestures which matches to a sentence, at the end system will produce the audio output which matches the performed gestures.

1.6 Communication Model for Hearing Impaired People

According to success these objectives we provide a communication model for hearing impaired people. This system provides the benefits for both ordinary people and hearing impaired people. It mainly reduces the communication gap between these two parties and increases the communication capability between these two parties. In this system capture the sign symbol from the system and generates the word string according to the sequence of the sign symbols. Then read the word string by using the system and generate the audio output for the ordinary people.

Users

Mainly focus hearing impaired people and doctors when they are communicating. Not only that after improvement of the system as a product, will close the gap between ordinary people and hearing impaired people.

Input and Output

Inputs to the system divide mainly two categories which are sign symbols and text. Output also divides into two categories as the nearly close to inverse of the input. That mean generated audio to ordinary people and text generated based on gestures that user perform.

1.7 Resource Requirements

Software Requirements	Hardware Requirements
• .Net Framework 4.5.1 • Web API 2 • MSSQL Server 2014 • MySql Server • Intellij Idea 15 • JavaScript • LeapJs • AngualarJs • Microsoft office package • Edraw Max	• Core i5 processor • 6GB DDR3 RAM • LeapMotion Device • Raspberry Pi 3 • Raspberry Pi Display

Table 1.1 - Software and Hardware Requirements

1.8 Project Document Structure

Final document of the project will be containing following chapters.

Chapter 2 – Literature Review

The Literature review chapter will consist of an in depth review carried out to understand the impact the strengths and weakness of various recommender types would have on the accuracy and the scalability of a recommender system, common problems faced by all types of recommendation systems in relation to accuracy and scalability and previous work done to overcome those problems.

Chapter 3 - Project management

Project management chapter will describe about the project management aspect of the project. It will contain the project plan and risk mitigation plans along with a discussion about the most suitable software development methodology for the project

Chapter 4- Requirements Specification

Requirement specification chapter will cover the process undertaken for stakeholder identification, methods adopted for the requirement elicitation from the stakeholders and the processes undertaken to identify the functional and non-functional requirements.

Chapter 5- System Architecture and Design

System Architecture and Design chapter will contain design decisions, design goals, high level architecture of the system along with various design diagram done to model the proposed system.

Chapter 6- Implementation

Implementation chapter will provide a detail description about how each functional and non-functional requirement was implemented, any problems and challenges encountered during the implementation stage and measures that were taken to overcome those challenges and problems.

Chapter 7- Testing

Testing chapter will contain details about the testing phase of the project. Starting from unit level testing it will go until actual prototype feature testing. All the test results gathered from above mentioned testing will be documented and analyzed.

Chapter 8- Evaluation

Evaluation chapter will be having details about the project evaluation. Expert evaluations, self-evaluation and reviews about those evaluations will be available in this chapter.

Chapter 9 – Conclusion

Analyses how the objectives were successfully achieved followed by modules contributed and problems encountered. Eventually future enhancements and concluding remarks are described.

1.9 Chapter Summary

Background and motivation is described in chapter one. Also Chapter one includes the problem in brief, Aim and Objectives which are relevant to this project. Chapter two mainly focus about the Literature review.

Chapter 2: Literature Review

Contents

- Chapter Overview
- Introduction
- Natural User Interfaces
- Devices Available in the Market
- Related Work
- Comparison between Existing Systems
- Chapter Summary

2.1 Chapter Overview

Previous chapter contained a brief introduction in to the project, by presenting the problem domain along with previous work done on the sign language domain, project aim and objectives which are to be achieved for the successful completion of the project. Literature Review chapter will discuss the NUIs available today and devices used to process human actions and related work done on the problem domain. Then the chapter will present the reader with a critical review of the possible techniques, technologies, algorithms, methodologies and various already available tools can be used to develop the proposed USL system in an effective and efficient manner.

2.2 Introduction

The previous chapter mainly focuses about the outlines of the background for this research project and discuss about the proposed solution to address this scenario which is face by normal and hearing impaired people. Not only has that it described the final outcome of the project through the desired aim and objectives. This chapter mainly focuses about the brief description about other systems which are similar projects to our proposed solution. Also functionalities of these products and the design of these products and the way the overcome this communication gap between these two parties. This also presents the differences between these systems and compared the technologies which are used by them. Also compare about the difference between these project when consider our project.

2.3 Natural User Interfaces

Natural User Interface (NUI) is which sounds simple but when dig into deep it become harder or confuse to understand.

2.3.1 Touch Screen

Touch screen is an input device where users can interact with systems using their fingers or a stylus. Actually it's an input device with a display screen which sensitive to pressure. User can interact with the system by touching the objects in the screen. Touch screen interfaces give the ability to control applications more intuitively than a cursor because it is more direct. Nowadays touch screens are popular thing among people. Because, users get familiar with mobile and tablets they want the same experience on desktop and public venue displays as well. Touch screens can provides a fast and more natural way of interactions for users.

Touch screen give control to the user by simple or multi-touch gestures with a stylus or one or more fingers. Some touch screens need special coated gloves to work. Touch screens gives the ability to interact directly with what is on the display, other than using keyboard or mouse. Touchscreens are common in devices like cell phones, game consoles, laptops, tablets, electronic voting machines, and etc. Touch screens also reduce space and complexity.

There three types of touchscreens.

Resistive – Resistive touchscreen is coated with a thin layer of electrically conductive layer that change the current when touch happened. That registered as a touch event and send the signal for processing. It give 75% efficiency and the thin metallic layer can be damaged by sharp objects. But, resistive touchscreens are not damage or affected by dust or water.

Surface wave – Surface wave technology use ultrasonic waves. Waves pass over the screen panel. When touch happens wave get changed. This change use to get the position of the screen where the touch happened and send the information for processing. This is the most advanced type among the three types of touch screens. But, they get damaged by outside objects.

Capacitive – Capacitive screens coated with a material that stores electrical chargers. When the screen is touched the circuit detect the touch and send the information for processing. We have to use our finger to interact with capacitive touch screens. Capacitive touch screens gives clear display and they are not affected by outside objects.

Figure 2.1 – Touch Screen

2.3.2 Gestures

Gesture recognition systems identify user movements and translate those movements to instructions. Actually it is a mathematical representation of human movements by a machine. Gestures can be made using any body part but, mostly using face or hand. When we using gestures, we have to consider the history of human communication. Because, there is no language that can communication without gesture.

Gesture recognition, facial recognition, gaze tracking, lip movements tracking and voice commands recognition are components of PUI (Perceptual User Interface). The reason why developers and companies are consider about PUI is to give user more natural more easy experience with computers. So, they don't have to get used to mouse and keyboard.

When it comes to personal computing, gestures are used as inputs for programs. Nowadays, some laptops shipped with gesture recognition devices like LeapMotion. Using gestures as inputs for computing in day today file, allows physically impaired people to interact with more natural ways. And when it comes to gaming, it is more effective

and more natural. There are social challengers other than technical ones in gesture recognition. Gestures must be universally acceptable, simple, known by everyone and etc.

Figure 2.2 - Gestures

Gesture recognition enables humans to interact with the computer more naturally.

2.3.3 Speech Recognition

Speech recognition is the capability of a program or a computer to identify spoken words or sentences and convert them to machine readable format. It allows user to use voice commands as input to a system. Most speech recognition software only identify limited number of words and only identify them if the speaker spoken very clearly. And also, there are cases where the software cannot identify different accents. Software engineers and electrical engineers develop technologies to recognize and translate spoken commands into texts.

Speech recognition (SR) adopted by applications to give user a natural and easy way to use them. Gaming commands, call routing, speech-to-text, voice dialing, search the web, simple data entry, prepare structured documents and voice search are some areas where SR been used these days.

Figure 2.3 - Speech Recognition

SR systems implemented in two ways. One is speaker independent systems, where no training needed and speaker dependent systems, where individual speaker reads text or vocabulary into the system. The speaker dependent systems analyze the speaker's voice and tune the system according to speaker.

Voice recognition refers to identify the speaker, not what they say. Voice recognition helps to simplify the speech recognition. It also can be used to authenticate or identify the speaker as a login to a system or as a security process.

2.3.4 Gaze Tracking

Gaze-tracking interfaces allow users to guide a system through eye movements. Eye tracking is the process of electronically locating the point of a person's gaze, or following and recording the movement of the point of gaze. Various technologies exist for accomplishing this task. Some methods involve attachments to the eye, while others rely on images of the eye taken without any physical contact.

One of the earliest applications of eye-tracking was enabling computer access for the disabled. A device that pinpoints the gaze point on a computer screen can allow a quadriplegic to operate that computer by 'pointing' with the gaze and 'clicking' by blinking the eyelids or staring at a certain point on the screen for a length of time, thereby obtaining the functionality of a mouse. An on-screen keyboard and numeric keypad can allow for text typing and calculations, and continuous eye-movement tracking can allow the user to draw diagrams or create graphs.

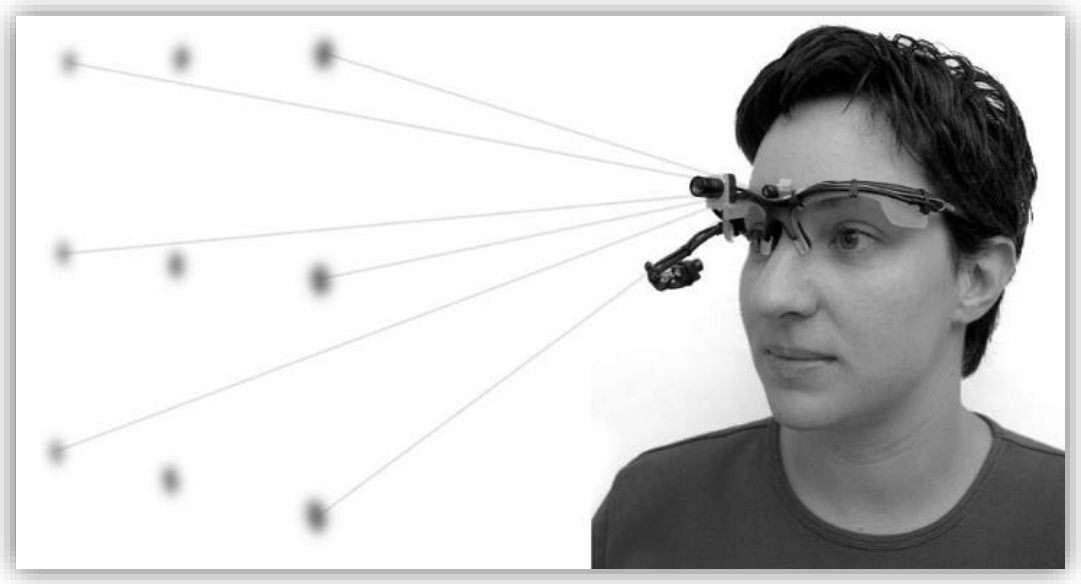


Figure 2.4 - Gaze Tracking

2.3.5 Brain-Machine Interfaces

Brain-machine interfaces read neural signals and use programs to translate those signals into action. A brain–computer interface (BCI), sometimes called a mind-machine interface (MMI), direct neural interface (DNI), or brain–machine interface (BMI), is a direct communication pathway between a brain and an external device. BCIs are often directed at researching, mapping, assisting, augmenting, or repairing human cognitive or sensory-motor functions.

Brain-computer interface is a collaboration between a brain and a device that enables signals from the brain to direct some external activity, such as control of a cursor. The interface enables a direct communications pathway between the brain and the object to be controlled. In the case of cursor control, for example, the signal is transmitted directly from the brain to the mechanism directing the cursor, rather than taking the normal route through the body's neuromuscular system from the brain to the finger on a mouse.

By reading signals from an array of neurons and using computer chips and programs to translate the signals into action, BCI can enable a person suffering from paralysis to write a book or control a motorized wheelchair or prosthetic limb through thought alone. Current brain-interface devices require deliberate conscious thought, some future applications, such as prosthetic control, are likely to work effortlessly. One of the biggest challenges in developing BCI technology has been the development of electrode devices and/or surgical methods that are minimally invasive. In the traditional BCI model, the brain accepts an implanted mechanical device and controls the device as a natural part of its representation of the body. Much current research is focused on the potential on non-invasive BCI.

Figure 2.5 - Brain-Machine Interfaces

2.3.6 Skinput

Skinput is an input technology that uses bio-acoustic sensing to localize finger taps on the skin. When augmented with a pico-projector, the device can provide a direct manipulation, graphical user interface on the body. Skinput represents one way to decouple input from electronic devices with the aim of allowing devices to become smaller without simultaneously shrinking the surface area on which input can be performed. While other systems, like SixthSense have attempted this with computer vision, Skinput employs acoustics, which take advantage of the human body's natural sound conductive properties.

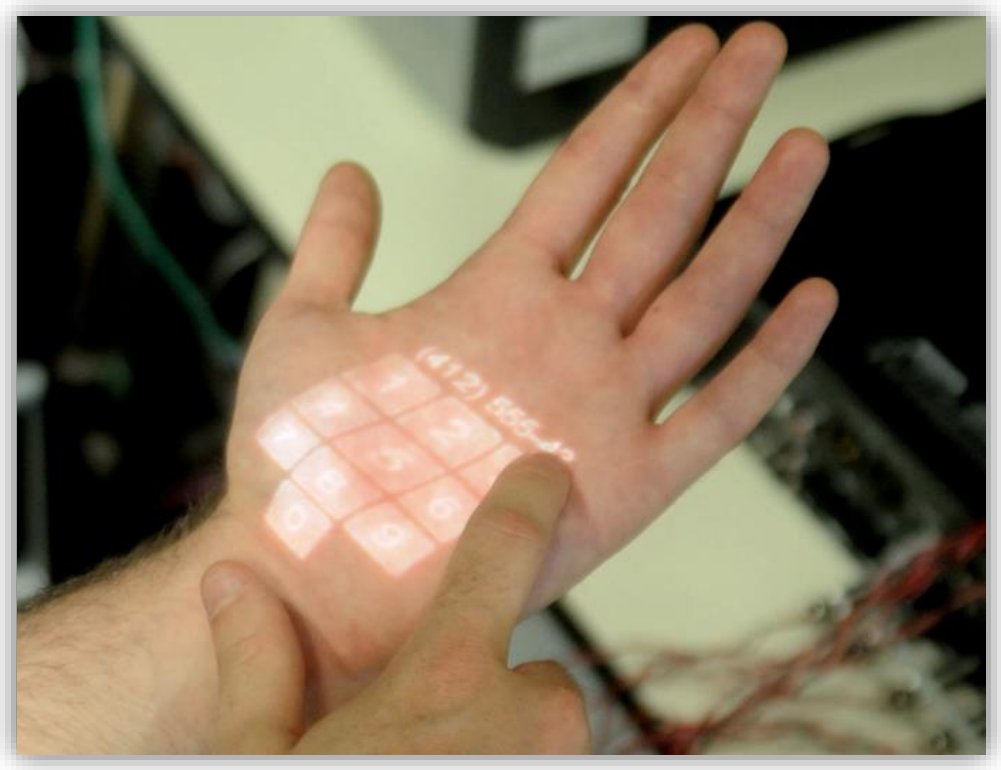


Figure 2.6 – Skinput

2.3.7 SixthSense

SixthSense is a gesture-based wearable computer system. SixthSense is a wearable gestural interface that augments the physical world around us with digital information and lets us use natural hand gestures to interact with that information. Although the miniaturization of computing devices allows us to carry computers in our pockets, keeping us continually connected to the digital world, there is no link between our digital devices and our interactions with the physical world. Information is confined traditionally on paper or digitally on a screen. SixthSense bridges this gap, bringing intangible, digital information out into the tangible world, and allowing us to interact with this information via natural hand gestures.

Figure 2.7 – SixthSense

2.4 Devices Available in the Market

There are various devices in the market for to capture and identify gesture, gaze and etc.

2.4.1 Kinect Device

The innovative technology behind Kinect is a combination of hardware and software contained within the Kinect sensor accessory that can be added to any existing Xbox 360 or to a computer. The Kinect sensor is a flat black box that sits on a small platform, placed on a table or shelf near the television you're using with your Xbox 360. Newer Xbox 360s have a Kinect port from which the device can draw power, but the Kinect sensor comes with a power supply at no additional charge for users of older Xbox 360 models. For a video game to use the features of the hardware, it must also use the proprietary layer of Kinect software that enables body and voice recognition from the Kinect sensor.

- Color VGA video camera - This video camera aids in facial recognition and other detection features by detecting three color components: red, green and blue. Microsoft calls this an "RGB camera" referring to the color components it detects.
- Depth sensor - An infrared projector and a monochrome CMOS (complementary metal-oxide semiconductor) sensor work together to "see" the room in 3-D regardless of the lighting conditions.
- Multi-array microphone - This is an array of four microphones that can isolate the voices of the players from the noise in the room. This allows the player to be a few feet away from the microphone and still use voice controls.

A further look at the technical specifications for Kinect reveal that both the video and depth sensor cameras have a 640 x 480-pixel resolution and run at 30 FPS (frames per second). The specifications also suggest that you should allow about 6 feet (1.8 meters) of play space between you and the Kinect sensor, though this could vary depending on where you put the sensor.

The Kinect hardware, though, would be nothing without the breakthrough software that makes use of the data it gathers. Leap forward to the next page to read about the "brain" behind the camera lens.

Figure 2.8 - Kinect Device

2.4.2 LeapMotion Device

The Leap Motion Controller is a motion-sensor for your computer think of it as a tiny Kinect that works with a Windows PC or Mac. The Controller tracks your hands and all 10 fingers, plus joints in 3D space, with far more precision than you'd expect: up to 1/100th of a millimeter accuracy, according to Leap Motion.

Put it down in front of your computer, in front of the keyboard or your laptop, and the space above it becomes a zone where you can use your hands to control stuff on your computer. Its sensory field, however, is limited to a narrow dome that extends above and around the tiny unit 2 feet above the controller, 2 feet wide, and 2 feet deep 8 cubic feet. It effectively covers a good chunk of your desktop space between you and your computer screen.

The device itself is a tiny black rectangular box smaller than a mouse rimmed in aluminum with a rubberized base. Under its smooth, glossy top are three infrared sensors and two cameras that do all the tracking. It has a little connector port for its included USB cable, and a small green LED light on the front that lights up when it's plugged in.

Figure 2.9 - LeapMotion Device

2.4.3 Myo Arm Band

Touch my thumb to my index finger and the music on my computer stops playing. I bend my wrist outward and the song changes. This concept of controlling interfaces and devices with gestures isn't a new one. Movies like "Minority Report" and "Iron Man" have popularized the idea, but none of the products available today work quite like those in the movies.

You may already be familiar with Microsoft's Kinect sensor, which lets gamers use gestures to navigate the interface of the Xbox 360 and Xbox One. There's also the Leap Motion, a device that offers similar gesture capabilities on Windows and Mac computers. But these products share a common problem, they can't read your hand motions if you move away from the sensor. That's because the Leap Motion and the Kinect both use cameras to identify gestures.

Canadian startup Thalmic Labs came up with a unique approach with its Myo armband. Rather than cameras, the band uses low-energy Bluetooth and a collection of sensors to read the muscle movements within your forearm. The band can recognize pinches, wrist turns and other hand motions that it then translates into specific controls on your computer and smartphone.

I've used the Myo for the past month to control Netflix, PowerPoint presentations and Spotify. It's an interesting idea with a ton of potential, but it still has its kinks. There were just some gestures that simply didn't work, and ultimately it's still a niche product that a majority of consumers can get by without. This is especially true when you factor in the price. The Myo will run you $199. That's more than double the price of the LeapMotion and $50 more than the Kinect for the Xbox One.

So, among these devices LeapMotion is the affordable device for my application. But, the best device for this application is the Myo arm band.

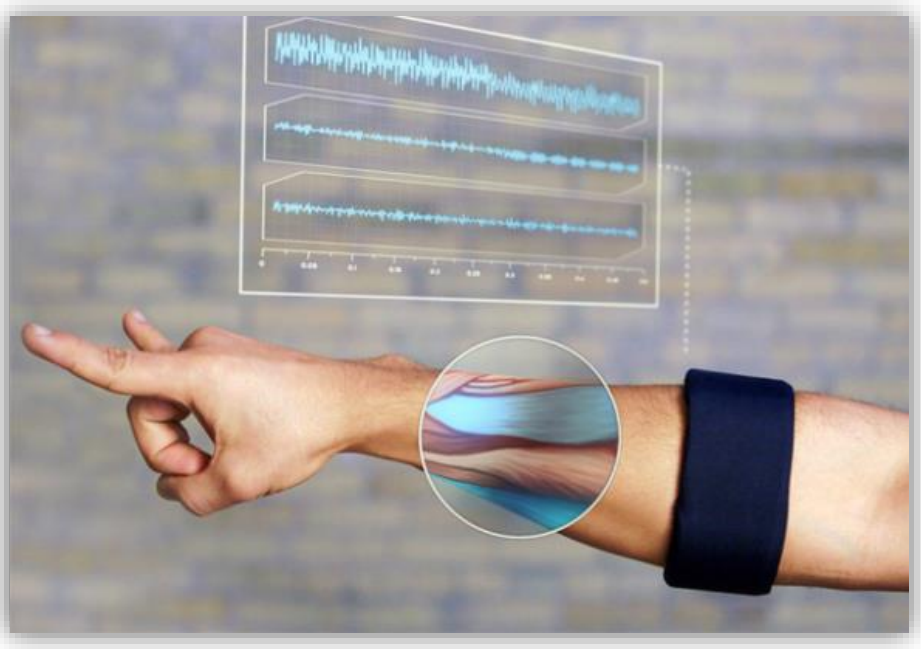


Figure 2.10 - Myo Arm Band

2.5 Related Works

Before select the project we searched the systems which are tries to reduce the communication gap between ordinary people and hearing impaired people. We mainly focus their actions which are taken to solve these issues. Result of that we found the below mention studies.

- Sign Language Translator – "The sound of signing"
- Sign Language to Speech Translation System Using PIC Microcontroller
- EnableTalk System
- Mimix speech to sign language real-time translation

2.5.1 Sign Language Translation – "The sound of singing"

This is another implementation of the sensors based system. This also fitted glove with the sensors. They mainly focus 26 English letters in American Sign Language. This is implemented in Team of engineers in USA.

In this glove also uses contact sensors, flexes sensors and accelerometers in three dimensions and gathers the hand motion and each finger position. These data are relevant to English 26 letters. After translate the gathering data it transmitted to base station, which is used to display the letter and pronounces the letter by using computer interface.

Also there is a game for check the user's ability to sign and it helps to sign language education.

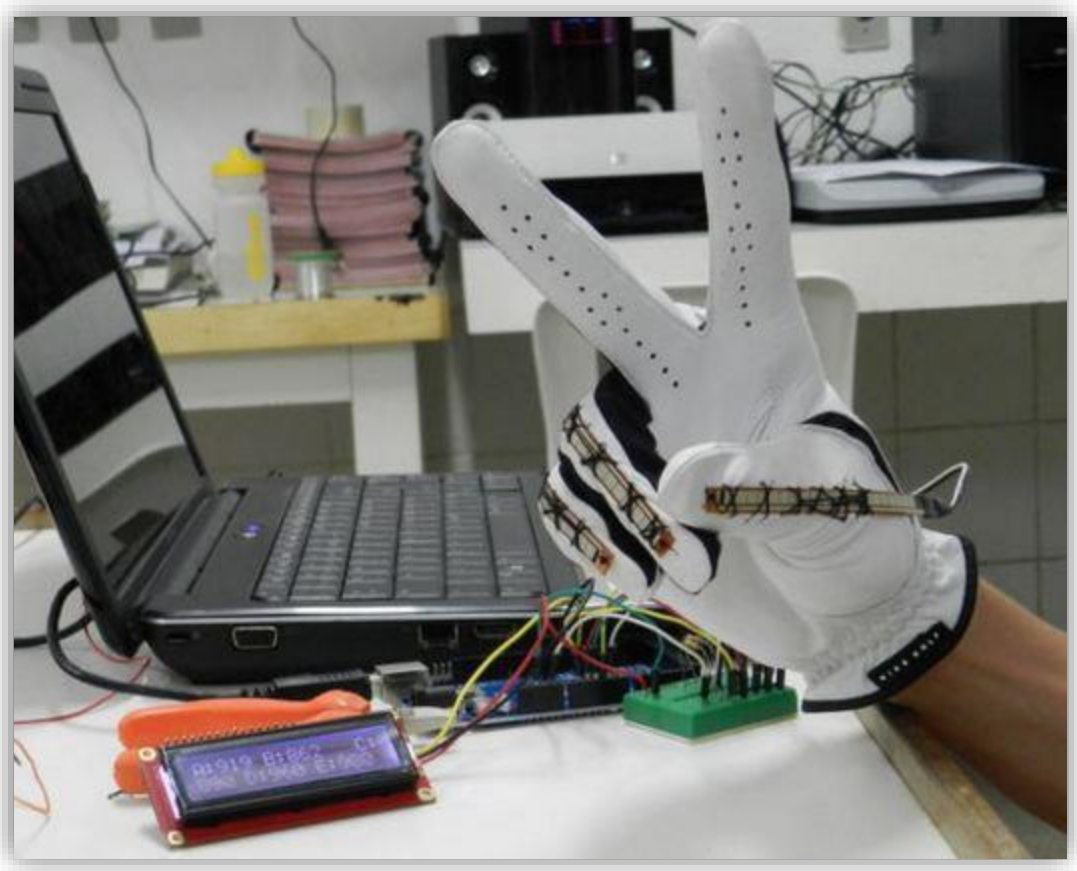


Figure 2.11 - Sign Language Translation

2.5.2 Sign Language to Speech Translation System Using PIC Microcontroller

This system mainly based on pic Microcontroller. This mainly focuses to assistance to dump people to reduce the communication gap between ordinary people. Also they mainly focus about facilitate the life cycle of these peoples. In this system senses the sign language and play the recorded video clips.

This system mainly uses APR9600 with PIC16F877A and flux sensors. All the flux sensors are fitted into a glove and by using these gloves capture the sign symbol. This system contain sensing unit, processing unit, voice storage unit and wireless communication unit. The result is forward to microcontroller by using a circuit.

2.5.3 Enable Talk System

When we considering this system they build this for Microsoft imagine cup 2012 and they won the first price. A team of Ukrainian researches called 'quadSuad' are build this system to the world for extending the communication and bring down the barrier of the communication between hearing impaired people. This is the hardware based system and they capture the sign symbol using hardware and processing using software.
For that they include a glove with sensors. The glove can sense the movement of the hand and finger according to wearer's movement and generate the output according to the hand of finger movement.

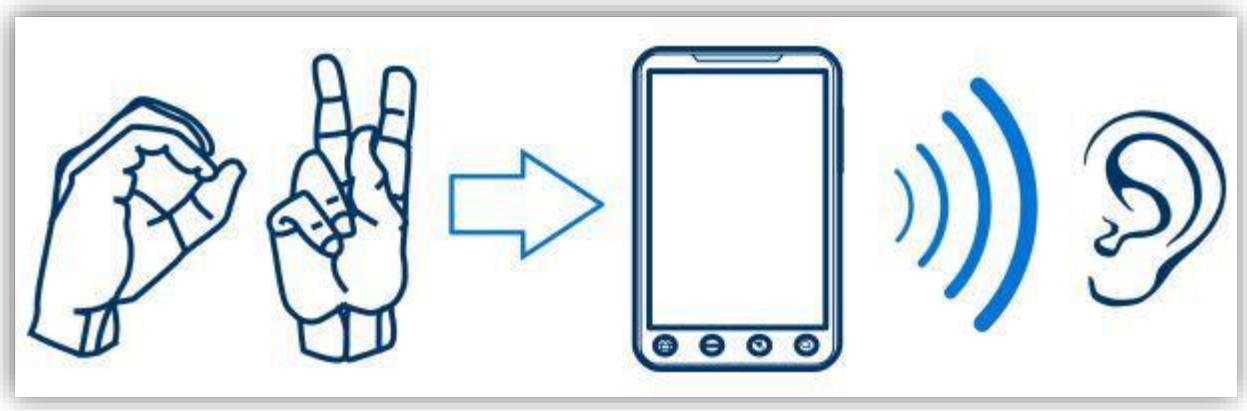


Figure 2.12 - Enable Talk System 1

For success this task they use the special glove with the sensors. This system produces the result with 90 percent accuracy. They include 11 flex sensors into their glove two in each finger and one belong to palm of the glow as well as They use a gyroscope, a compass, two accelerometers and a Bluetooth module. This Bluetooth module transfers all the gathered data by the gloves from the user signs symbols to a mobile device. All the data processing by using the microcontroller and send the output via Bluetooth connection.
They implement this system by using Microsoft API and Bing API created for windows phone 7.

Also they use the rechargeable battery to the glove. Battery can recharge by using the USB cable. Also they use solar power system to extend the interval between charging.

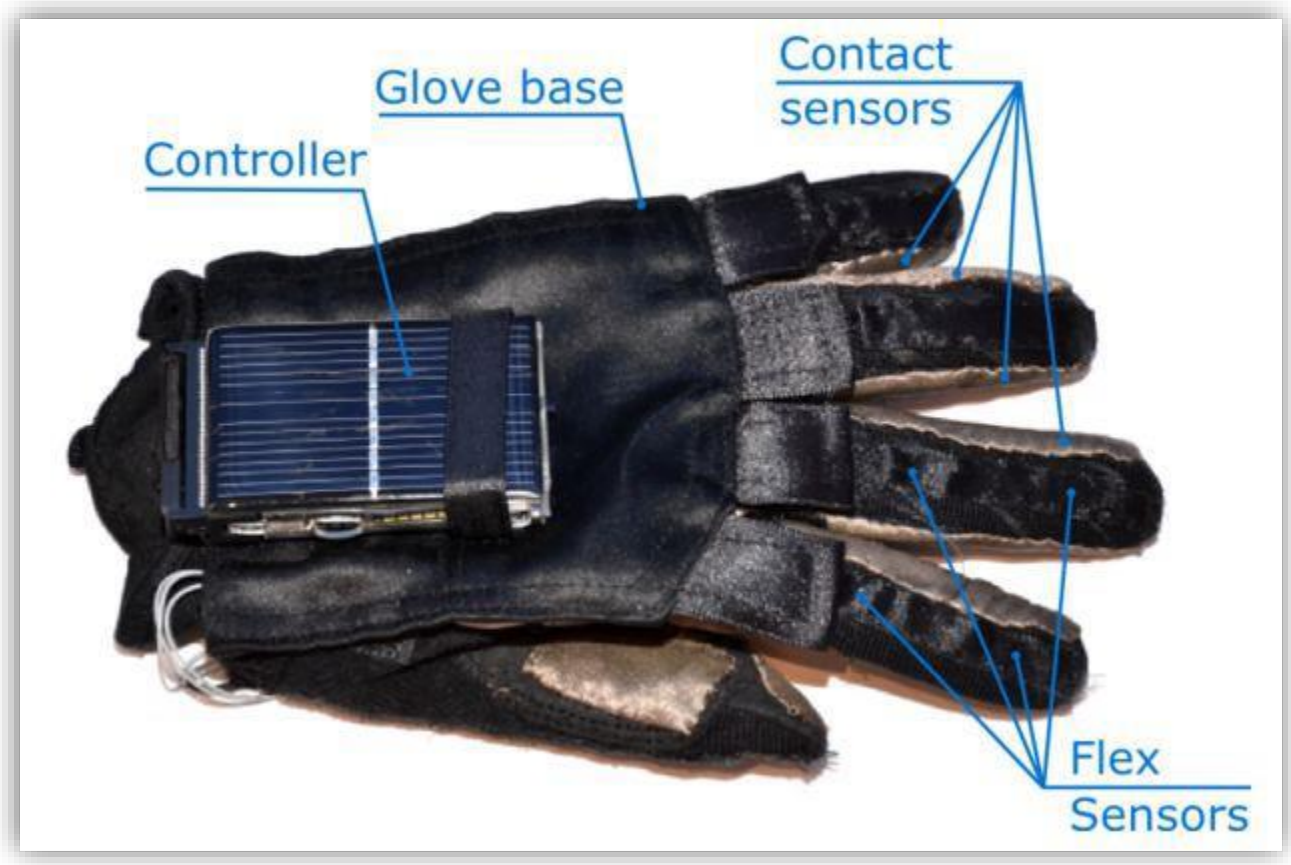


Figure 2.13 - Enable Talk System 2

This system still can translate small phrases and they have to build up relevant library for the system. They mainly focus to native signers and hearing impaired college students. Also they have to increase the processing speed. Anyway this is one the best model which is implemented in this area.

2.5.4 Mimix Speech to sign language real-time translation

This is the mobile application which is used transform spoken and written words into sign language by using 3D avatar. Their interpretation based on ASL (American based sign language) and interpreted by using 3D as signed.

In the real time it translated the speech into sign language clear with two way communication. This is main based for hearing impaired people or deaf people who use the sign symbols. So this application tries to help both deaf and hearing people communicate.

They mainly provide mainly three features such as speech into sign language, text into sign language and text into voice. A team of Jordan and Lebanon implement this system for increase the primary education in the hearing impaired persons in their country.

They use Microsoft technology such as Dot Net framework, Microsoft SAPI, MS windows phone CTP, MS Silverlight, MS expression design for implementation the system. Now their Android Application also available in the Google Play store.

Figure 2.14 - Mimix Speech System

2.6 Comparison between Existing Systems

Feature	Approach 1	Approach 2	Approach 3	Approach 4
	EnableTalk System	Sign Language Translator- The sound of signing	Mimix	Sign Language to Speech Translation System Using PIC Microcontroller
Sign language to Speech Conversion	√	√	×	√
Speech to Sign language Conversion	×	×	√	×
Interpretation of the words/sign symbols	√	×	√	×
Sinhala language support	×	×	×	×

Table 2.1 - Comparison between Existing Systems

2.7 Chapter Summary

In this chapter discusses about the existing technologies and existing system which are similar to our project. Also find out the solutions which are used for achieving the goal. Also use the comparison table to illustrate the points easily. The next chapter is used to discuss the technology adoption of the project and discuss about the technologies which are used in this project.

Chapter 3: Project Management

Contains

- Chapter Overview
- Project Management Methodology
- Development Methodology
- Research Methodology
- Chapter Summary

3.1 Chapter Overview

Previous chapter discussed about literature related to problem domain and this chapter discussed about information related to project management process of the project. It will start from identifying a suitable project management methodology and moving on to time and resource allocations of the project. Then it will discuss about the possible risks and appropriate risk management plans. Finally discussion about selecting a software development methodology for the project.

3.2 Project Management Methodology

Any project would have scope, time and cost as the constraints of the project and it is important to manage the above mentioned constraints properly to produce a quality output from the project undertaken. In a research project, managing those constraints going to be harder compared to an industrial project due to frequent requirement fluctuations, which highlights the importance of a suitable project management methodology. PRINCE2 is a widely used project management methodology which encompasses the high level management, control and organization of a project. It was decided to adopt PRINCE2 as the project management methodology due to the previous exposure and experience with it.

3.2.1 Time Allocation

Following table 3.1 represents an overview of the identified main tasks of the (Refer appendix A for complete Gantt chart). Research projects require the literature review process to be carried till the end of testing phase due to frequent requirement fluctuations of research projects. Subsequently the documentation of project will too spread throughout the project lifetime as it will be the ultimate deliverable of this exercise

No	Task	Duration	Start Date	End Date
1	Initial research	22 days	05 Jan 2016	26 Jan 2016
2	Preparation & Submission of TOR	49 days	27 Jan 2016	15 Mar 2016
3	Literature review phase	153 days	16 Mar 2016	15 Aug 2016
4	Requirement specification phase	77 days	05 Mar 2016	20 May 2016
5	Designing phase of the system	19 days	10 Apr 2016	28 Apr 2016
6	Implementing phase of the system	163 days	01 May 2016	10 Aug 2016
7	Testing phase of the system	62 days	15 Jun 2016	15 Aug 2016
8	Critical evaluation phase of the system	17 days	05 Aug 2016	22 Aug 2016
9	Preparation & submission of the final document	292 days	04 Feb 2016	22 Nov 2016

Table 3.1 – Time Allocation

3.2.2 Constraints and Dependencies

The success of the project, will depend on following constraints and dependencies.

- Time constraint – A major challenge for the successfulness of the project is the whole software development life cycle has to be carried out individually.
- Lack of prior knowledge – Due to lack of knowledge on gesture recognition and gesture recognition devices system may delayed

3.2.3 Risks and Mitigation Plan

Risk 1	Failure to keep with the up-to-date domain knowledge and technologies				
Risk Factor	High	Occurrence Frequency	High	Impact	High
Description	Gesture recognition and sign language implementation is a very active area where new ideas and concepts are brought forward on a regular basis making difficult to keep it up to date knowledge of the domain. Surrounding technologies may change over the project time schedule. This will be a risk that the system may not be developed using the most up to date domain knowledge nor using the latest technologies.				
Mitigation	• Check for any new work done on the domain area on a weekly basis • Check for technology changes on a weekly basis • Keep frequent communication with domain experts about the latest developments				
Risk 2	Not being able to achieve the expected accuracy level for gesture recognition				
Risk level	High	Occurrence Frequency	High	Impact	High
Description	Due to time constraints and changes to requirements and technologies the developed system may not be able to achieve the expected accuracy rates for its users				
Mitigation	• Try to keep up with the latest developments • Finish the prototype early to have enough time to do changes for accuracy related things				
Risk 3	Lack of community support for setting up the language				
Risk level	High	Occurrence Frequency	High	Impact	High
Description	If the system not get the recognition, then people may not give their support for it. So, the system may not able to achieve its goals like setting the system for global usage.				
Mitigation	• Speared the news among community that disable people needs help and each and every one can help them • Use social media to speared the news, so that lots of people get influenced				
Risk 4	Community may miss use or corrupt the system				
Risk level	High	Occurrence Frequency	High	Impact	High

Description	Same users may configure the language incorrectly. This may results in bad credits for the system and disable people may not use the system				
Mitigation	• Keep track of users who configure the language and worn them if they miss use the system • Review language with the help of language professionals				
Risk 5	Not have the required hardware				
Risk level	High	Occurrence Frequency	High	Impact	Low
Description	Hardware that we are using to recognize gestures may not use by everyone or not popular with the community. So, users have to specifically buy the hardware in order to use the system or train the system				
Mitigation	• Using popular hardware in the market • Using affordable hardware • Give users the opportunity to configure or train the system as much as can without the hardware				

Table 3.2 - Identified Risks for USL System

3.3 Development Methodology

A development methodology is a framework that is used to structure, plan and control the process of developing an information system. In order to avoid schedule and cost overruns and to mitigate frequent requirement changes of the project a suitable development methodology has to be adopted. With the aim identifying a suitable development methodology, key characteristics of several software development methodologies are evaluated. Following table shows the characteristics of several frequently used software development methodologies and the below evaluation is based on the available resources, nature of the project and ability to meet the identified requirements.

Development Methodology	Pros	Cons	Characteristics
Waterfall Model	• Easy to understand and use • Simple enough to handle due to the rigidity of the model • Saves significant amount of time • Allows for easy testing and analysis • Easy to arrange tasks	• Not applicable for maintenance projects • Does not allow changes to the project in the testing phase • No option to know possible outcome of a project • Not excellent for long and ongoing projects	• Top down development with independent rigid phases • Linear module • Requirements should be known in the upper hand • Heavy documentation • Step by step process

Prototype Methodology	• Increased user involvement • Gives clear idea about the functional process of the software • Reduces the risk of failure in a software functionality • Reduces time and cost since the defects can detected earlier • Quick user feedback • Identify missing functionality easily	• Chances of extension in management cost • Excessive involvement of client can affect processing • Too many changes affect the workflow of the software • Insufficient requirement put dependency on prototype • Users may get confused the prototype and actual system	• Developed based on currently known requirements • Higher rate on customer interaction
Rapid Application Development	• Changing requirement can be handled • Get feedback from customers for improvement • Makes the development process effortless • Progress can be measured • Reduced development time • Quick initial reviews • Integration from very beginning solves a lot of integration issues	• Dependent on strong team members for identifying business requirements • Requires highly skilled developers • High dependency on modeling skills • Management complexity is high • Requires user involvement • Inapplicable to cheaper projects	• Iterative development • Software prototyping • Fast development with minimum planning and high quality systems
Agile Methodology	• Realistic approach to software development • Functionality can be developed rapidly • Promotes teamwork • Easy to manage • Flexible to developers	• Risk of maintainability and extensibility • Difficult to adapt to new team members • Very high individual dependency • Heavily depend on customer interaction	• Iterative & incremental development • Agility for rapidly changing • Accelerate delivery • Customer interaction

Spiral Methodology	• Risks are reduced • Excellent for large and complex projects • Requirements can be captured more accurately • Allows to add functionality • Suitable for high risk projects	• Management is complex • Process is complex • Excessive documentation • End of project not known in the early stage • Might get continued and never ends • Not ideal for low risk projects	• Iterative development • Continuous refinement of the final software product • Emphasizes risk analysis

3.3 Development Methodology

Waterfall method is suitable for projects with a stable set of requirements but in a research project by nature requirements change during each phase of the software development life cycle. Since linear model with rigid phases doesn't support requirement fluctuations of experimental and research development, traditional waterfall approach is not suitable for a research project. Therefore, it is evident that this project needs to select a methodology where it has more iterative approach than the waterfall methodology. When considering the Rapid Application Development methodology, it can be seen that it uses minimal planning and then quickly moves in to prototype development at an early stage of the project duration. It is evident that it is a must to have a certain level of understanding of the final products' functionalities to start developing the prototype at an early stage. But in a research project functionalities of the prototype can't be identified with a higher degree of certainty at the start of the project and it can be concluded that this methodology is not suitable for a research project due to the constraint of not knowing the functionalities of the prototype in an early of the project. As listed in the table 3.11, Agile methodology uses iterative and incremental development strategy there will be lot of testing, customer interactions and discussions throughout the development phase which may result in time overrun. Additionally Agile methodology is more suited to handle projects with projects with massive scopes and larger number of stakeholders on various levels of an organization. Since the project undertaken has a major constraint on the time availability and neither involved with a huge scope nor a larger number of stakeholders it can be concluded that agile methodology isn't suitable to handle the project in hand. Similar to the Rapid Application Development methodology Prototype methodology too focus on moving into the development of the prototype at an early stage of the project and poses the same problem like in the Rapid Application development methodology as the functionalities of the prototype can't be determined at an early stage of a research project. Additionally it may increase the complexity of the system and may lead to scope enhancements of the system due to the higher rate of customer interaction leading to time overruns. Therefore it can be deduced that Rapid Application Development methodology isn't suitable to conduct the project undertaken.

The Spiral methodology supports continuation of cycles of the software development life cycle without clear termination conditions which would help to counter frequent requirement fluctuations which would lead to timely delivery of the prototype. Additionally the Spiral methodology helps to analyses the risks that would affect the final product and finding those risks in advance can lead to speed up the development process. As the major challenges

of the project in hand are the time constraint and frequent requirement fluctuations it can concluded that Spiral methodology would be the most suitable development methodology.

3.4 Research Methodology

As highlighted by Dubois and & Gadde (2002) a research can be categorized in to two categories known as inductive researching and deductive researching. Deductive approach is aimed at proving and testing a hypothesis while the inductive approach is concerned with generation of new theory emerging from data. The project in hand falls in to the deductive researching approach as the aim of the project is to prove **USL** system can solve the limitations mentioned in the introduction chapter.

3.5 Chapter Summary

This chapter pointed out the importance of a suitable project management methodology to get the best out of project. PRINCE2 was chosen as the most suitable project management methodology due to previous experience and exposure with it. Then project was broken down to tasks and allocation time based on the work to be done. So the work can continue without dependencies. Then identify the constraints and appropriate mitigation plans were discussed. Then identify a suitable development methodology for the project, several methodologies were evaluated and decide to adopt the spiral methodology for development. Then brief about research methodology and adopt deductive research methodology due to the reason that the project was based on proving the hypothesis of creating a universal sign language system is more suitable. The next chapter is the requirement specification, it will discuss about the entire requirement engineering aspect of the proposed project.

Chapter 4: Requirements Specification

Contains

- Chapter Overview
- Requirement Elicitation Process
- Stakeholders
- Use Case Diagram
- Functional Requirements
- Non Functional Requirements
- Chapter Summary

4.1 Chapter Overview

The previous chapter discussed about project management and risks involved in the project. This chapter starts with presenting the process taken to gather information from stakeholders for the proposed **USL** system. Next it dig deeper into requirement analysis models carried out to gather and identify requirements of the system. Lastly it discusses how the project scope will be refined and restructured to accommodate the identified requirements.

4.2 Requirement Elicitation Process

In order to ensure efficiency of the requirement gathering process and counter the limitations associated with each requirement elicitation method several approaches were utilized parallel to gather requirements from various stakeholders. Table 3.1 provides an overview of the factors that lead to the selection of each utilized requirement elicitation method.

Method No 01	Literature Review
Literature review carried out on the new gesture identification devices, several sign language implemented systems and limitations of those systems to identify problematic areas concerning accuracy and scalability of a sing language implemented system	
Advantages	• Identify ways that make implementation of the system easy • Get a better understanding about currently available systems • Identify limitations of available systems to come up with a better solution • Identifying certain areas that have be included in the questionnaire and interview process for better clarifications
Disadvantages	• Due to delays in publications literature material may not present the latest developments and limitations of the related domains • Time consuming effort due to the time and effort required review the vast amount of literature available
Method No 02	Questionnaire
A questionnaire focusing on identifying the end user requirements for a sign language system was prepared. Since sing language system is interested by doctors and hearing impaired people, hard copies of the questionnaire along with an online survey was made available.	
Advantages	• Time saving method compared to other elicitation methods • Ability to cover more areas, more communities • Ease of compression the results due to standardization of questions
Disadvantages	• Success of the gathered data depends on the honesty of the participants

	• May restrict participants from sharing additional information due to standardization of the questions • Difficulties faced when understanding answers given to open ended questions
Method No 3	Formal Interviews
A series of formal interviews were carried out with the sign language users with the aim of identifying what are missing in the current sign language systems along with their suggestions for a new sign language system	
Advantages	• Ease of gather information and clarifying doubts with experts while the interview is going on • Allows to elicitation of miss leading requirements and guidelines for implementation
Disadvantages	• Inability to interview wider audience due to time taken to conduct an interview • Subjective interpretation of the problem by the experts
Method No 4	Observations of end user operations
Several observation sessions of users using sign language system in different environments were conducted to identify the difficulties faced by the users of sign language system with relation to use same set of gestures.	
Advantages	• Direct insight to the limitations faced by the users • Ability to identify new limitations that may not identified in the requirement gathering stage
Disadvantages	• Participant behavior may be affected by observer presence
Method No 5	Self-evaluation
Several self-observation sessions in accessing sign language system in different environments were conducted to self-identify the difficulties faced by users of the sign language system.	
Advantages	• Direct insight to the limitations faces by the users • Ability to identify new limitations that may not identified in the requirement gathering stage • Can be used to validate the identified requirements
Disadvantages	• Certain requirements may be ignored due to subjective to self-experiences

Table 4.1 – Evaluation of Requirement Elicitation Method

4.2.1 End User Questionnaire

Structure of the questionnaire

The questionnaire distributed among the potential end users of the system contained questions to identify,

- End user expectation in a sign language identification system – to understand end user expectation and currently what are the systems that they use
- User behavior patterns in a sign language identification system – to understand where are most needed areas
- Want is the way preferred for a sign language identification system – to understand the easiest way for uses to access the system

Limitations of Questionnaire Process

- Limited response - Though the questionnaire was made available to a larger number of potential end users of different levels of the society, not all of the questionnaire were returned back and substantial amount of the returned were not adequately completed. Therefore the gathered responses may not represent the absolute view of all the end users.
- Feedback credibility - Since the questionnaires were filled by individuals at their own free will there is no way to authenticate the credibility of the feedbacks given. Therefore the gathered responses may not represent the absolute view faced by the end users.

Requirement elicitation of questionnaire

$$Success\ Rate = \frac{\text{Compledte Questionnaires}}{\text{All Questionnaires}}$$

$$Success\ Rate = \frac{119}{144}$$

$$Success\ Rate = 82.64\ \%$$

Figure 4.1 - Questionnaire success rate

According to the above Figure 4.1 the successful response rate was above 82% for the questionnaire. So, it can concluded that the questionnaire process was successful.

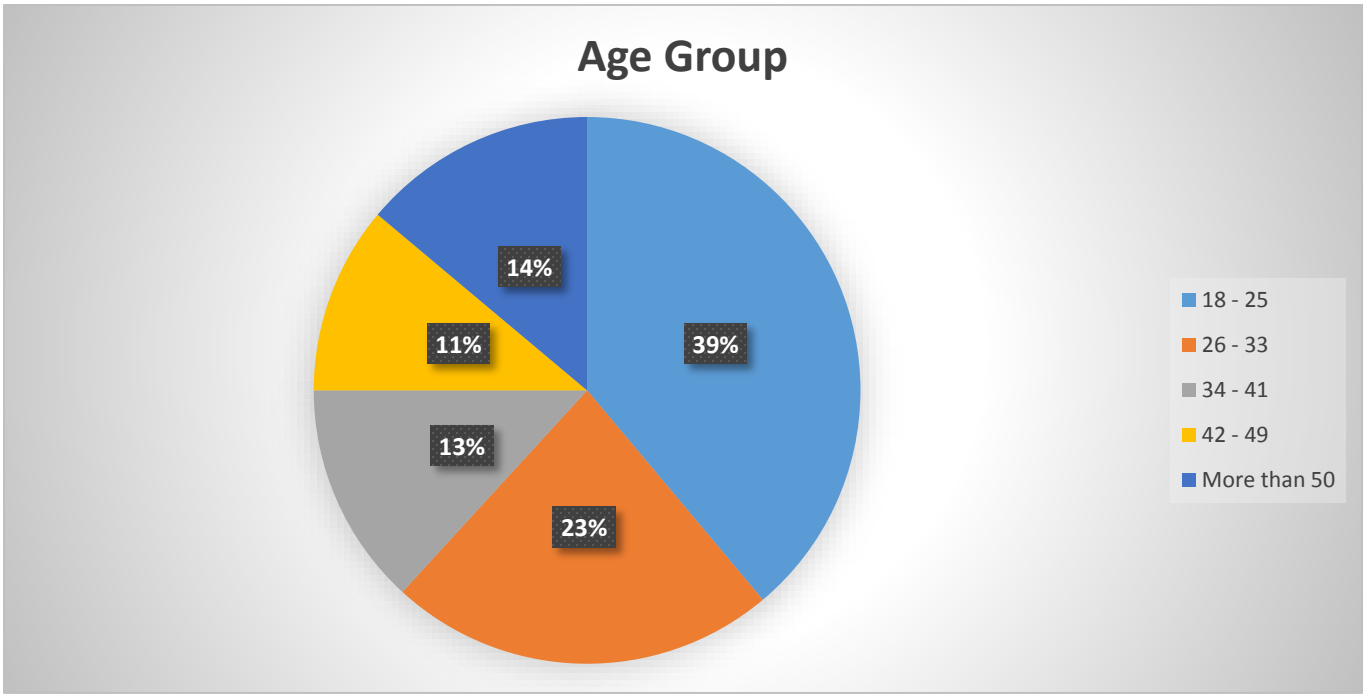


Figure 4.2 - Age Group Representation

The above Figure 4.2, highlights that the questionnaire was mostly answered by the participants of 18-25 age group with a 39% contribution along with 26-33 age group with 23% and more than 50 age group with 14% contributions. These statistics shows that the questionnaire has managed to gather requirements from all age groups more than 18.

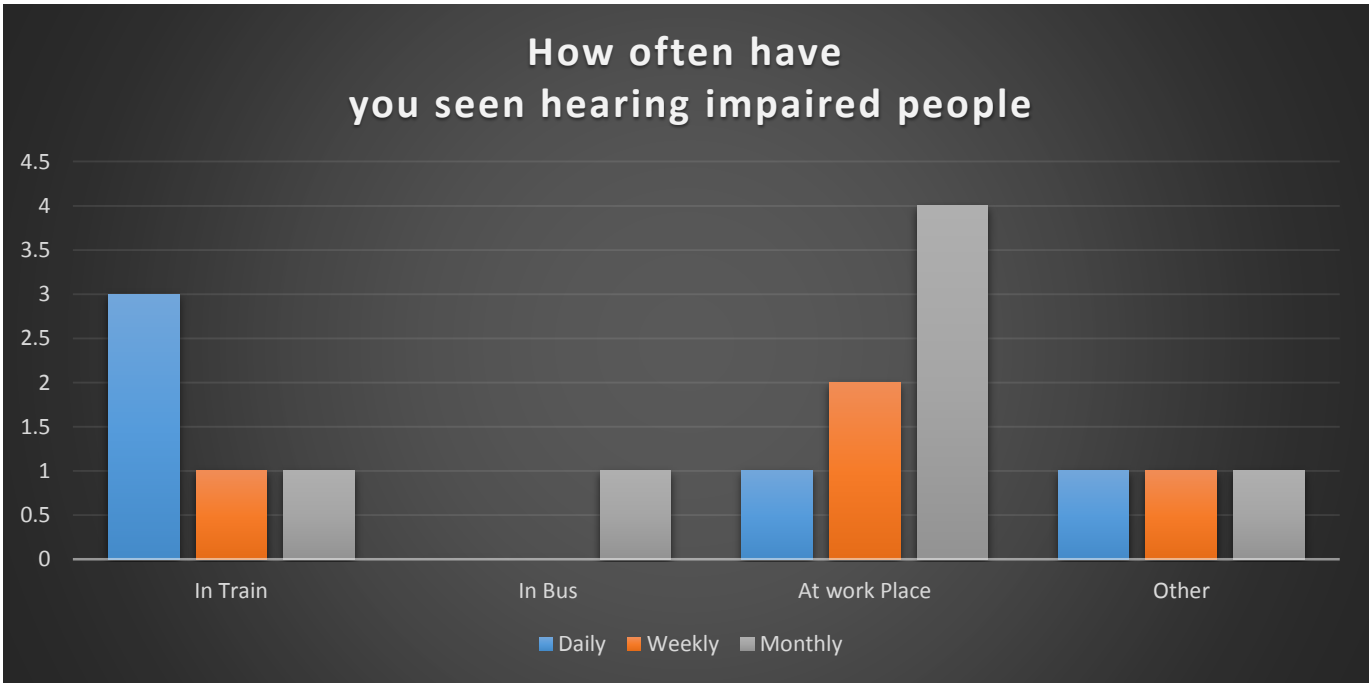


Figure 4.3 - Impaired People Seen Frequency

According to the results shown on Figure 4.3 it can be concluded that, at work place seen frequency is greater than the train seen frequency. Therefore, the results show that there is a likeliness of seen a hearing impaired person higher at work place and in train is higher than other places.

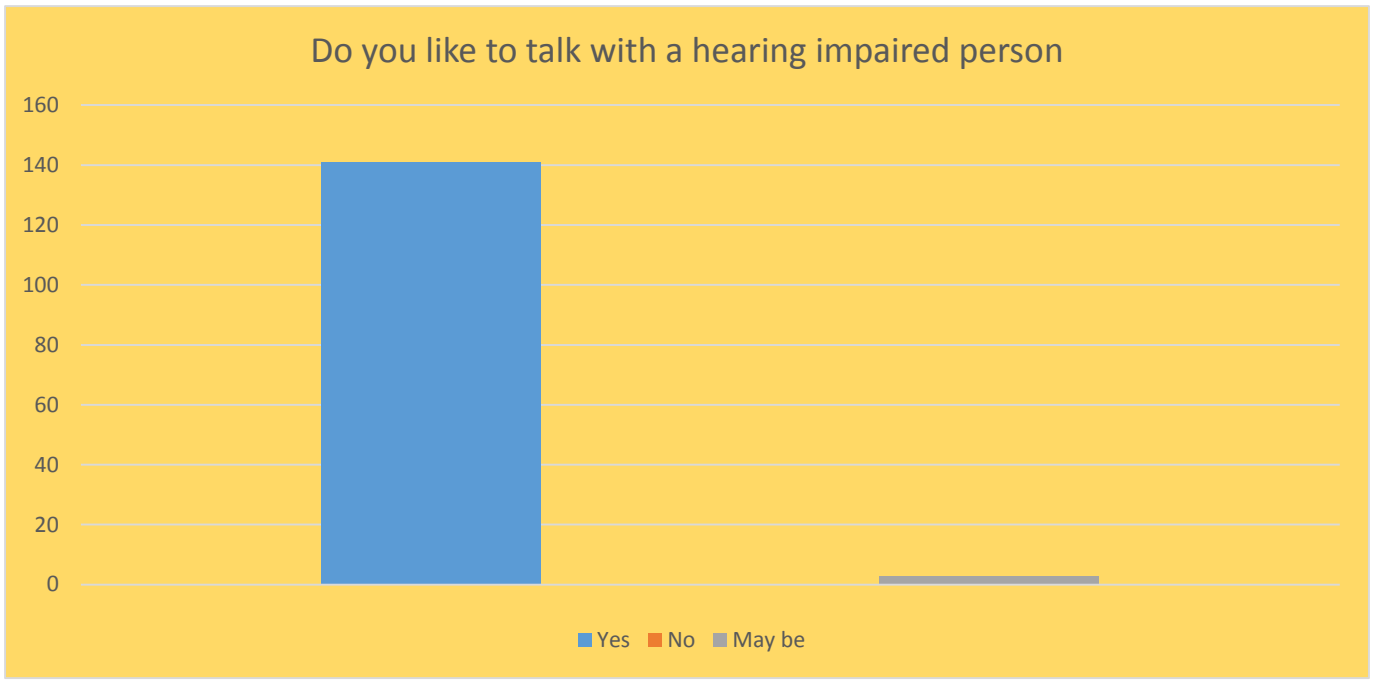


Figure 4.4 - Likeliness of taking to a hearing impaired people

According to the chart, it shows that people are willing to talk with hearing impaired people. No one said “no” to talk with a hearing impaired person. So, it shows that normal people are willing to talk with them.

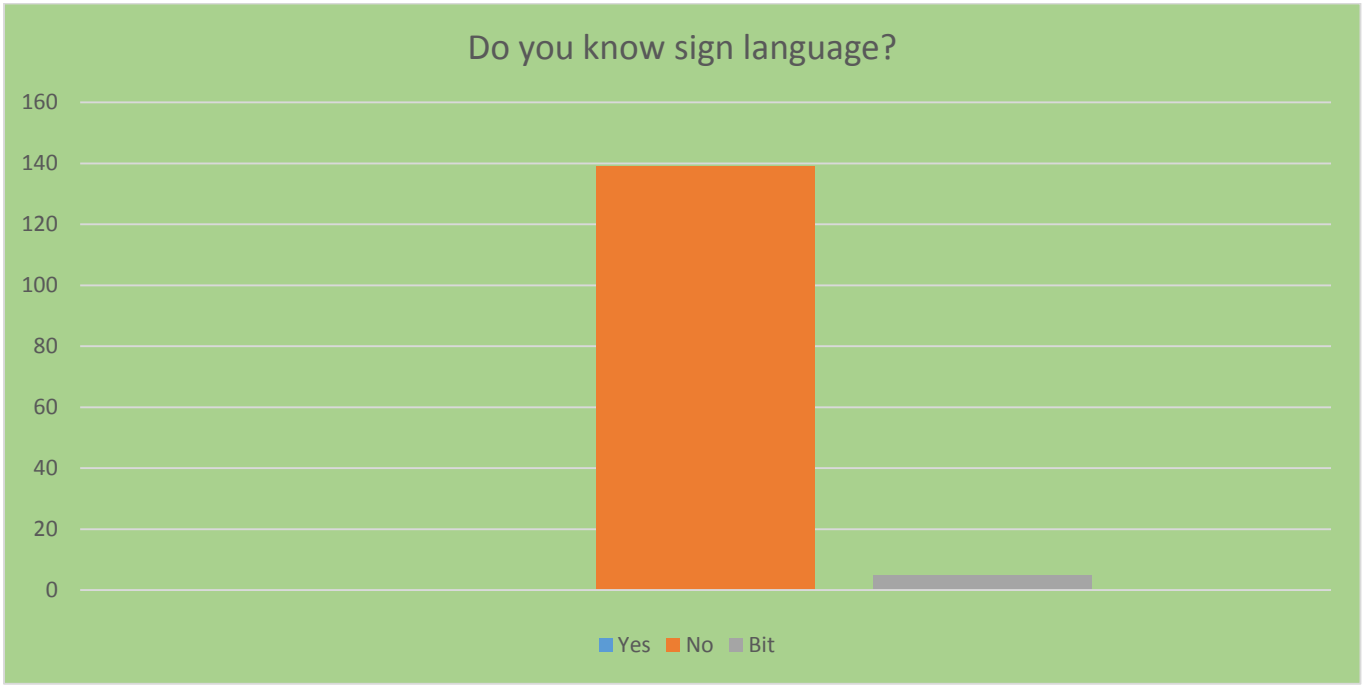


Figure 4.5 - Literacy of Sign Language

Chats shows that no one have the ability to communicate with sign language. Few people know bit of signs, but surely they don't have the ability to use sign language to communicate with a hearing impaired person.

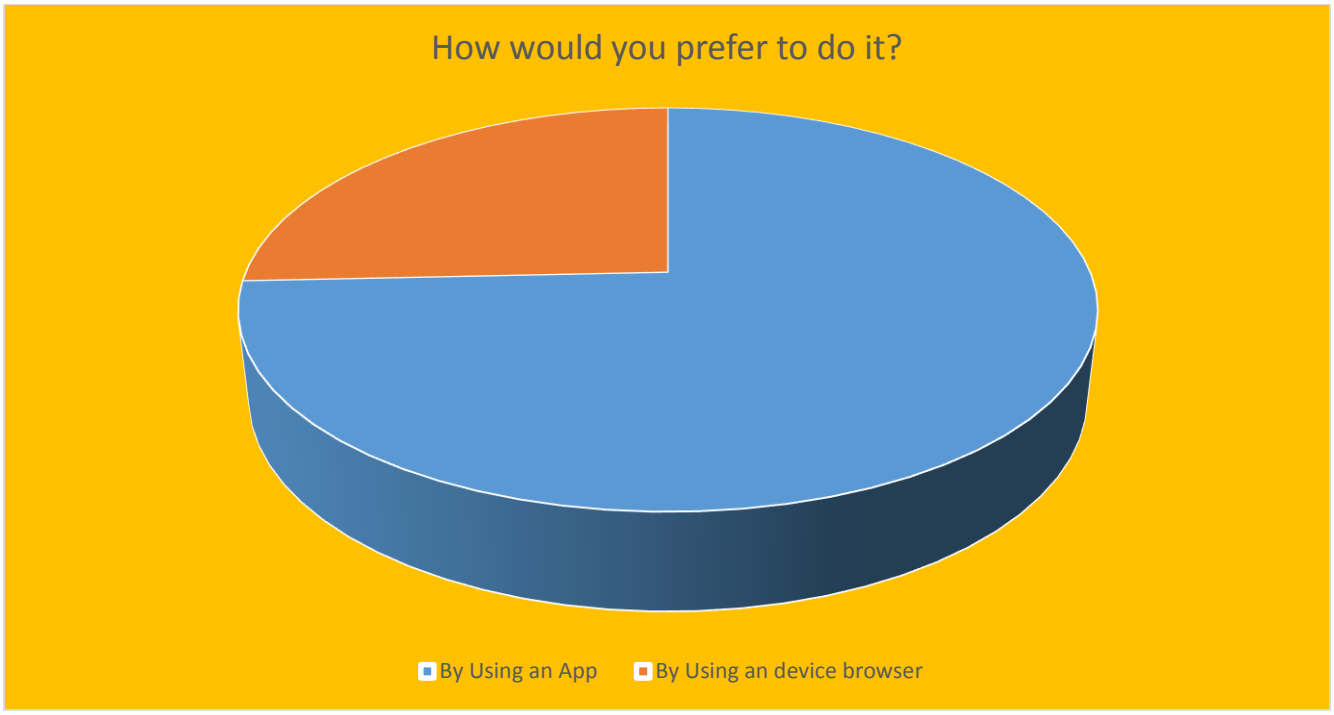


Figure 4.6 - Smart Phone User Access Preference

As shown in the figure 4.6, it can be clearly seen that the readers prefer to access online news site through an application, rather than using a default browser when using a smart phone

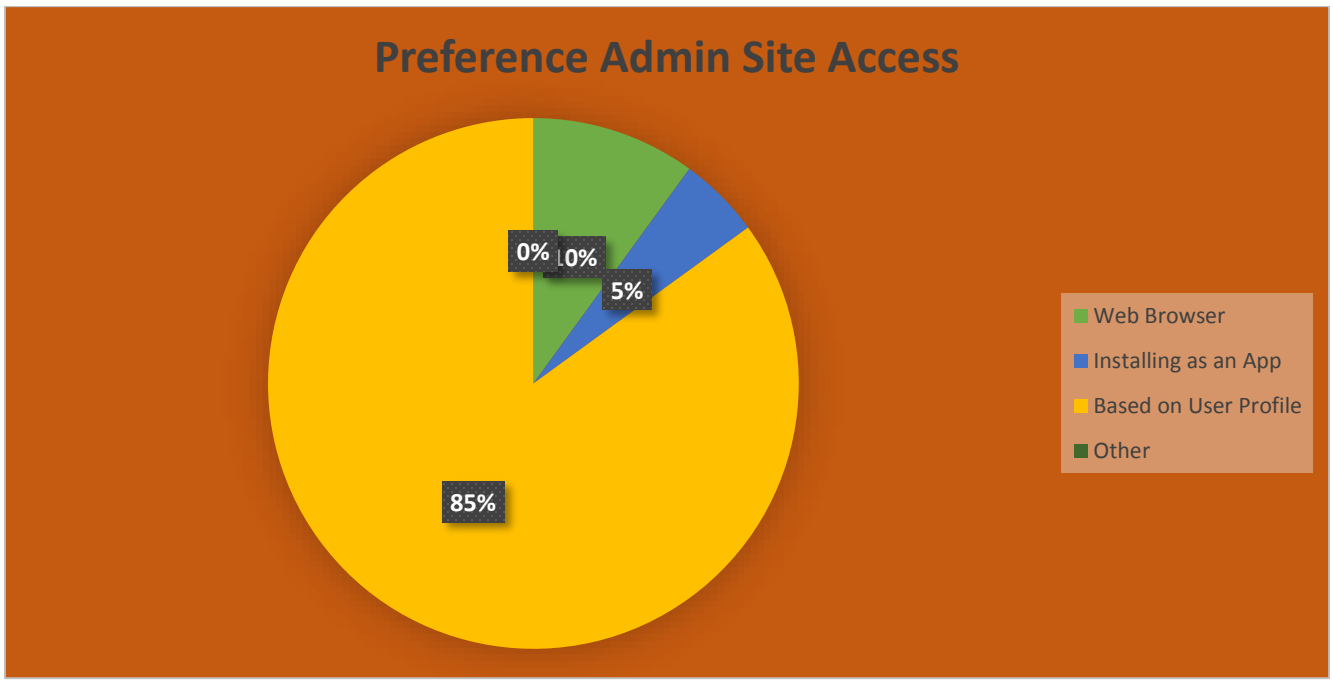


Figure 4.7 - Preference Admin Site Access

As per figure 4.7 it can be seen that the best option is to access admin site in sign language system is through the user profiles.

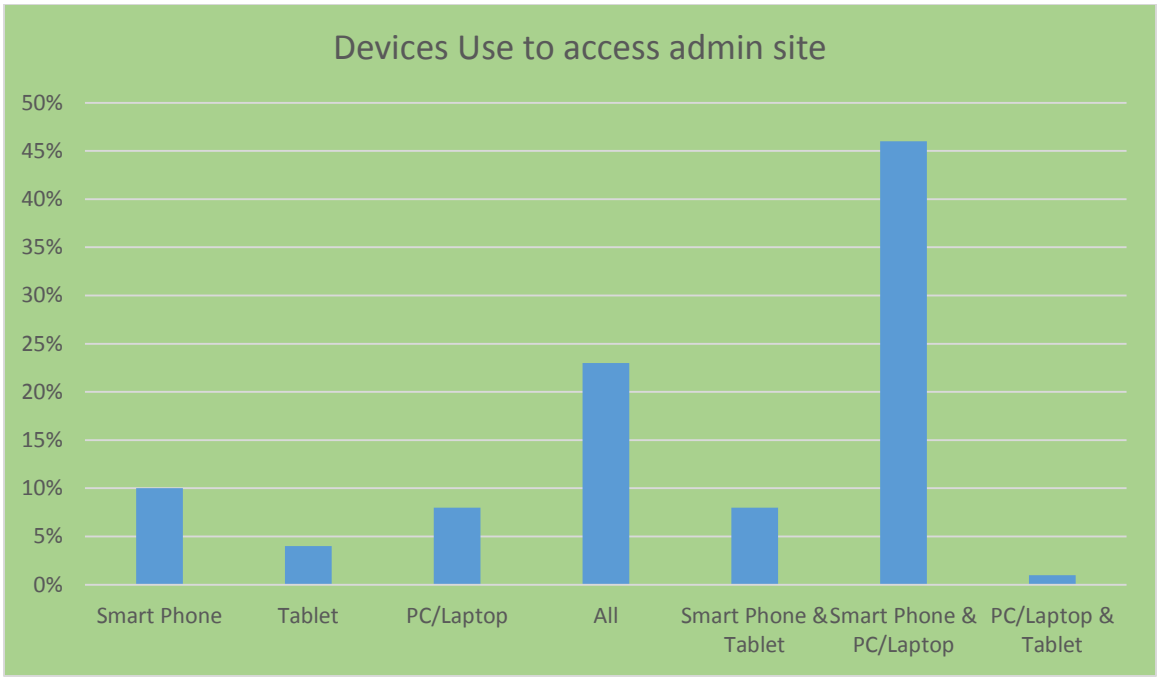


Figure 4.8 - Devices use to access admin site

4.2.2 Formal Interviews

Several formal interviews were conducted with hearing impaired people, teachers who teach the sign language with the target of identifying problems that they faced and find out better approaches. And find out what should include in **USL** system.

Limitations of the Interviews

- Preservation of self-interests – People may express that they want thinking just about them self. So, it may resulting that they are not giving us the best responses for what we asked.
- Lack of knowledge – Certain people may not understand about what the system do. So, they may enable to provide a clear feedback.

4.2.3 Finding from Requirement Elicitation

Following table shows a summary of findings gathered through the requirement elicitation process.

Finding	Literature Review	Questionnaire	Formal Interview	Observations	Self-Evaluation
Should be scalable enough to support both mobile and standard web users		✓		✓	✓
Should be able to configure other languages	✓	✓			
Should be able to ship the system as a easily portable device					✓
Should simple enough for users		✓	✓		
Should be able access by every one					
Should be developed specifically to provide a higher accuracy rate and maintain a higher scalability level	✓			✓	✓
Should be provide above 70% accuracy rate		✓			✓
Should be able be define different languages					✓
Should be able to check the language define by others		✓	✓		✓

Table 4.2 – Summary of Findings

4.3 Stakeholders

Stakeholders and Roles

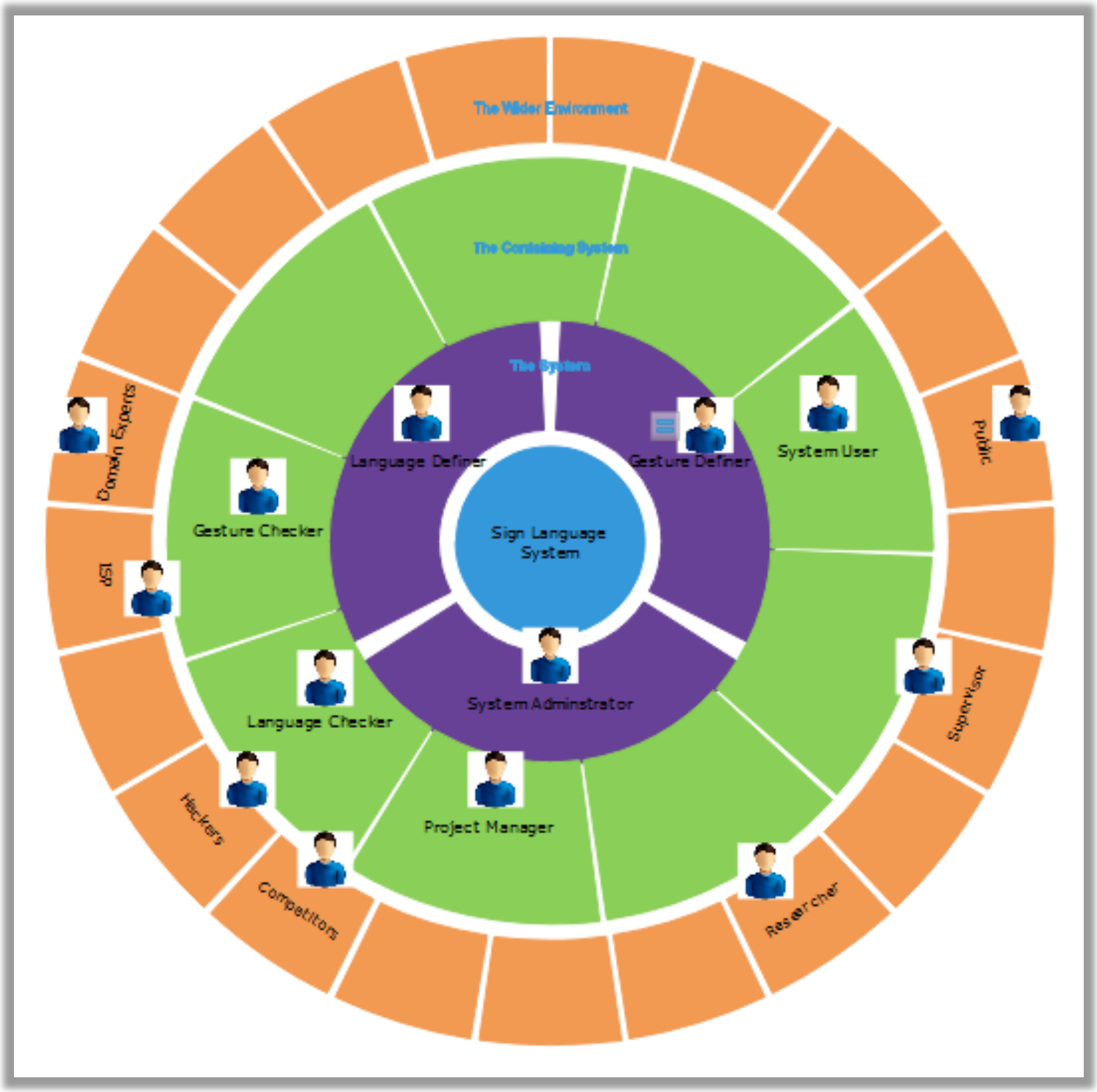


Figure 4.9 - Onion Diagram for the Proposed System

Above onion diagram represents the identified stakeholders and their roles in proposed solution and below listed are the pressure points of the diagram:

1. Project manager should make sure that the system is met with required standards and qualities.
2. Project manager should make sure that the system is user friendly.
3. Project manager should make sure that the system will be financially beneficial for the on line news company owner.
4. Project Manager should make sure that the system is superior and secure
5. Researcher should make sure that the whole process goes on according to plan and is be able to finish it on time.

Table No 4.3 Lists the roles and viewpoints of each identified stakeholder

Stakeholder	Role	Viewpoint
System User	Operator/ Functional Beneficiary	Expects to use the system to communicate View, rate, comment on suggestions and improvements
Language Definer	Operational role support	Configure different languages to gestures according to sign language
Gesture Definer	Operational role support	Configure different gestures according to sign language
System Administrator	Operational role administration	Easily administrates the whole system. Easily recover the system on a failure
Language Checker	Functional support	Verify assigned language words correctness
ISP	Financial beneficiary	Increase the profits by providing efficient Internet service for the readers
Competitors	Negative	Would want to identify drawbacks of the system and implement a better solution.
Hackers	Negative	Hack the system and make it unresponsive or incorrect
Public	Negative	To point out the weaknesses of the solution and expect the researcher to correct it
Domain experts	Expert	To provide expert opinion about the technologies and methodologies used for the project.
Government	Political Beneficiary	To make sure that system makes available to hearing impaired people.
Supervisor	Advisory	Provide advices and guidance to successfully finish the project.
Project Manager/ Researcher	Managerial/ Financial beneficiary/ Intellectual	Wants to assure smooth flow of the project Want to develop a bug free system which meets the requirements in time

Table 4.3 – Stakeholders and Roles

4.4 Use Case Diagram

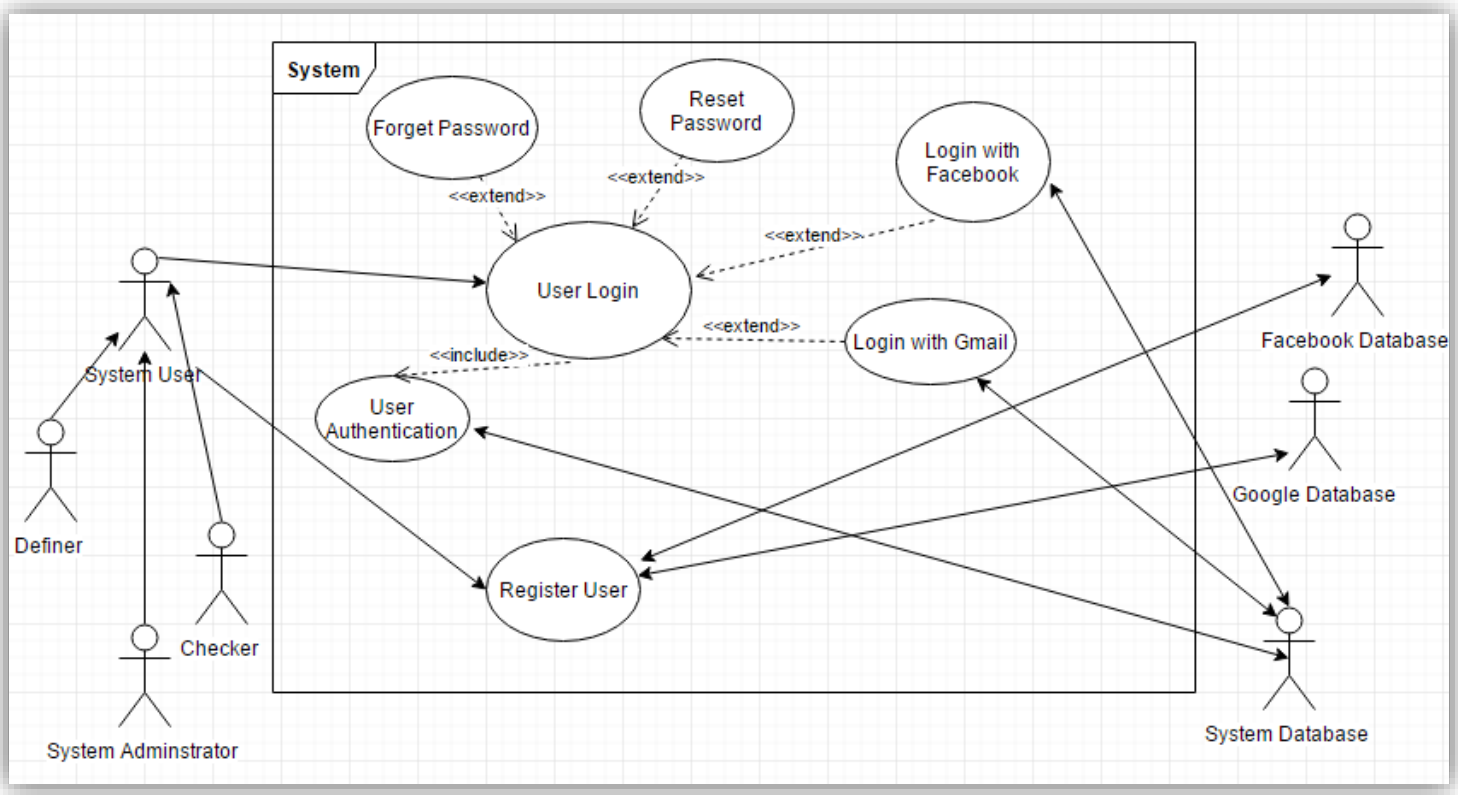


Figure 4.10 - Use Case Diagram for USL System 1

Use Case Descriptions

Use Case Name	User Login
Description	System Administrator, System User, Language Definer, Language Checker, Gesture Definer, Gesture Checker who request entry to the system must prove who they are. This determines what he or she has access to.
Actors	Primary Actors - System Administrator, System User, Language Definer, Language Checker, Gesture Definer, Gesture Checker, System Database
Pre-Conditions	The USL system is operational via internet.
Flow of Events	Basic Flow : 1. System shows login screen with username, password, reset password and forget password items. 2. User enters a username and password. 3. The system responds by checking the combination of username and password against the recorded list of valid User from the System Database. 4. The system responds by notifying the user that the username and password are valid and allows user access to the application.
	Alternative Flow :

	1. In step 2a, if the user wants to reset password, then the system asked the old password, new password and retype new password. 2. In step 2b, if the user forgot his or her password, then the system asked for email address. 3. In step 1a, if the user wants to login with Facebook username and password, system will check them and allows to login to the system. 4. In step 1b, if the user wants to login with Gmail username and password, system will check them and allows to login to the system.
Exceptions	1. In step 2a, if the user gives a wrong old password, then the system responds by notifying the customer that the password incorrect and ask customer to try again. 2. In step 2b, if the user gives an invalid email address, then the system responds by notifying the user that the email isn't valid and ask customer to try again. 3. In step 3, if the username and password combination is not valid, the system responds by notifying the user that the combination is invalid and ask user to try again.
Post-Condition	The validated username show in the system and status changed as logged in.

Table 4.4 - Use Case Description for User Login

Use Case Name	Register User
Description	User can be a checker, definer or a user. For register as a checker or a definer system administrator must accept them by reviewing their actions
Actors	Primary Actor – System User, Language Definer, Language Checker, Gesture Definer, Gesture Checker, System Database Secondary Actor - System Database, Facebook Database, Google Database
Pre-Condition	The USL system is operational via internet. Valid email address which customer can access.
Flow of Events	Basic Flow : 1. The system asks the user to select how they want to register. 2. User select USL system user option and user enter the personal details including chosen username, password, retype password, email, and phone number and submit the registration form. 3. System checks the details between other users, and saves them in the USL database and sends the verification email. 4. System prompts the user to check the mail and verify the USL system account. 5. User clicks the verification link in the USL system verification mail. 6. Link redirected the user to USL home page with a welcome message. Alternative Flow :

	1. In step 2, if the user select Facebook option, then the system only need username and password of the Facebook account. 2. In step 2, if the user select Google option, then the system only need email and password of the Google account. 3. In step 3, if the user enters a username which is already there, system suggests some usernames for user with notifying that the entered username already exists.
Exceptions	1. In step 3, if the password mismatch or didn't meet the password word requirements, then the system responds by notifying the user that the password mismatches or didn't meet the required requirements and ask user to try again. 2. In step 5, if the user didn't click the verification link within three days, system erases all records for the registration attempt.
Post-Condition	The user is successfully registered with the system

Table 4.5 - Use Case Description for Register User

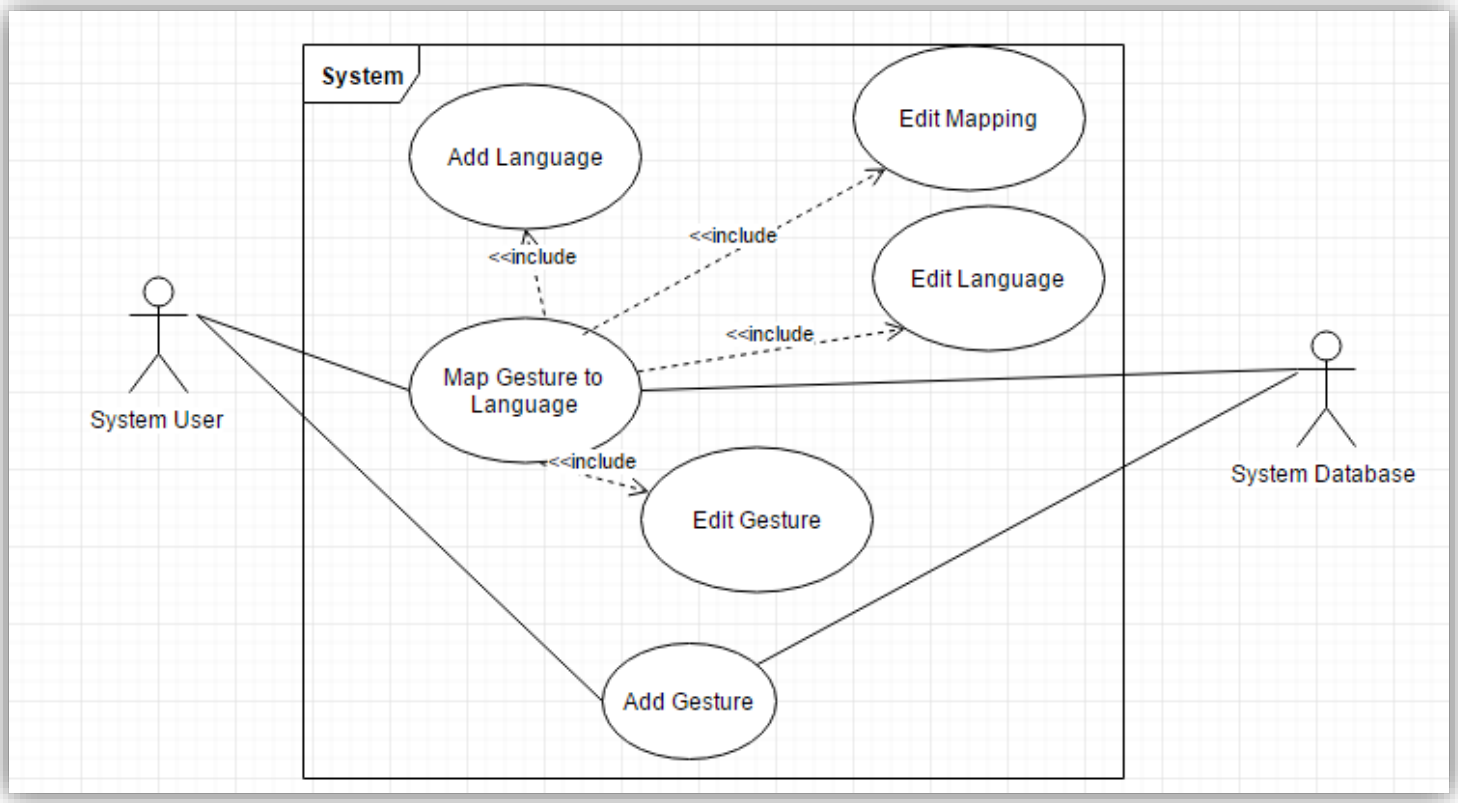


Figure 4.11 - Use Case Diagram for USL System 2

Use Case Descriptions

Use Case Name	Add Gesture
Description	User can add a Gesture
Actors	Primary Actor – System User Secondary Actor – USL Database
Pre-Condition	The USL system is operational via internet.
Flow of Events	Basic Flow : 1. System user click Add Gesture button. 2. System shows gesture define page with hand virtually. 3. User perform the gesture as system request. 4. System user fills all the details and click save button. 5. System saves the provided data, shows the success message and directs to the next page which shows the gesture details. Alternative Flow : 1. In step 1, if user navigate to another page and come back to Add Item page, all the provided data shown in the page.
Exceptions	If the system user didn't provide required data, system responds by notifying the user that required values are not provided.
Post-Condition	New gesture added successfully to the database.

Table 4.6 - Use Case Description for Add Gesture

Use Case Name	Map Gesture to Language
Description	User can add language or a word for a gesture
Actors	Primary Actor – System User Secondary Actor – USL Database
Pre-Condition	The USL system is operational via internet.
Flow of Events	Basic Flow : 1. System user click Edit button for a gesture. 2. System shows gesture details page. 3. System user fills all the details and click save button. 4. System saves the provided data, shows the success message and directs to the next page which shows the gesture details. Alternative Flow : 1. In step 1, if user navigate to another page and come back to Add Item page, all the provided data shown in the page.
Exceptions	If the system user didn't provide required data, system responds by notifying the user that required values are not provided.
Post-Condition	New language added successfully to the database.

Table 4.7 - Use Case Description for Map Gesture to Language

4.5 Functional Requirements

Requirement Prioritization

Due to time and resource limitations it might difficult to implement every identified requirements that identified in the requirement elicitation process. So, identified requirements are prioritized in order to identify most important and less important requirements. Following table shows the priority levels and descriptions.

Level	Description
Critical	Requirements that are represent the core functionality of the system
Important	Requirements that are not essential
Desirable	Requirements that are intended to implement in further developments

Table 4.8 – Requirement Prioritization Levels

The following table shows the identified functional requirements and the priority levels of the system.

Requirement Description	Level	Use Case Mapping
Any new user should be able to register with the system by providing a username and a password.	D	Register User
When registering to the system user can use the Facebook account credentials to authenticate	D	Register User
When registering to the system user can use the Google account credentials to authenticate	D	Register User
Any registered reader should be able to login to the system using his/her credentials	D	User Login
If the authentication fails the system should display an error message to the user with the authentication failure	D	User Login
After a successful login system should display user's hands virtually, when user move the hand on top the LeapMotion device	C	Logged into the System
A logged user should have the ability to record or show the sentence when user perform sign language gestures	C	Logged into the System
A logged user should be able to define or map gestures to words	C	Logged into the System
A logged user should be able to comment about errors in the defined language	I	Indicate preference
The system should be able to record specific language for a user	C	Using the system
Users should be able to update gesture language mapping if they are incorrect	C	Edit Mapping

Administrator should be able to delete user account that misbehave the system	I	Edit Users
Administrator should be able to edit data in the system	C	Edit Language/ Edit Mapping

Table 4.9 – Functional Requirements of the System

4.6 Nonfunctional Requirements

Here are the identified nonfunctional requirements of the USL system. These nonfunctional requirements is to provide a better user satisfaction, improve performance and etc.

Nonfunctional Requirement	Description
Usability	Usability is the ease of use and learnability of the system. System should be effective, efficient and bring satisfaction to users who use it. So the system should ensure higher level of usability by making the prototype available to users with minimum training. The user interface has to be simple and visually appealing to use. Help messages will be provided where necessary and a new user should be able to familiar with the functionalities and the capabilities of the USL system very quickly.
Performance	System should respond quickly for the user requests and additionally the system should be light weight to support resource constraint devices like smart phones and work properly without crashing on all types of devices and platforms.
Scalability	Another important requirement of the USL system is scalability or a higher level of maintainability. When people get to know about this system, they may want to develop the system much more to help hearing impaired people. So the system should be scalable enough to handle to massive change in the future.
Accuracy	In this prototype, it's not conceding the accuracy of the gesture recognition. But, as the system go alone, system should be able to accurately understand the gestures which user perform.

Table 4.10 – Non Functional Requirements

4.7 Chapter Summary

This chapter mainly gather data and define requirements for the new USL system. Chapter starts with ways that used to gather requirements for the system from identified stakeholders. Questionnaires and interviews used as the main requirement elicitation technique other than literature review. Then analyses the gathered data and documented to refine the requirements. The key stakeholders and their roles were identified via an onion diagram, then the chapter went to document list of functional and nonfunctional requirements. Next chapter will be the System Architecture and Design, which will be focusing on the detailed system design of the prototype based on the requirements identified from this chapter.

Chapter 5: System Architecture & Design

Contains

- Chapter Overview
- High Level Design
- System Design
- Domain Model of USL System
- Sequence Diagrams
- Design and Architecture Optimization
- Dependency Diagram
- Entity Relationship Diagram
- Design Goals
- Chapter Summary

5.1 Chapter Overview

Having discussed requirements gathering in the previous chapter, this chapter focuses on the architecture and the design of the USL system. Chapter will first discuss the high level architecture of the system, introducing the sub components. This will be followed by the component level design of the system which focuses the functionalities of each sub component in order to meet the requirements in requirement specification. Finally, integration process of these sub components into main system will be discussed

5.2 High Level Design

High level design of the system provides a clearer picture of the system to everyone. Which allows to fill details step by step about the system components.

5.2.1 USL System

A Rich Picture is an approach to explore a situation and define a high-level solution and then express the solution through diagrams to create a preliminary design model that can be further expanded with an appropriate design methodology. Figure 5.1 illustrated the Rich picture diagram of the USL system.

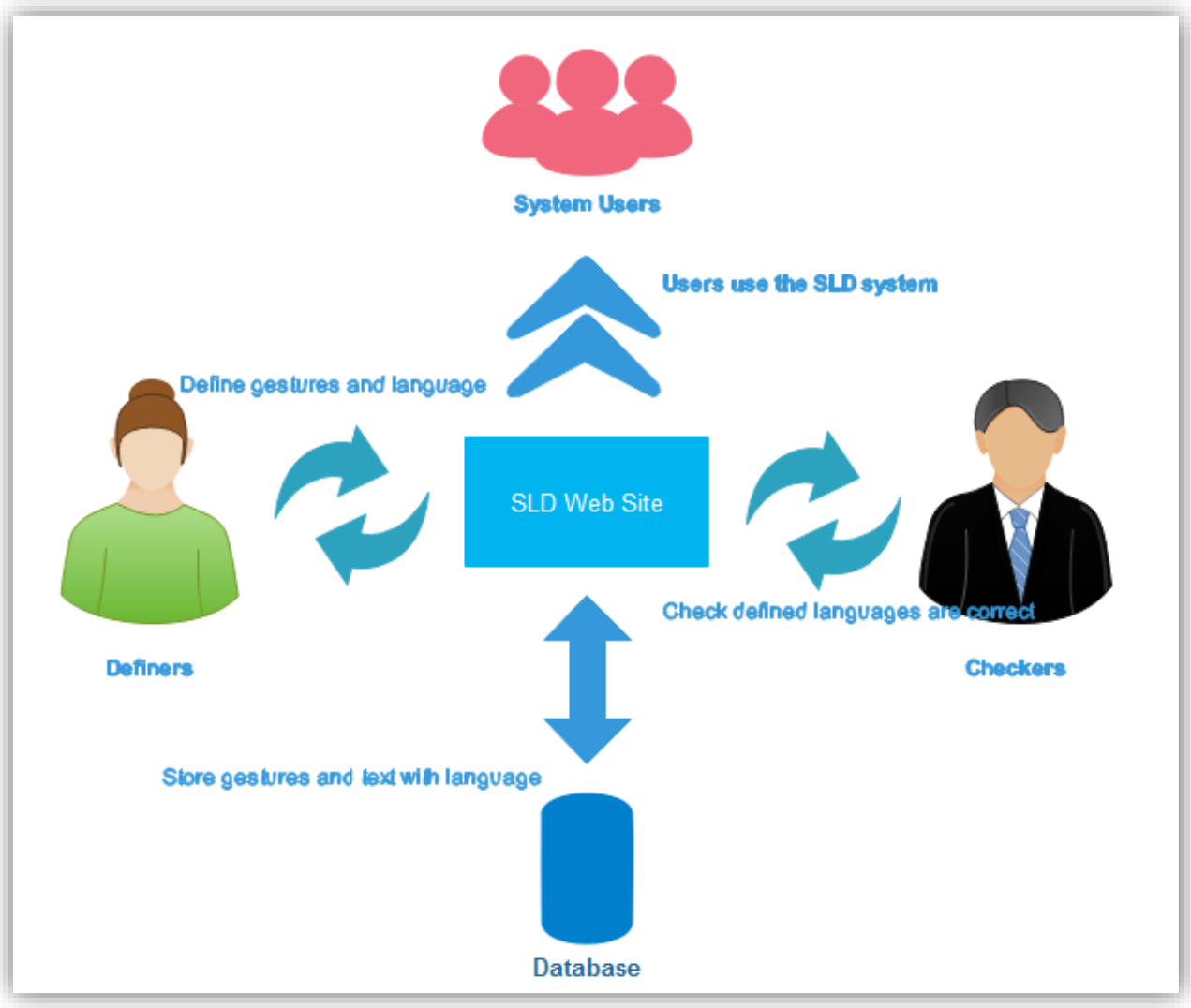


Figure 5.1 – High Level Diagram

According to the above Figure 5.1 the main stakeholders of the USL system are definers, checkers and general users of the system. Definers upload or configure a sign language with gestures with mapping text or audio clip. This can be done in two levels, in user level or in system level. When definers define a language in system level checks will check that language and verify its correctness. To use the system, users can define their own language as well as they can use a defined language.

5.2.2 High Level Architecture

Tier architecture is a software architecture model which used by most of the industry applications. Main advantage of the three-tier architecture is the independency of the three tiers from allowing any of the three tiers to be upgraded or replaced independently. Three-tier architecture, a version of tier architecture is capable of resolving issues like scalability, security, fault tolerance and due to the above mentioned reasons the high level architecture of the proposed USL System was modelled using the three-tier architecture.

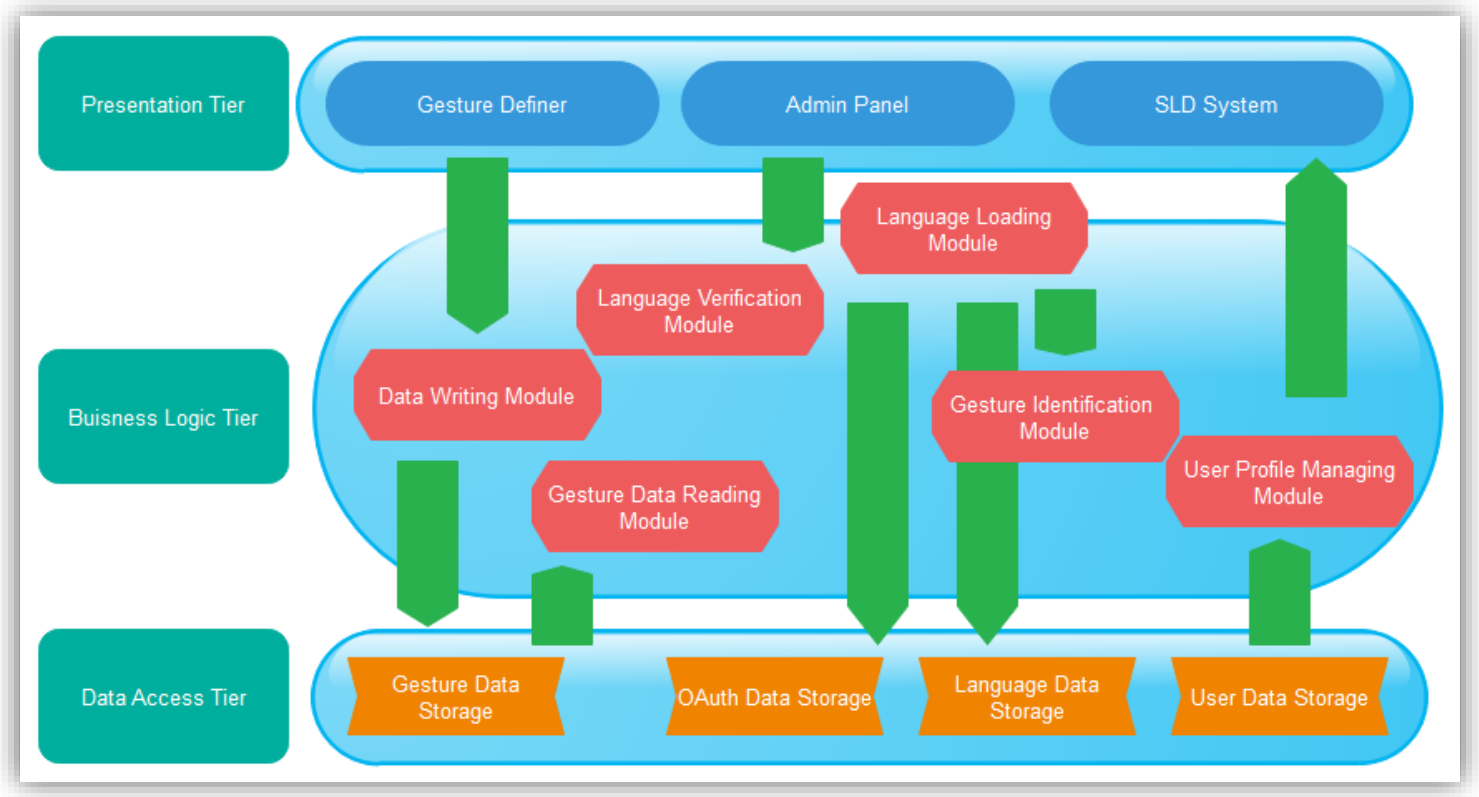


Figure 5.2 – High Level Architecture

According to the above diagram presentation tier exposing 3 modules to the outside world.

- **Gesture Definer:** This module will capture gestures defined by users and store that data with a key in the data storage
- **Admin Panel:** This module is used to configure language types, map gestures to text or audio files and all configurations in the system
- **USL System:** This module is used to present system defined or user defined languages to users

The Business logic tier contains several modules which would carry out the business logic of the proposed system.

- **Data Writing Module:** This module responsible for write all kinds of data to the back end
- **Language Verification Module:** This module is used to verify system languages defined by users. This part implemented in a manual way, but add it to the system as a module hoping to automate it in the future.
- **Gesture Data Reading Module:** Gesture data can be stored in two ways, text and audio. This module will map the appropriate data with the gesture
- **Gesture Identification Module:** This module will capture gestures that user perform
- **Language Loading Module:** This module will load the language according to user selection
- **User Profile Managing Module:** This module manage all user data including checker and definers data

Data storage tier contains with 4 data storages which would store the information that have to be used for SDL system

- **Gesture Data Storage:** This storage will store all gesture related data performing on the system
- **OAuth Data Storage:** This data store data related users who logged in with their Facebook and Google account
- **Language Data Storage:** This storage will store all language specific data with relation of gestures
- **User Data Storage:** This storage will store user data and their actions in the USL system

5.3 System Design

Design methodology is a set of procedure that one follows from the beginning to the completion of the software development process. The nature of the methodology is dependent on several main factors like software development environment, type of the software being developed, the requirements of the users, and the time schedule. Here in this project I'm using OOADM since, it's capable of modelling complex software systems. OOADM approach supports development of object oriented software. OOADM approach is widely used in the industry as key development methodology due its capability in modelling complex and larger software. OOADM is capable of breaking the complex software system down into its various objects, combining the data and the functions that operate on the data into a single unit.

5.4 Domain Model of USL System

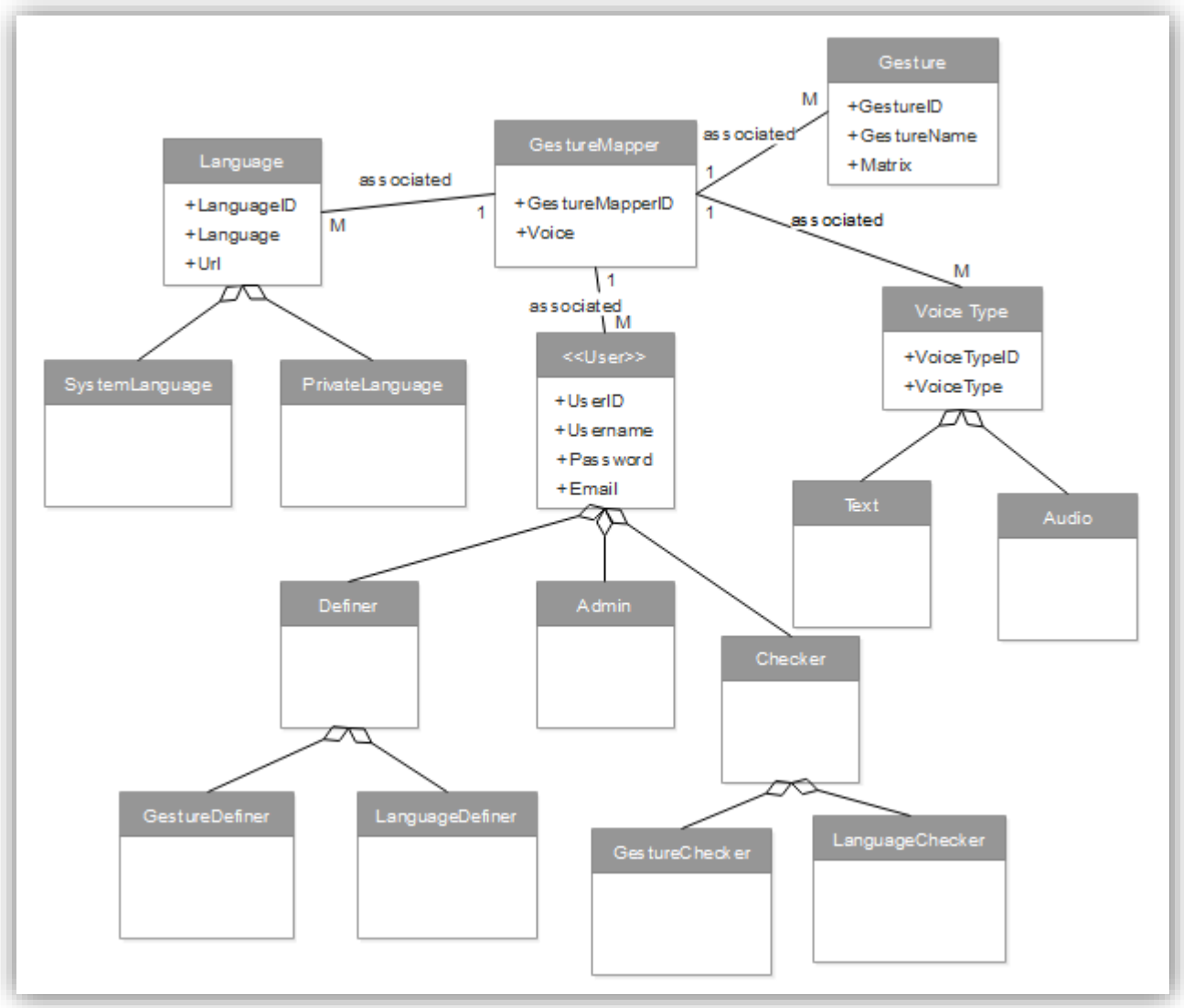


Figure 5.3 Domain Model

Table 5.1 provides an overview of the classes contained in the domain model.

Class	Description
User	A user who use the system is modelled by this class
Definer	This class represent different types of definers in the system
LanguageDefiner	Language definers who add text or audio to created gestures is carried out by this class
GestureDefiner	Gesture definers who perform gestures according to sign languages is carried out by this class
Checker	This class represent different types of checkers in the system
LanguageChecker	Language checkers who check system defined languages according to actual language are modelled thorough this class

GestureChecker	Gesture checkers who check defined gestures are valid according to sign language are modelled thorough this class
Admin	A user who has all the permissions to do anything is defined by this class
VoiceType	This class represent different types of voices in the system
Text	This class models the text type of voice
Audio	This class models the audio type of voice
Gesture	This class models the gesture
GestureMapping	This class carried out the modelling to map gestures to language
GestureNaming	Naming the gesture according to language is modelled by this class
Language	This class represent different types of languages in the system
SystemLanguage	This class represent actual sign language
PersonalLanguage	This class represent user defined language

Table 5.1 - Domain Model Description

5.5 Design and Architecture Optimization

In order to optimize the implementation of the USL system, following design patterns adapted.

Singleton Design Pattern

Singleton pattern was used to implement the entity framework connection class, since one instance of the connection class is required throughout the application.

MVC Design Pattern

The REST services were modelled using the MVC pattern to clean separation between the business logic involved in those REST service from the presentations to make those layers independent to support any future enhancements that can happen in the system.

5.6 Dependency Diagram

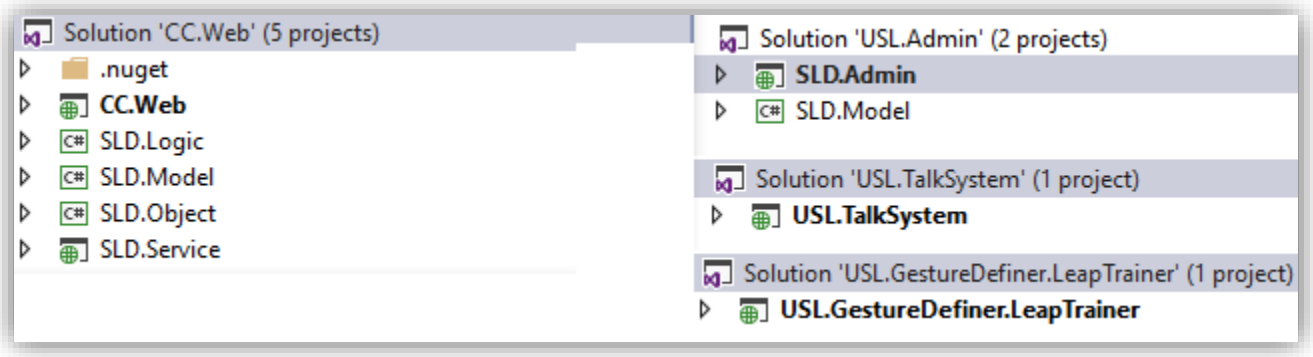


Figure 5.4 – Dependency Diagram

USL System contrians 4 modules. Which works independently. Reason for this apporch is to enable much extensibility to the system.

5.7 Entity Relationship Diagram

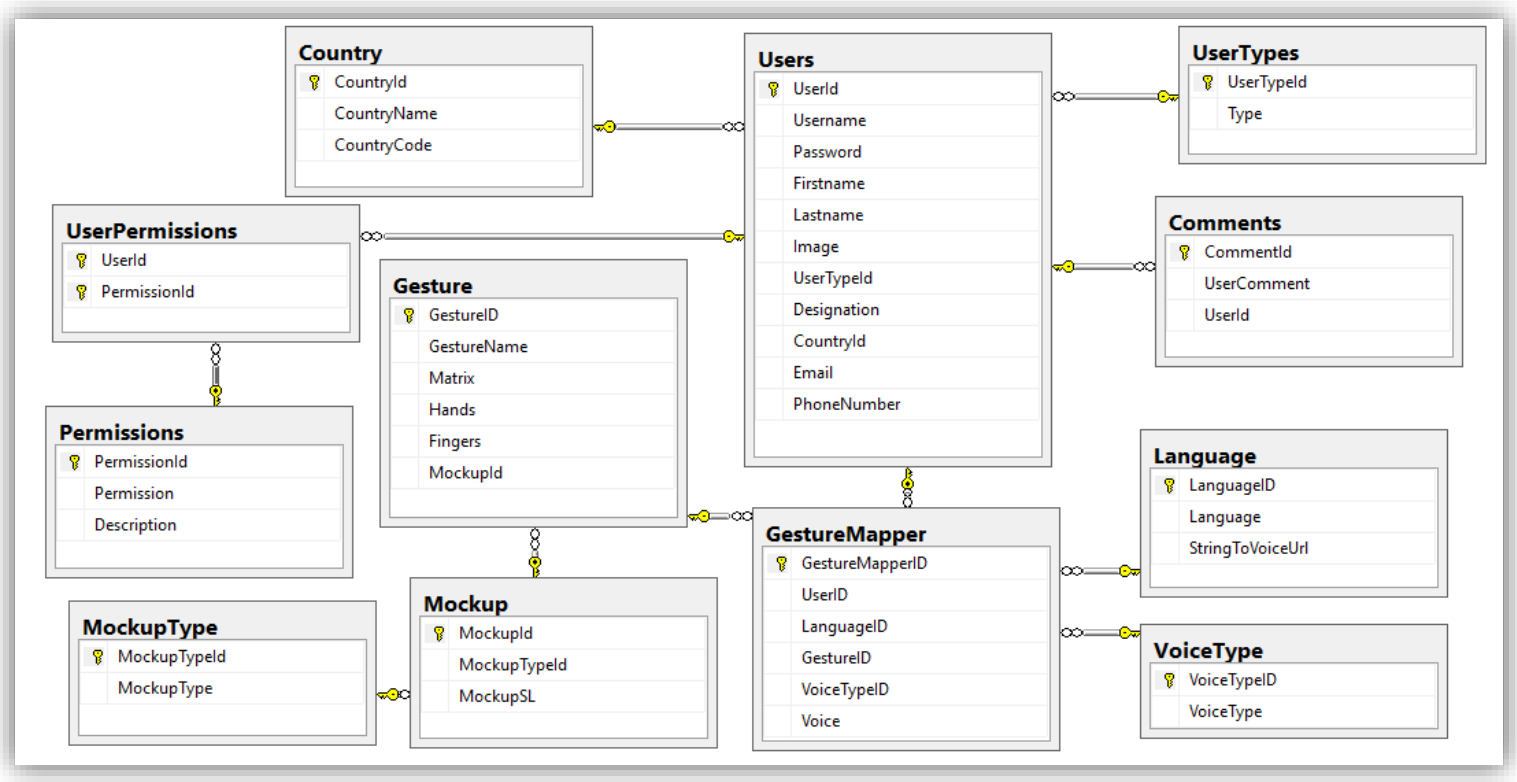


Figure 5.5 – Entity Relationship Diagram

5.8 Design Goals

Following are the main design goals that were adhered during the design phase of the overall system.

Accuracy

Accuracy of all the modules is vital to achieve successful sign language system. During the design phase careful consideration was given to achieve the highest accuracy rate for the USL system.

Scalability

Since scalability is one key success factor for a USL system to improve the days go by, was given to achieve the highest possible scalability for the USL system

Adaptability

The users should be able to adapt to the functionalities of the system within limited time period. Users should be able to use the system with minimum training.

5.9 Chapter Summary

This chapter discussed the architecture and design of the USL system. It begins with high level system design and discussed the modules it has. Gesture capturing and identification, saving with language, user profile updating like modules identified as main modules of the system. Then low level design diagrams such as domain model are discussed. The next chapter will focus on the prototype implementation of the USL system.

Chapter 6: Implementation

Contains

- Chapter Overview
- Technologies
- Implementation of REST Service
- Implementation of Rest Service
- Implementation of Admin Module
- Implementation of Dashboard Module
- Implementation of Gesture Definer
- USL System Implementation
- Chapter Summary

6.1 Chapter Overview

Since, we discussed design of the proposed system in the previous chapter, in this chapter it will be focused on implementation of USL system including frameworks, environments, APIs and languages. It starts off with a discussion on the technology selection for the implementation of the prototype followed by detail account of the implementation process, problems encountered and solutions agreed upon will also be discussed in using appropriate code snippets and screenshots.

6.2 Technologies

Selection of data storage

Microsoft SQL Server is a very stable, fast, extremely popular and affordable database engine. There is a tremendous amount of support and resources available on the web and from Information Technology providers. Plus, it's relatively easy to work with. So, among all the databases I selected MSSQL server to implement USL system.

Selection of ORM

Object-relational mapping (ORM) in computer science is a programming technique for converting data between incompatible type systems in object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language.

USL system require a database it was key to find out a suitable ORM framework. Since in this system use MSSQL database, entity framework selected as the ORM framework.

Approach to capture gestures from users

A gesture and pose learning and recognition framework for the LeapMotion**.** Since we are presenting a layer to capture gestures from users it was decided to use LeapTrainer framework to capture the user interaction due to its support for Leapmotion device. Each user who want to define a new gesture, this framework is ideal for them.

Web service exposure approach

There are two web service exposing approaches known as the SOAP and REST. When comparing SOAP and REST, it can be decided that the most suitable approach to expose the web service is to use the REST approach due to the below mentioned reasons.

- REST services can be consumed by any type of a client
- REST services are lightweight and consumes less bandwidth
- Rest services adopt advanced security mechanisms
- Ease of learning and ease of expanding the service

Selection of programming language for Rest Service

Since .NET framework has web service supports Leapmotion device, it give the ability to develop systems rapidly and functionalities it gives and it makes deployment much easy, so it was decided to use the C# language to develop the USL system.

Selection of an IDE and a deployment environment

It was decide to use Visual Studio 2015 as the IDE and IIS as the deployment environment. VS2015 was selected due to previous exposure and IIS was selected due to its capability in resolving the project dependencies intelligently.

6.3 Implementation of Rest Service

All the dashboard user interaction capturing module was decided to be implemented using web services which would be developed using the .NET framework. ASP.NET Web API is a framework for building web APIs on top of the .NET Framework.

```
0 references
public class UserController : ApiController
{
    IUserLogic _userLogic = new UserLogic();

    0 references
    public IEnumerable<UserDto> GetAllUsers(int userTypeId)
    {
        return _userLogic.Users(userTypeId);
    }
}
```

Figure 6.1 – User Rest service end point

It does not contain any logic there. Just returning whatever comes from the repository of logic layer. When we need more security or other end point related validations, we can implement those stuff here.

This is logic layer code snippet, every class is implemented using an interface for extensibility and as a better code practice. Logic layer classes only contain business logics that are required in the system.

```
1 reference
public class UserLogic : IUserLogic
{
    IUserRepository _userRepository = new UserRepository();

    2 references
    public IList<UserDto> Users(int userTypeId)
    {
        return _userRepository.Users(userTypeId);
    }
}
```

Figure 6.2 – User Logic Layer Implementation

```
2 references
public interface IUserLogic
{
    2 references
    IList<UserDto> Users(int userTypeId);
}
```

Figure 6.3 – User Logic Layer Interface

Accessing the data model happens in data layer. It connect to the database through ORM framework and do the CRUD operations that are required.

```
2 references
public IList<UserDto> Users(int userTypeId)
{
    return context.Users.Select(u => new UserDto
    {
        UserId = u.UserId,
        Username = u.Username,
        Firstname = u.Firstname,
        Lastname = u.Lastname,
        UserTypeId = u.UserTypeId,
        Country = context.Countries.Where(c => c.CountryId == u.CountryId).Select(
        Designation = u.Designation,
        Image = u.Image
    }).Where(u => u.UserTypeId == userTypeId).ToList();
}
```

Figure 6.4 – User Data Layer Implementation

```
2 references
public interface IUserRepository
{
    2 references
    IList<UserDto> Users(int userTypeId);
}
```

Figure 6.5 – User Data Layer Interface

In this data access layer using Linq and entity framework, get the required data from the database and convert the data to a custom object and return the data. The purpose of converting database entity data to a custom object is to optimize the lazy loading of data through the framework.

```
<ArrayOfUserDto xmlns:i="http://www.w3.org/2001/XMLSchema-instar
  <UserDto>
    <Country>Sri Lanka</Country>
    <Designation>Software Engineer</Designation>
    <Firstname>Chathuranga</Firstname>
    <Image>chathuranga_wijesinghe.jpg</Image>
    <Lastname>Wijesinghe</Lastname>
    <Password i:nil="true"/>
    <UserId>1</UserId>
    <UserTypeId>1</UserTypeId>
    <Username>kaddi</Username>
  </UserDto>
  <UserDto>
    <Country>Sri Lanka</Country>
    <Designation>Software Engineer</Designation>
    <Firstname>Chathuranga</Firstname>
    <Image>chathuranga_wijesinghe.jpg</Image>
    <Lastname>Rukshan</Lastname>
    <Password i:nil="true"/>
    <UserId>2</UserId>
    <UserTypeId>1</UserTypeId>
    <Username>crw</Username>
  </UserDto>
</ArrayOfUserDto>
```


Figure 6.6 – Json of User Response

Complexity of the code: Hosting a REST service through IIS is not a very challenging task as all the key required functionalities are inbuilt in to the Visual Studio 2015 and the implementation effort would only be in reading the http request and extracting of the JSON object.

Problems faced: None

6.4 Implementation of Admin Module

The admin module which works as the configuring all USL system related things. So, it was decided to implement using the .NET MVC framework since all the CRUD operations handled by the admin module and .NET MVC framework provides rapid implementation of CRUD operations through the framework. Framework generates all the view for CRUD operations for a particular controller.

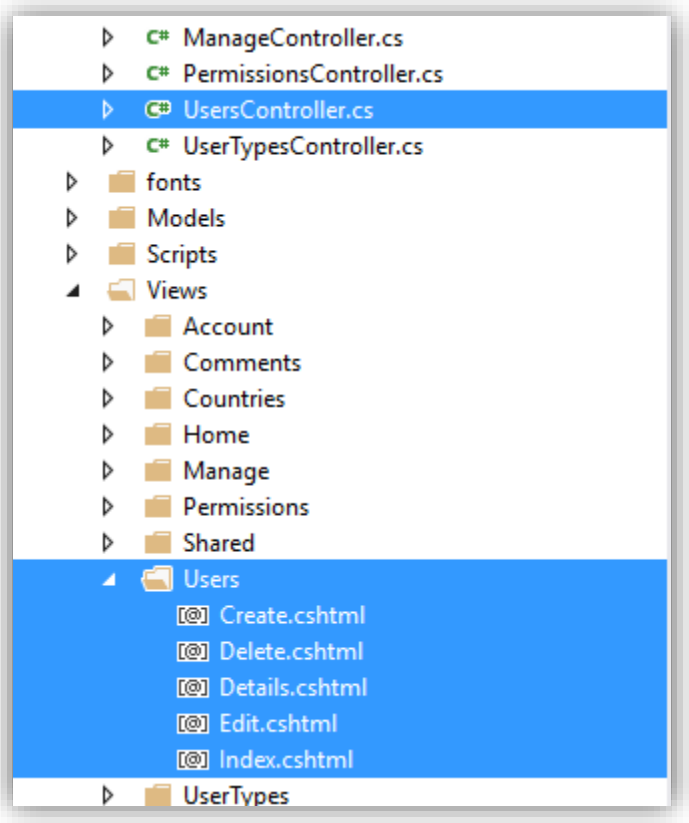


Figure 6.7 – User Controller and Associated Views

When creating the controller we can map a database context and a database entity so that it will create all the CRUD operations in controller by default.

```
0 references
public class UsersController : Controller
{
    private SLD_DatabaseEntities db = new SLD_DatabaseEntities();

    // GET: Users
    0 references
    public ActionResult Index()
    {
        var users = db.Users.Include(u => u.UserType).Include(u => u.Country);
        return View(users.ToList());
    }

    // GET: Users/Details/5
    0 references
    public ActionResult Details(int? id)...

    // GET: Users/Create
    0 references
    public ActionResult Create()...

    // POST: Users/Create
    // To protect from overposting attacks, please enable the specific properties you want to b
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    0 references
    public ActionResult Create([Bind(Include = "UserId,Username,Password,Firstname,Lastname,Ima

    // GET: Users/Edit/5
    0 references
    public ActionResult Edit(int? id)...

    // POST: Users/Edit/5 ...
    [HttpPost]
    [ValidateAntiForgeryToken]
```

Figure 6.8 – User Controller End Points

Complexity of the code: Since .Net MVC framework all UI and calls to database there are not much challenges to face other than customize the template to match other modules.

Problems faced: None

6.4 Implementation of Dashboard Module

6.4.1 Implementation of a View

The dashboard module is developed using ASP.NET Web API 2 and AngularJs. Also, used the HotTowel.Angular.Breeze template. Each view associated with a controller and those are separated within folders to be specific.

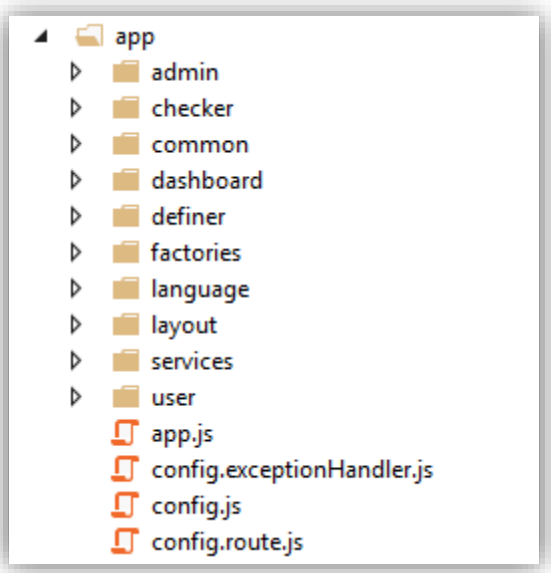


Figure 6.9 - Client File Structure

Each controller has a activate function which call a common function. That common function ensures the data is loaded before viewing the page by returning a promise.

```
angular.module('app').controller(controllerId, ['$routeParams', '

function admins($routeParams, common, config, userFactory) {
    var getLogFn = common.logger.getLogFn;
    var log = getLogFn(controllerId);
    var keyCodes = config.keyCodes;
    var vm = this;

    vm.users = [];
    vm.title = 'Admins';
    vm.refresh = refresh;

    activate();

    function activate() {
        common.activateController([getAdmins()], controllerId)
            .then(function () {
                log('Activated Admin View');
            });
    }

    function getAdmins()...

    function refresh()...
}
```

Figure 6.10 – Admin Controller

```
function activateController(promises, controllerId) {
    return $q.all(promises).then(function (eventArgs) {
        var data = { controllerId: controllerId };
        $broadcast(commonConfig.config.controllerActivateSuccess
    });
}
```

Figure 6.11 – Common Activate Function

In the view data binding happens using AngularJs which is a very powerful framework for databinding. Also it enables two way binding.

```
<section class="mainbar" data-ng-controller="admins as vm">
    <section class="matter">
        <div class="container-fluid">
            <div class="row-fluid">
                <div class="form-search">...</div>
            </div>
            <div class="row">
                <div class="widget worange">
                    <div data-cc-widget-header title="{{vm.title}}"></div>
                    <div class="widget-content user">
                        <div class="padd" data-ng-repeat="u in vm.users.data">
                            <div class="user">
                                <img data-cc-img-person="{{u.Image}}" class="img-p
                                <div class="user-details">
                                    <small></small> from <small>{{u.Country}}</sma
                                    <h3>{{u.Designation}}</h3>
                                    <h6>{{u.Firstname}} {{u.Lastname}}</h6>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
                <div class="widget-foot">
                    <div class="clearfix"></div>
                </div>
            </div>
```

Figure 6.12 – User View

6.4.2 Implementation of the Logger

Client side logger implemented as separate factory and inject into the common module. Since it's a common module it contains all types of logging such as success, error, warn and warning.

```
angular.module('common').factory('logger', ['$log', logger]);

function logger($log) {
    var service = ...;

    return service;

    function getLogFn(moduleId, fnName) {
        fnName = fnName || 'log';
        switch (fnName.toLowerCase()) { // convert aliases
            case 'success':
                fnName = 'logSuccess'; break;
            case 'error':
                fnName = 'logError'; break;
            case 'warn':
                fnName = 'logWarning'; break;
            case 'warning':
                fnName = 'logWarning'; break;
        }

        var logFn = service[fnName] || service.log;
        return function (msg, data, showToast) {
            logFn(msg, data, moduleId, (showToast === undefined) ? true : showToa
        };
    }

    function log(message, data, source, showToast) {
        logIt(message, data, source, showToast, 'info');
    }
```

Figure 6.13 – Common Logger

6.4.3 Implementation of the Routing

All routing has been handle using AngularJs route provider in the separate file.

```
// Collect the routes
app.constant('routes', getRoutes());

// Configure the routes and route resolvers
app.config(['$routeProvider', 'routes', routeConfigurator]);
function routeConfigurator($routeProvider, routes) {

    routes.forEach(function (r) {
        setRoute(r.url, r.config);
    });
    $routeProvider.otherwise({ redirectTo: '/' });

    function setRoute(url, definition) {
        $routeProvider.when(url, definition);
        return $routeProvider;
    }
}

// Define the routes
function getRoutes() {
    return [
        {
            url: '/',
            config: {
                templateUrl: 'app/dashboard/dashboard.html',
                title: 'dashboard',
                settings: {
                    nav: 1,
                    content: '<i class="fa fa-dashboard"></i> Dashboard'
```

Figure 6.14 – Route Handler

6.5 Implementation of Gesture Definer

For implement the gesture definer module I used LeapTrainerJs, since it gives the functionality I need and it's a fast extending framework for LeapMotion device. Bit of customization had to be done using Angular and JavaScript. Additional data will capture and store locally when the gesture is defined, when the save button clicks all the stored data will store in the database.

```
var controller = new Leap.Controller({
    enableGestures: true,
    frameEventName: 'animationFrame'
});

var hands = 0;
var fingers = 0;
var isDone = false;

var handCountDisplay = document.getElementById('handCount');
var fingerCountDisplay = document.getElementById('fingerCount');

controller.on('frame', function (frame) {
    hands = frame.hands.length;
    fingers = frame.fingers.length;

    if (!isDone) {
        handCountDisplay.innerText = hands;
        fingerCountDisplay.innerText = fingers;
    }
});
controller.connect();

var app = angular.module('myApp', []);
app.controller('myCtrl', function ($scope, $http, $window) {
```

Figure 6.15 – Customize LeapTrainer Gesture Definer

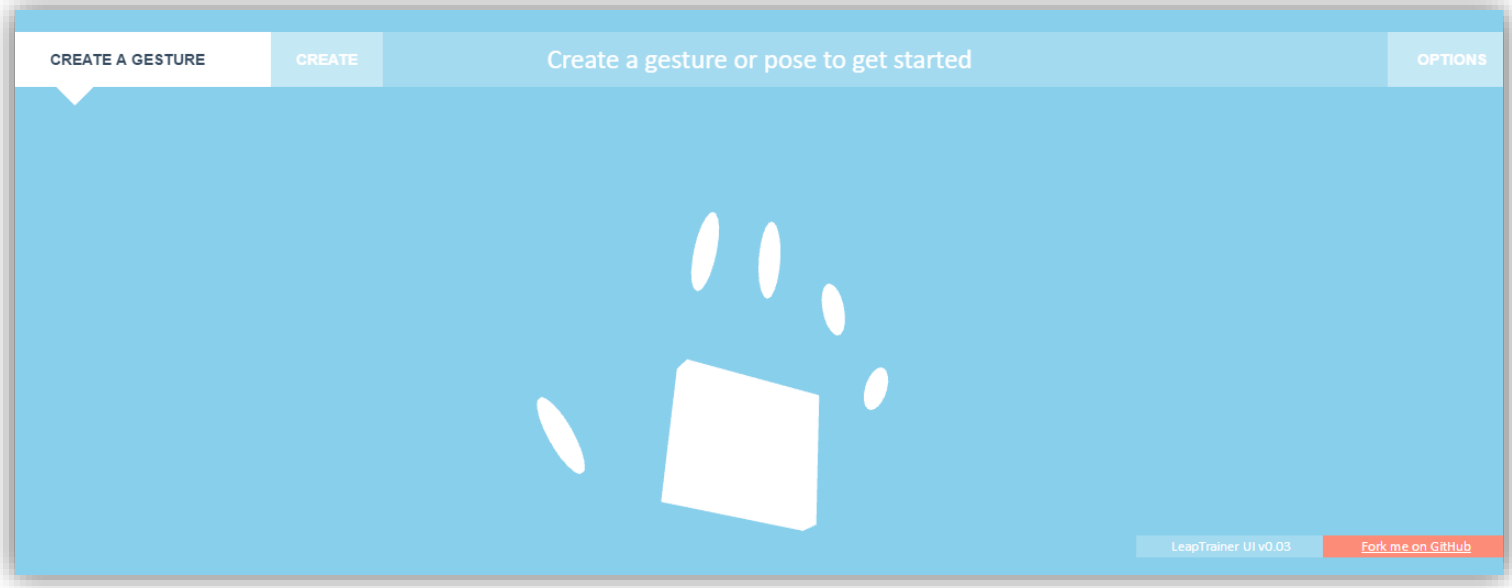


Figure 6.16 – Gesture Definer Module

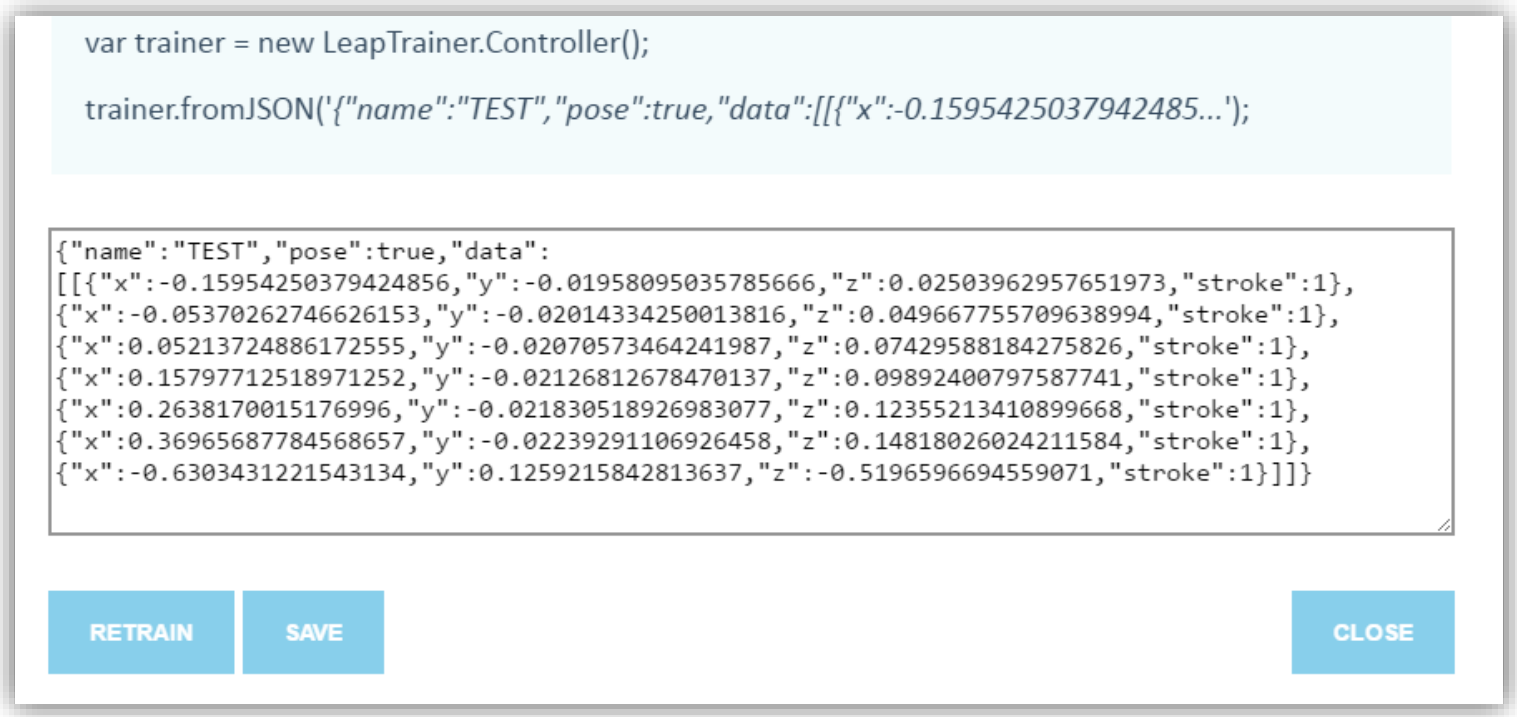


Figure 6.17 – Gesture Definer save Interface

Complexity of the code: Since LeapTrainerJs is a well-developed module there are not much challenges to face other than customize the template to match other modules.

Problems faced: None

6.6 USL System Implementation

For implementation of the USL system, just used AngularJs and Javascript. Mostly used predefined unchangeable gestures defined in the JavaScript itself. Customize gestures are mapped according to user or language selected through the database.

```
var trainer = new LeapTrainer.Controller();

trainer.fromJSON('{"name":"LEFT","pose":false,"data":[[{"x":0.3016
trainer.fromJSON('{"name":"RIGHT","pose":false,"data":[[{"x":-0.27
trainer.fromJSON('{"name":"RIGHT_","pose":true,"data":[[{"x":-0.34
trainer.fromJSON('{"name":"LEFT_","pose":true,"data":[[{"x":0.3472

trainer.on('LEFT', backspace);
trainer.on('RIGHT', talk);
trainer.on('LEFT_', backspace);
trainer.on('RIGHT_', talk);

var uri = 'api/gesture';

var app = angular.module('myApp', []);
app.controller('myCtrl', function ($scope, $http) {
    $http({
        method: "GET",
        url: "api/gesture"
    }).then(function mySucces(response) {
        //$scope.myWelcome = response.data;
        angular.forEach(response.data, function (Gesture, index) {
            //$scope.myWelcome = Gesture.UserID;
            trainer.fromJSON(Gesture.Matrix);
            trainer.on(Gesture.GestureName, function () {
                if (checkLanguage(Gesture.Voice) && fingers == Ges
                    words.push(Gesture.Voice);
                }
                setData();
            });
        });
```

Figure 6.18 – USL System Customize Gestures Code

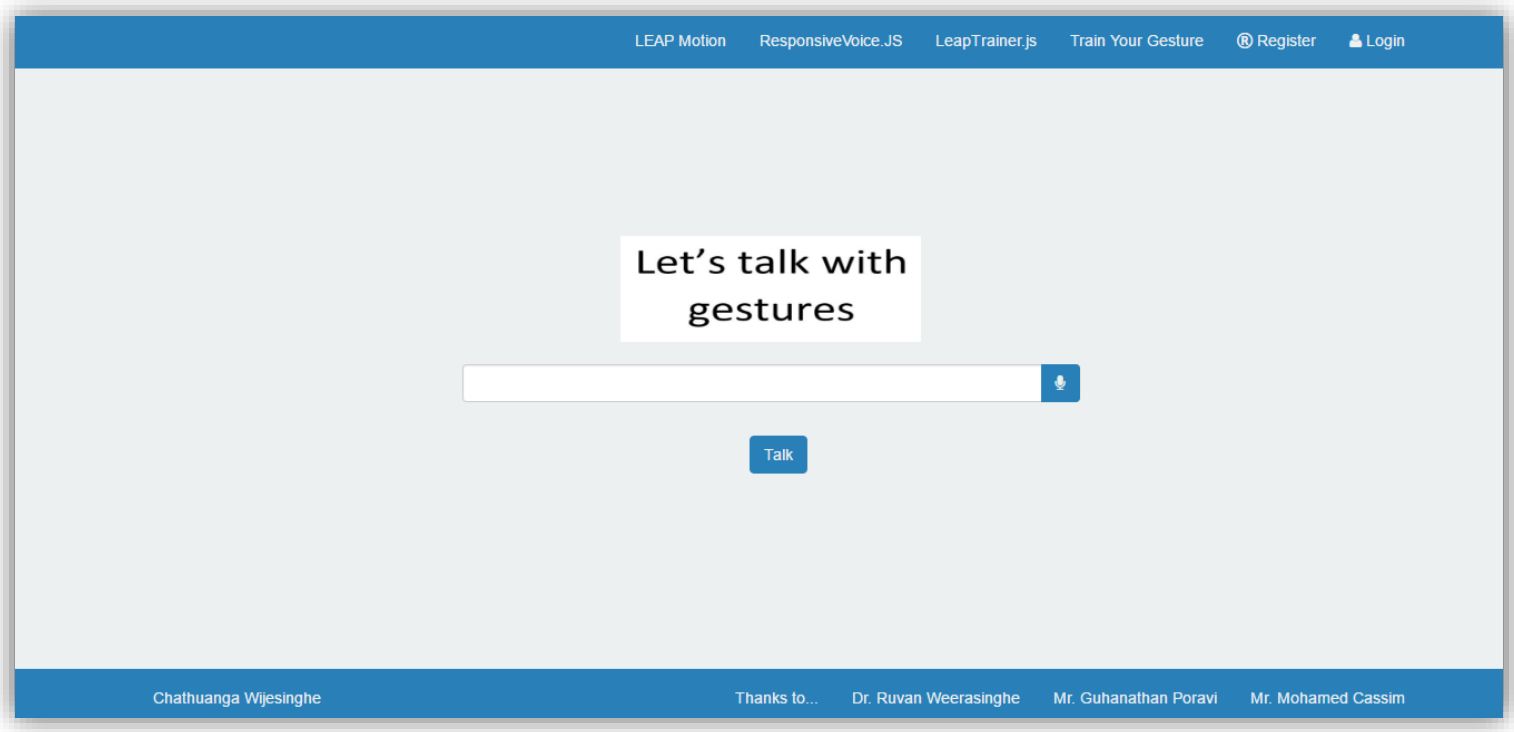


Figure 6.19 – USL System Interface

Complexity of the code: Since AngularJs and HTML 5 is day to day use technologies there wasn't much challenge to implement USL System.

Problems faced: None

6.7 Chapter Summary

This chapter discussed about the implementation process of the USL system. Chapter started with justifying the use of core engine development language and then move on to module implementation.

Chapter 7: Testing

Contains

- Chapter Overview
- Objectives of Testing
- Testing Criteria
- Functional Requirements Testing
- Module and Integration Testing
- Nonfunctional Requirements Testing
- Limitations of the Testing Process
- Chapter Summary

7.1 Chapter Overview

Having discussed the implementation of the proposed USL System in the last chapter, this chapter will be focused on testing the functional and nonfunctional requirements of the implemented system with the intension of making the requirements complete and up to the expected level. This chapter first outlines the testing criteria, testing methods and testing levels and finally will provide an evaluation of the testing results.

7.2 Objectives of Testing

Software testing is performed to verify that the completed software package functions according to the expectations defined by the requirements.
The main objectives of the testing process for USL system are:

- To verify and validate the functional requirements of the USL system.
- To verify and validate the nonfunctional requirements of the USL system
- To identify the defects of the system and make sure that the final product it error free.
- To further enhance the system based on the test results.

7.3 Testing Criteria

Testing the implemented system is the process of an examining an application to ensure that it satisfies the functional and non-functional requirements and meets quality expectations. Software quality can be measured in two ways as described below

Software functional quality - Mainly focus on the combination of the product development characteristics with the technical requirements of the given design based on the functional requirements.

Software structural quality - This uses to measure the performance of the functional requirements of the product with the identified non-functional requirements.

7.4 Functional Requirements Testing

Spiral methodology, which is the adopted software development methodology, allows the flexibility and freedom of carrying out the testing of the software parallel with the implementation phase. Hence the testing of the implemented functional requirements was carried out parallel to the implementation of the USL system with the black box testing approach.

Table 7.1 shows a summary of the functional requirement testing results.

Functional Requirement	**Pass Rate**	**Status**
Any new user should be able to register with the system by providing a username and a password.	100%	Pass
When registering to the system user can use the Facebook account credentials to authenticate	100%	Pass

When registering to the system user can use the Google account credentials to authenticate	100%	Pass
Any registered reader should be able to login to the system using his/her credentials	100%	Pass
If the authentication fails the system should display an error message to the user with the authentication failure	100%	Pass
After a successful login system should display user's hands virtually, when user move the hand on top the LeapMotion device	100%	Pass
A logged user should have the ability to record or show the sentence when user perform sign language gestures	100%	Pass
A logged user should be able to define or map gestures to words	100%	Pass
A logged user should be able to comment about errors in the defined language	100%	Pass
The system should be able to record specific language for a user	100%	Pass
Users should be able to update gesture language mapping if they are incorrect	100%	Pass
Administrator should be able to delete user account that misbehave the system	100%	Pass
Administrator should be able to edit data in the system	100%	Pass

Table 7.1 - Tested Functional Requirements

7.5 Module and Integration Testing

USL System has many software modules in its architecture (refer to section 5. Of the document) where one module was coupled with another in the form of the output of one module is consumed by another module. Since the all modules has to work for the completion of the USL System, it was decided to carry out a module and integration testing on the USL System with the black box testing approach. Each module was tested for 15 times and table 7.2 provides a summary of module and integration testing results of the USL system.

Module	Input	Expected Output	Actual Output	Status
Admin Module	All user, language, gesture, gesture mappings like data	All CRUD operations done to the data should record successfully in the database	All data was captured successfully	Passed
Service Module	All front end data related filters	The data should be return as a json	All data was returned successfully	Passed
Dashboard Module	All front end data	The data should be shown to the users and navigation to other modules	All data was returned successfully	Passed

Gesture Definer	All gestures should capture	Gesture data should capture and persistence in the database	All data was persisted successfully	Passed
USL Talk Module	User gestures	Gesture capturing and making sentences	All gesture related sentences as audio	Passed

Table 7.2 - Summary of Module and Integration Testing

7.6 Non Functional Requirements Testing

7.6.1 Accuracy Testing

A major nonfunctional requirement of the USL system was the accuracy of the user performed gesture capturing and recognition. A self-defined language should provide an accuracy rate above 70% and definers defined language provide an accuracy rate above 40%. Accuracy of the recommendations was determined by using the following equation.

$$Success\ Rate = \frac{Gestures\ Recognized}{Gestures\ Performed}$$

For the testing purpose, fifteen different users were selected and they were told to create user accounts and 10 of them defined their own language and 5 of them using a common language. At the end of using the language they prefer, all the data are gathered and analyzed for accuracy testing.

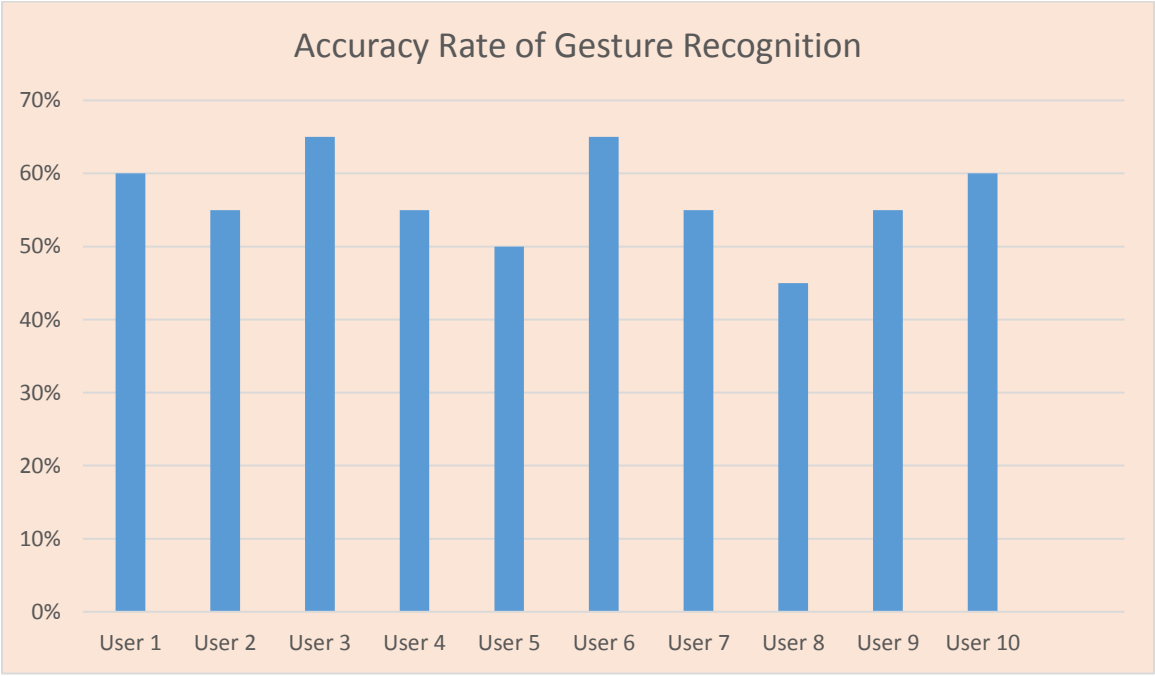


Figure 7.1 - Accuracy Rate of Gesture Recognition for Common Languages

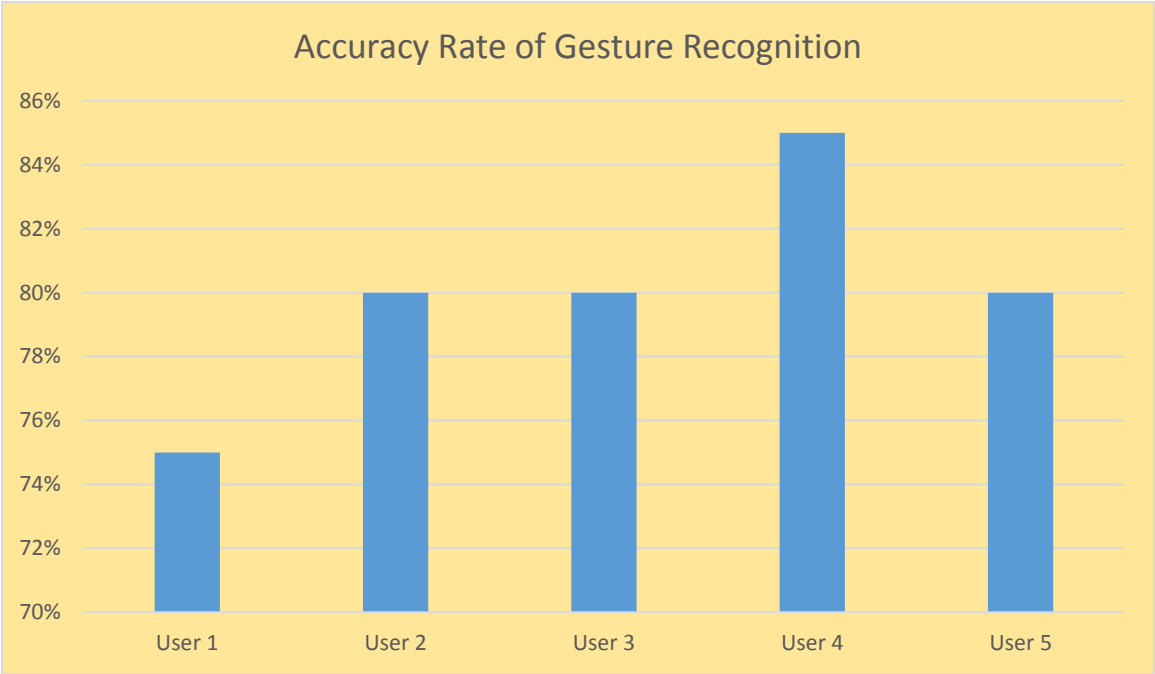


Figure 7.2 - Accuracy Rate of Gesture Recognition for User Language

With the above mentioned statistics it proves that the personal languages gesture recognition accuracy rate is much higher than the common language gesture recognition accuracy rate.

7.6.2 Performance Testing

A performance testing of the USL system was carried out to test the response time of the system and Table 7.3 shows the statistics of the performance testing process carried out using a standard web browser and each test case was executed for twenty times and the average time was taken for the analysis.

Test Case	Purpose	Input Data	Expected Result	Actual Result	Comment
1	Loading the Dashboard page for first time	NA	Three seconds or less	2.57	Pass
2	Redirect Dashboard page	NA	Three seconds or less	1.25	Pass
3	Redirect to USL System	NA	Three seconds or less	2.36	Pass
4	Redirect to USL Admin	NA	Three seconds or less	1.45	Pass
5	Redirect to gesture definer	NA	Three seconds or less	2.43	Pass
6	Perform a CRUD operation	NA	Three seconds or less	1.51	Pass

Table 7.3 - Performance Testing of USL System Using Google Chrome

Though the performance testing results were positive, it's has to be noted that the testing was not carried out in an actual client- server network where other network devices such as routers would have increased the latency of the requests and response.

7.7 Limitations of the Testing Process

Testing was not carried out on a real client server environment

Testing process for the performance was not carried out using a real client server environment. It was carried out using the same machine. As a result of this, current performance results may not represent the true performance values since, in a real network world bottle necks can affect the above performance results.

Not enough time for the system to mature

A major requirement of the USL system is that users can define their own language. Since the system not publically hosted in the web, only selected set of users has the ability to use and test the system.

7.8 Chapter Summary

This Chapter focused on the testing aspect of the prototype and the chapter started with outlining the purpose of the testing phase, testing criteria and testing methods. Under software functional quality testing, functional requirement testing and unit and integration testing were carried out using black box testing approach. Next chapter will describe the evaluation process carried out on various evaluation criteria of the USL project.

Chapter 8: Evaluation

Contains

- Chapter Overview
- Evaluation Criteria
- Evaluators
- Evaluation Methodology
- Evaluation Survey Questions
- Evaluation Survey Findings
- Chapter Summary

8.1 Chapter Overview

The previous chapter described the implementation of the system according to the relevant technologies. This chapter evaluates the developed system with testing evaluation process. All the test data and the Results are mention in this chapter.

8.2 Evaluation Criteria

The following criteria were identified for the evaluation of the project. And these were selected with the aim of covering the phases of the project.

Criteria	Description
Overall Concept	It is develop in a way that the system can used to define sign languages and gestures as user needs
Scope and depth of the project	Since sign languages are using by every country, it is important to get views and comments about the scope of the project from users
System architecture	Evaluate whether the design, architecture and implementation of each module is completed properly
Solution and Prototype	Assessment should be done on the prototype to determine whether the prototype acts as a proof of concept of the USL project.
Usability, performance and accuracy of the prototype	Evaluate the non-functional requirements of the USL to determine the extent the non-functional requirements were implemented.
Limitations and future enhancements	Identification of limitations of the USL system and potential future enhancements that needs to addressed.

Table 8.1-Evaluation Criteria

8.3 Evaluators

Below mentioned evaluators were identified for the evaluation process of the project and high priority will be given to the domain experts in software engineering as a high emphasis of this project was given to the use sign language gesture recognition.

Evaluator	Description
End Users	Group of sign language users in Sri Lanka were selected for the evaluation of the overall concept, usability, performance, accuracy and etc.
Software engineers & architects	Group of well experienced and currently working software engineers and architects were selected to evaluate the System design, Architecture and implementation and future enhancement criteria

Table 8.2-Evaluator Groups

8.4 Evaluation Methodology

Evaluation process of a project exposes the success of the project and provides feedback with regards to implementation and other major phases of the project such as problem, analysis, design and implementation. The project was evaluated using a combination of qualitative and quantitative methods. The quantitative evaluation is more or less achieved through the testing phase of the project. So, more effort given to the qualitative evaluation of the project in this section. Evaluation of the USL project was carried out through questionnaire and interview approaches, but major emphasis was given to questionnaire approach due to time constraints faced at the end of the project.

8.5 Evolution Survey Questions

The below table 8.3 provides a summary of the questions that were asked from various evaluator groups during the evaluation survey.

No	Question
User Related	
1	Do you know people with hearing disabilities?
2	Do you have any experience in sign language?
Overall Concept	
3	What is your general idea about the USL project?
4	What would be the impact this solution would have on the selected user groups?
Scope and depth of the project	
5	Do you think the scope of the project is acceptable for postgraduate level?
6	What is the depth of the prototype should have addressed sign language system?
System architecture	
7	What are your comments about the architecture?
8	Do you think the decisions made in the implementation phase are acceptable and justifiable?
9	What are your suggestions or should improve in the system architecture?
Solution and Prototype	
10	Do you think the implemented prototype have the ability to solve the problem?
11	Do you think the presented solution have the ability to solve the problem?
12	What are the comments on the features offered by the system?
Usability, performance and accuracy of the prototype	
13	What will be the rate that you give to the usability, performance and accuracy of the prototype?
Limitations and future enhancements	
14	What are the limitations you see in the solution?
15	What are your recommendations for limitations you mentioned?
16	What are the new features that you think for USL project?

Table 8.3-Evaluation Survey Questions

8.6 Evaluation Survey Findings

The evaluation results are presented as a summary with comments and suggestions of the evaluators.

8.6.1 Overall Concept

"This kind of software's are used in other countries to communicate with people with communication problems"

Dr. Vindya Wijesinghe

"It save out time in busy clinics, patient also benefits by expressing their problems, taking descriptive history won't be a challenge anymore"

Dr. Hemal Rupasinghe

"I believe your project concept highlights what the software industry should provide to disable people. Basically, it creates a whole new level and a set of opportunities for the disable people to be normal again"

Mr. Chamil Jeewantha (Architect zone24x7)

Evaluation Model	Summary of Feedback
Concept	The feedback from the evaluators about the concept of the project was positive. Especially the hearing impaired people pleased to have a system like this. Also the public also interested since they also want to help the hearing impaired people without abundant them in the society
Review	Talking with hearing impaired people will not be a challenge any more. Most of the evaluators state that this project concept will become more useful in the future.

Table 8.4 - Summary of Evaluation Feedback on Project Concept

8.6.2 Scope & depth of the project

"The depth of the system is quite outstanding and I believe this effort has covered enough scope for a postgraduate level research"

Mr. Eranda Lakshiantha

"Sign language systems are a very broad area and it consists of a substantial depth as well. The depth of the application is quite satisfactory for a postgraduate level research"

Mr Udara Rajapaksha (QA Architect Camboi Solutions)

Evaluation Model	Summary of Feedback
Scope	Overall comment was that the project scope is a challenging and the addressed depth of the project is sufficient enough for a post graduate level research.
Review	Sign language systems are a broad area which has a vast depth and due to those reason it was in doubt to the depth the research should go in to. With the feedback of the domain experts it can be concluded that the depth addressed in this research is sufficient enough for a post graduate level research.

Table 8.5 - Summary of Evaluation Feedback on Project Scope and Depth

8.6.3 System Design, Architecture and Implementation

"Modularization and use of design pattern like Observer and Singleton have contributed on a positive note towards the system and the hybrid design approach is also well thought of"

Mr. Yohan Shamika (Senior Software Engineer)

"The adopted three tier architecture would be sufficient enough at the moment, but it would have been better if the architecture was thought for a SaaS (Software as a Service) platform"

Mr. Asanka Sanjaya (Software Engineer)

"I believe use of .Net related technologies allow the rapid development and scalability of the project as both are light weight components"

Mr. Tiroshan Madushanka (Software Engineer)

Evaluation Model	Summary of Feedback
Design, Architecture and Implementation	Overall comment was that design, architecture and implementation is up to level of acceptable. Some criticized the layout mismatch in the modules.
Review	The argument made that the current architecture would not support a SaaS platform is a valid point but the current architecture was derived for typical client server architecture and the most suitable architecture approach was chosen for the modelling of the system. If this system is going to be converted to a SaaS then the architecture of the system would have to be thought from the beginning.

Table 8.6 - Summary of Evaluation Feedback on Design, Architecture and Implementation

8.6.4 Usability Accuracy and Performance

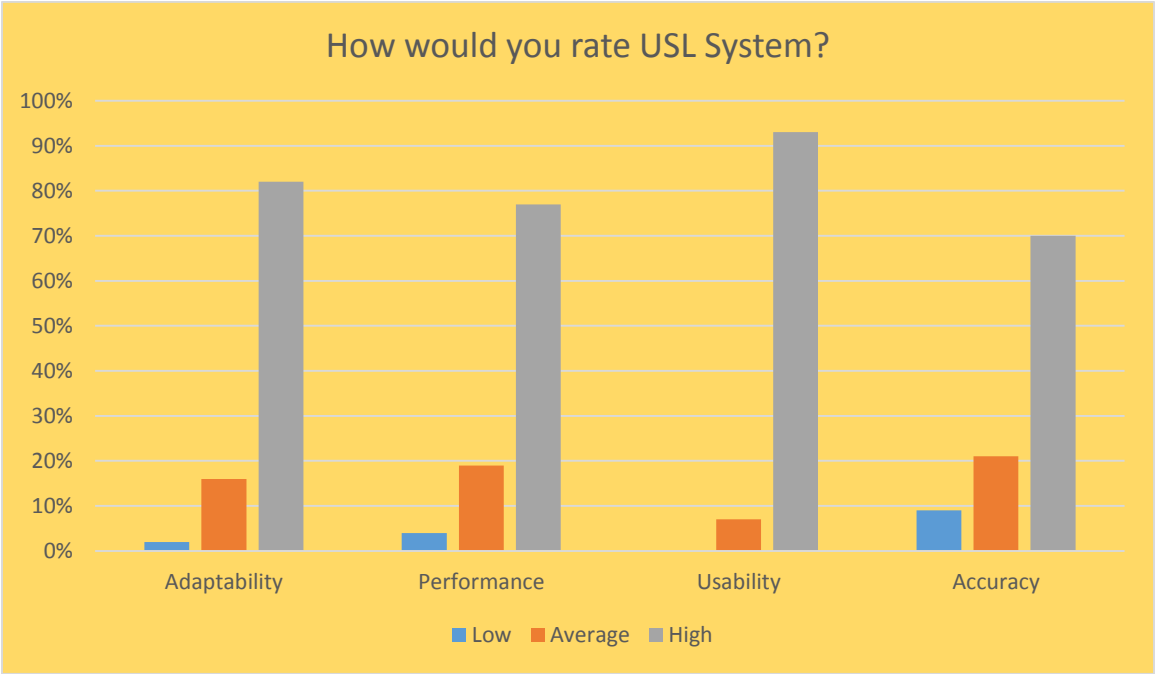


Figure 8.1- Achievements of Nonfunctional Requirements

Figure 8.1 shows that the readers were highly satisfied with the nonfunctional requirements provided by the USL system with all the evaluated nonfunctional requirements were above 70% in the high category.

8.7 Chapter Summary

This chapter started with describing evaluation criteria, evaluation methodology and selected different types of evaluators to evaluate the different phases of the USL project along with the justifications for those selections. Then Questionnaire and Interviews were selected as the evaluation approaches with justification followed by the evaluation feedbacks discussion and reviewing.

Chapter 9: Conclusion

Contains

- Chapter Overview
- Achievements of Aim and Objectives
- Knowledge from Course Modules
- Use of Existing Skills
- Learning Outcomes
- Problems Faced
- Limitations of the Research
- Future Enhancements
- Summary

9.1 Chapter Overview

The previous chapter presented the results of the evaluation process. This chapter will focus on achievements of the aim and objectives of the project. Problems faced during life cycle of the project, limitations of the project and identified future enhancements.

9.2 Achievements of Aim and Objectives

Aim

The Prime aim is build a system which can reduce the communication gap between ordinary people and hearing impaired people. In this system capture the sign symbol and it converts to the mapping word. An application where users can define gestures, as they want and map them to different language words to use it to convert gestures to sentences to overcome the sign language system's gaps for future.

The aim was successfully achieved in the project allocated time period and the prototype was evaluated by the domain experts and end users.

Objectives

Objective	Description
Prepare Terms of Reference	The Terms of Reference (TOR) defined the aim and the objectives of the research was already submitted as an assignment and also all the TOR data is included as chapter 1 of the report.
Literature Survey	Carryout a study or a literature survey on Natural User Interfaces available today, research and critically analyses existing NUI application today in the market, perform a survey among disable people to find out the easy ways for them to interact with computers. Investigate sign language systems that are available for disable people and their capabilities and drawbacks.
Study of a software development methodology	After reviewing several software development methodologies Spiral methodology was selected due to its flexibility in handling frequent requirement changes. For detailed explanation of this selection refer to chapter 3 of this document
Requirement elicitations process	Requirement gathering process for the USL system was done using the end users, domain experts and self-evaluation and many other requirement gathering techniques were used. Detailed requirement elicitation process details contains in chapter 4 of this document.
Software and hardware resources	After analyzing the requirements for the USL system most suitable technologies, tools, APIs, libraries, platforms, algorithms and hardware requirements were determined. A detailed description of the software and hardware resources selection in included into chapter 6 of this report.

Develop the prototype	Prototype was developed using identified requirements. Chapter 6 will contains the relevant code fragments, problems faced, solutions found and etc. during the development phase.
Testing of the prototype	Testing of the prototype was done using through a devised testing plan and testing was done in quantitative and qualitative aspects. This information is added to this report as Chapter 7
Evaluation of the work carried out	Chapter 8 describes the evaluation process carried out to evaluate the project and then conduct a review of the evaluation findings to determine how far the project has succeeded.
Documentation	Documentation of each phase that was carried throughout the project life cycle. Chapters from 1 to 9 contain the documentation of the each objective of the project.

Table 9.1 - Objectives achievements

9.3 Knowledge from Course Modules

MSc program modules were not directly connected to the research project. But, the following course modules gave the knowledge to successfully completion of the project.

Module	Description
Software Oriented Architecture	Web services module contributed for the designing and development of the REST web services of the prototypes' service layer.
Enterprise Development	Enterprise development module contributed for the database connectivity part and MVC application for data manipulating part.
Research Methods	Research methods module contributed to the initial project hypothesis identification and validation and later for the reviewing of various aspects related to the project.
Data Mining	Data mining module contributed to the creating a model to identify gestures more accurately

Table 9.2 - Knowledge from Course Modules

9.4 Use of Existing Skills

- Existing skills on UML diagrams such as Use Case diagrams, Context diagrams, ER diagrams and etc. were used successfully for the designing process of the system
- Existing skills on .NET framework such as C#, MVC, Services, Entity Framework used to successfully completion of the project.
- Scripting languages like Javascript, Jqury and AngularJs were used to completion of the project since they are very much familiar.

9.5 Learning Outcomes

- During the MSc program provide knowledge on various software engineering related topics and give basic understanding of them didn't enough to go and use them in a commercial project like an expert. So, since they were used in the research project, had to go on and learn deeper about them. Therefore self-learning, online documentation and discussion with domain experts were used to gain the necessary knowledge to complete this project.
- Critical thinking and formal documentation skills were developed through gradual learning and hands on experience.
- Evaluating the project qualitatively and quantitatively required sound knowledge on software quality assurance and testing skills and these knowledge were enhance through self-learning process.
- Web application hosting and other related things were self-learned and used for the successful completion of the project.
- Learn so much about disable people and what kind of difficulties they faced in the society.
- Learn about new devices like Kinect, Leapmotion and etc. and get the deeper understanding about how the works

9.6 Problems Faced

Time Constraint

In a research project the inherited risk of frequent requirement changes and lack of domain knowledge threatened to overrun the time allocated to the project. A possible solution to this problem was found by adapting the spiral development methodology, which promises to keep the development iterative way. It helped to lower the time constraints and lead the project to a successful one.

Lack of academic publications on sign language implementation

Since sign language implementation mostly done by research level it was difficult to find information about enterprise level applications. This problem was overcome by detailed self-study.

Lack of knowledge and on sign language

Just knowing there is something called sign language is not enough to go through this project. And there weren't online tutorials for sign language and also these are no class for sign language.

9.7 Limitations of the Research

LeapMotion device build on x86 architecture. It is not Raspberry Pi compatible. So, for now system cannot be configure or build as a portable device. Also the accuracy depend on the size of training data set and Leap Trainer framework. Sometimes in dark environments cause to generate inaccurate results or unrecognized the user inputs by LeapMotion device.

9.8 Future Enhancements

In this system we mainly focus on use sign language devices or techniques used for a specific sign language in the world getting to use by other sign language. In future I hope to use more powerful gesture recognition techniques. Also, I try to develop my system as a portable device using Raspberry Pi like processer. When consider accuracy, it

is more than when using a camera for capturing the motion. I hope to move my project to the area of wearable device. Like the 9.1 figure. Finally I try to build a system with minimum cost.

Figure 9.1 – Future USL System

9.9 Summary

In this chapter discuss about the flow of the project and the future work. Also mention the limitation which are face in during the process and include the conclusion of the whole process. Achievement of the object is one of the main part of this chapter and it shows the path of the project.

REFERENCES

1. Alessio Malizia, Andrea Bellucci (2012) 'The Artificiality of Natural User Interfaces', Association for Computing Machinery, available: http://dl.acm.org/citation.cfm?id=2093563 [accessed 07 August 2015]

2. Werner Kurschl, Mirjam Augstein, Holger Stitz (2012) 'Adaptive User Interfaces on Tablets to Support People with Disabilities', Mensch & Computer, available: http://dl.mensch-und-computer.de/handle/123456789/2943 [accessed 20 August 2015]

3. Marti A. Hearst (2011) 'Natural Search User Interfaces' Association for Computing Machinery, available: http://dl.acm.org/citation.cfm?id=2018414 [accessed 4 November 2015]

4. Nine percent Lankans deaf-Health Ministry [Online]. Available: http://www.nation.lk/edition/news-online/item/10242-nine-percent-lankans-deafhealth-ministry.html [accessed 4 November 2015]

5. H.C.M. Herath, W.A.L.V.Kumari, W.A.P.B Senevirathne and M.B Dissanayake (2013), IMAGE BASED SIGN LANGUAGE RECOGNITION SYSTEM FOR SINHALA SIGN LANGUAGE, Department of Electrical and Electronic Engineering, Faculty of Engineering, University of Peradeniya

6. Sri Lankan Sign Language [Online]. Available: ttp://www.lankasign.lk [accessed 4 November 2015]

7. Sri Lankan Sign Language [Online]. Available: http://www.rohanaspecialschool.org/sri-lankan-sign-language-dictionary [accessed 4 November 2015]

8. Wfdeaf.org. (2016). WFD | World Federation of the Deaf. [online] Available at: https://wfdeaf.org/ [Accessed 28 December 2015]

9. DSS of GCSC. (2016). Types of Sign Language. [online] Available at: https://dssofgcsc.wordpress.com/for-professors/types-of-sign-language/ [Accessed 28 November 2015]

10. M.wikihow.com. (2016). 5 Ways to Communicate with a Deaf Person - wikiHow. [online] Available at: http://m.wikihow.com/Communicate-With-a-Deaf-Person [Accessed 28 December 2015]

11. Mimix-Technology [Online]. Available: http://static2.docstoccdn.com/docs/149408995/Mimix-Technology [Accessed 28 December 2016]

12. Lindgaard, G. and Chattratichart, J. (2007)'Usability testing: what have we overlooked?'In Proceedings of the SIGCHI conference on Human factors in computing systems[e-journal]. pp. 1415- 1424. Available through ACM [Accessed 15 March 2016]

13. Shamonsky, D. (2016). The Idea of a Natural User Interface is Not Naturally Easy to Grasp. [online] ICS - Integrated Computer Solutions. Available at: http://www.ics.com/blog/idea-natural-user-interface-not-naturally-easy-grasp [Accessed 18 Feb. 2015]

14. WhatIs.com. (2016). What is natural user interface (NUI)? - Definition from WhatIs.com. [online] Available at: http://whatis.techtarget.com/definition/natural-user-interface-NUI [Accessed 24 Jan. 2015]

15. Pmdtec.com. (2016). CamBoard pico flexx. [online] Available at: http://pmdtec.com/picoflexx/ [Accessed 20 Sep. 2015]

16. Tobii Gaming. (2016). Tobii Gaming - Eye Tracking Enabled. [online] Available at: https://tobiigaming.com [Accessed 11 Sep. 2015]

17. Researchgate.net. (2016). ResearchGate - Share and discover research. [online] Available at: https://www.researchgate.net/ [Accessed 19 Jun. 2016]

18. Ovadia, D. (2016). Natural User Interface (NUI) in a Nutshell. [online] Onysus Software Ltd. Available at: https://www.onysus.com/natural-user-interface-nui/ [Accessed 2 Oct. 2015]

19. Theeyetribe.com. (2016). The Eye Tribe. [online] Available at: http://theeyetribe.com/ [Accessed 5 Oct. 2015]

20. Motion, L. (2016). Leap Motion. [online] Leapmotion.com. Available at: https://www.leapmotion.com/ [Accessed 13 Sep. 2015]

21. Intel. (2016). Intel® RealSense™ Technology. [online] Available at: http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html [Accessed 19 Oct. 2015]

22. Cdc.gov. (2016). RDC - Research Data Center Homepage. [online] Available at: https://www.cdc.gov/rdc/ [Accessed 19 Dec. 2015]

23. Microsoft Research. (2016). Skinput: Appropriating the Body as an Input Surface - Microsoft Research. [online] Available at: https://www.microsoft.com/en-us/research/project/skinput-appropriating-the-body-as-an-input-surface/ [Accessed 12 Dec. 2015]

24. Oculus.com. (2016). Oculus. [online] Available at: https://www.oculus.com/ [Accessed 15 Dec. 2015]

25. Mistry, P. (2016). The thrilling potential of SixthSense technology. [online] Ted.com. Available at: https://www.ted.com/talks/pranav_mistry_the_thrilling_potential_of_sixthsense_technology?language=en [Accessed 29 Dec. 2015]

26. Myo.com. (2016). Myo Gesture Control Armband. [online] Available at: https://www.myo.com/ [Accessed 23 Dec. 2015]

27. Microsoft Research. (2016). Academic Programs - Microsoft Research. [online] Available at: https://www.microsoft.com/en-us/research/academic-programs/ [Accessed 13 Dec. 2015]

28. Student Disability Services. (2016). Student Disability Services - Student Disability Services. [online] Available at: http://studentaffairs.jhu.edu/disabilities/ [Accessed 13 Jan. 2016]

29. Gaikwad, (2016). Hand talk (assistive technology for dumb)- Sign language glove with v…. [online] Slideshare.net. Available at: http://www.slideshare.net/pankajkul/hand-talk-assistive-technology-for-dumb-sign-language-glove-with-voice-38052703 [Accessed 14 Jan. 2016]

30. Planar. (2016). Touch Screen Displays, Kiosks, & Monitors. [online] Available at: http://www.planar.com/products/touch-displays/ [Accessed 23 Jan. 2016]

31. Computerhope.com. (2016). What is touchscreen? [online] Available at: http://www.computerhope.com/jargon/t/toucscre.htm [Accessed 25 Jan. 2015]

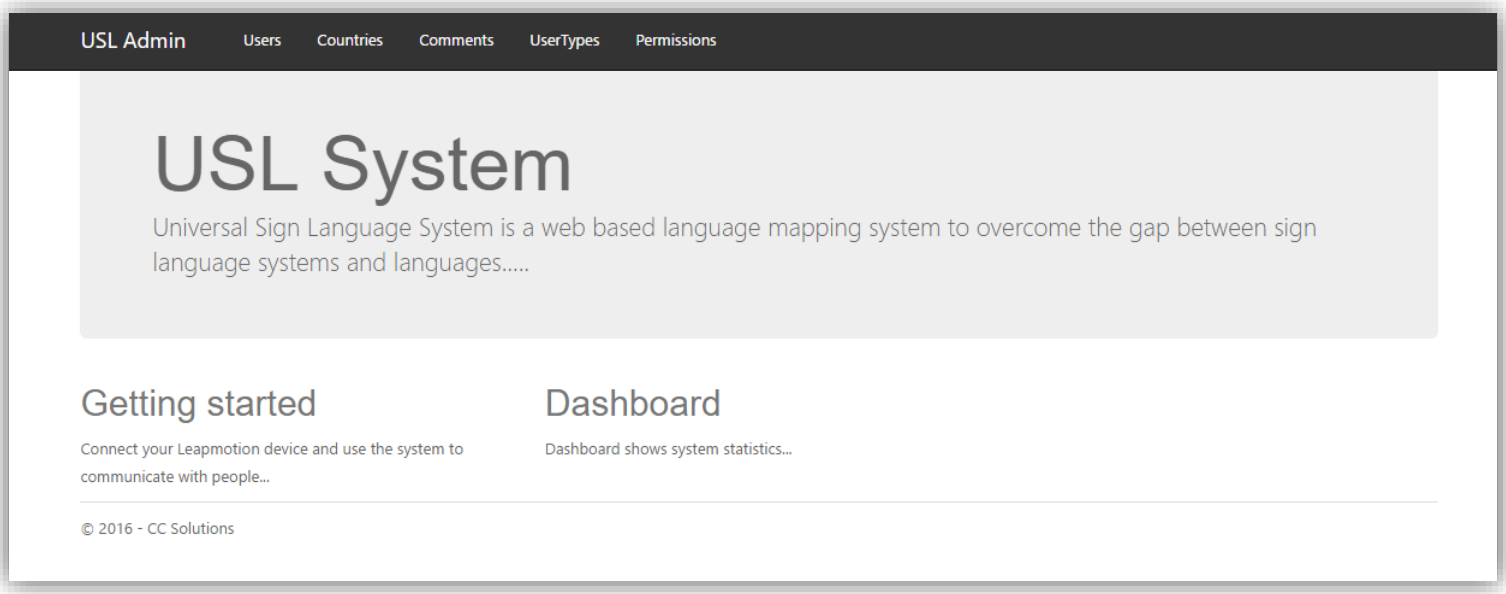
32. Microsoft Research. (2016). Muscle-Computer Interfaces (muCIs) - Microsoft Research. [online] Available at: https://www.microsoft.com/en-us/research/project/muscle-computer-interfaces-mucis/ [Accessed 21 Jan. 2016]

33. WhatIs.com. (2016). What is touch screen? - Definition from WhatIs.com. [online] Available at: http://whatis.techtarget.com/definition/touch-screen [Accessed 19 Jan. 2016]

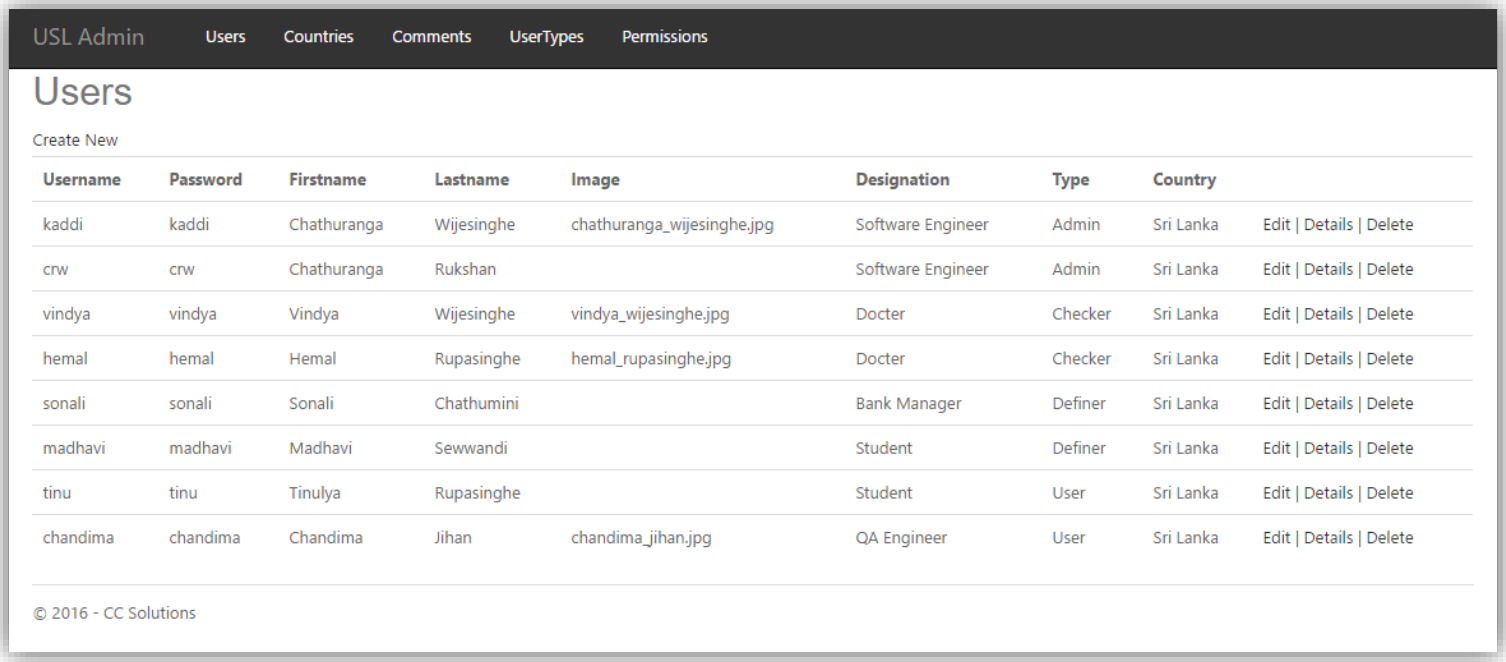
34. WhatIs.com. (2016). What is eye tracking (gaze tracking)? - Definition from WhatIs.com. [online] Available at: http://whatis.techtarget.com/definition/eye-tracking-gaze-tracking [Accessed 18 Jan. 2016]

35. WhatIs.com. (2016). What is brain-computer interface (BCI)? - Definition from WhatIs.com. [online] Available at: http://whatis.techtarget.com/definition/brain-computer-interface-BCI [Accessed 6 Oct. 2015]

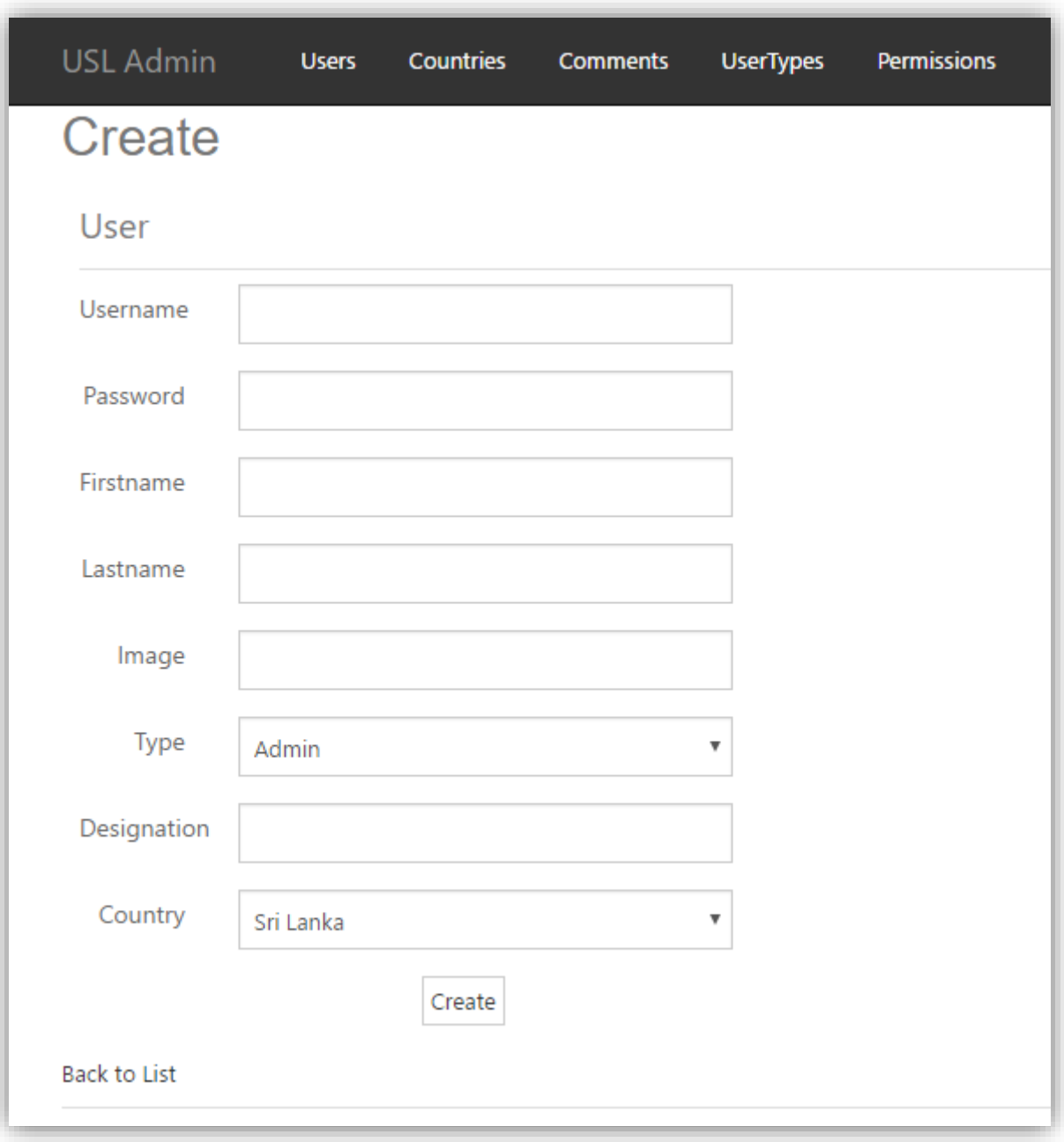
Appendix A – Admin Module Screen Shots



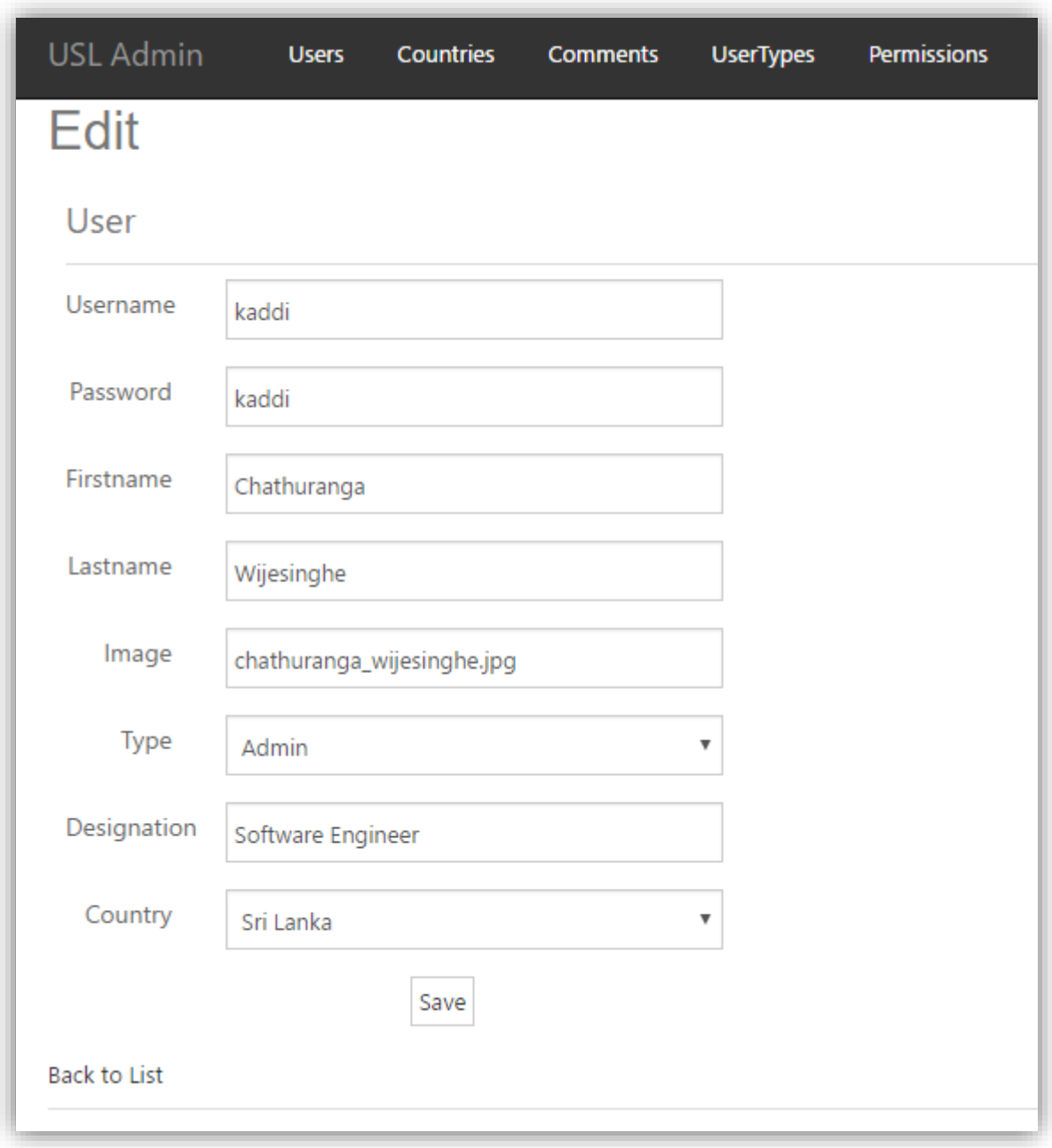
USL Admin Home Page



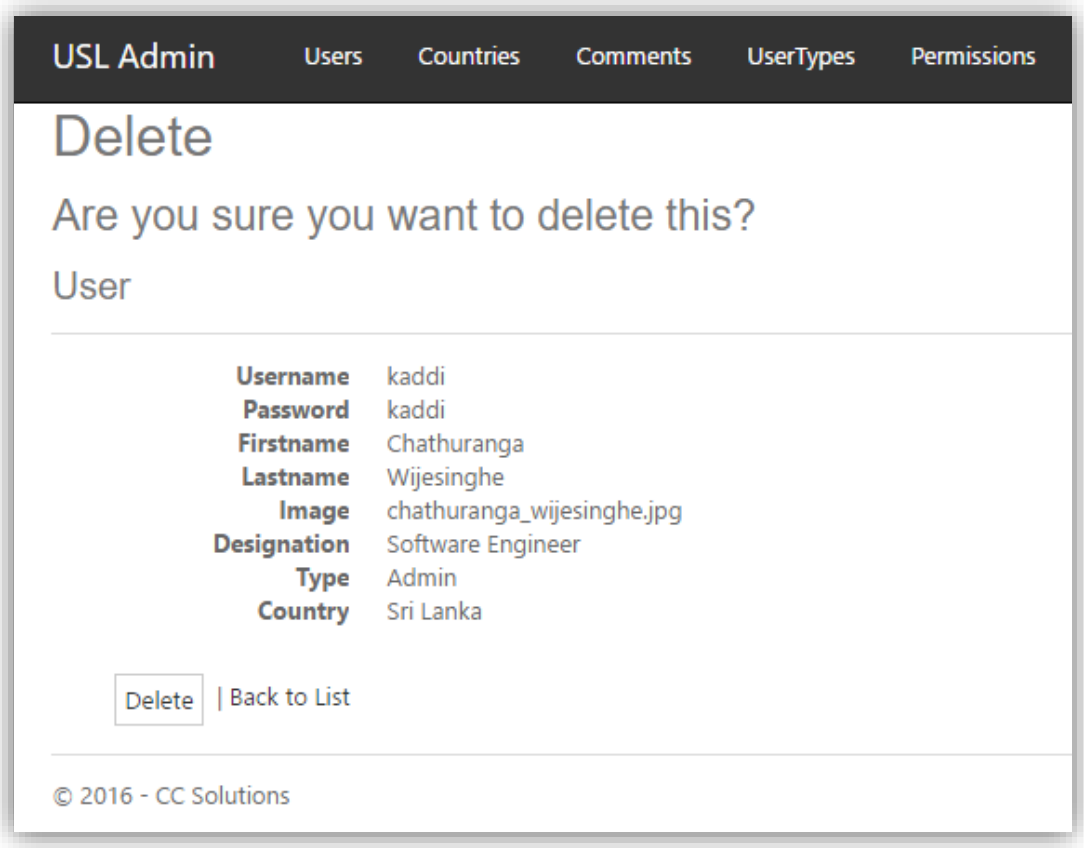
USL Admin Users Index Page



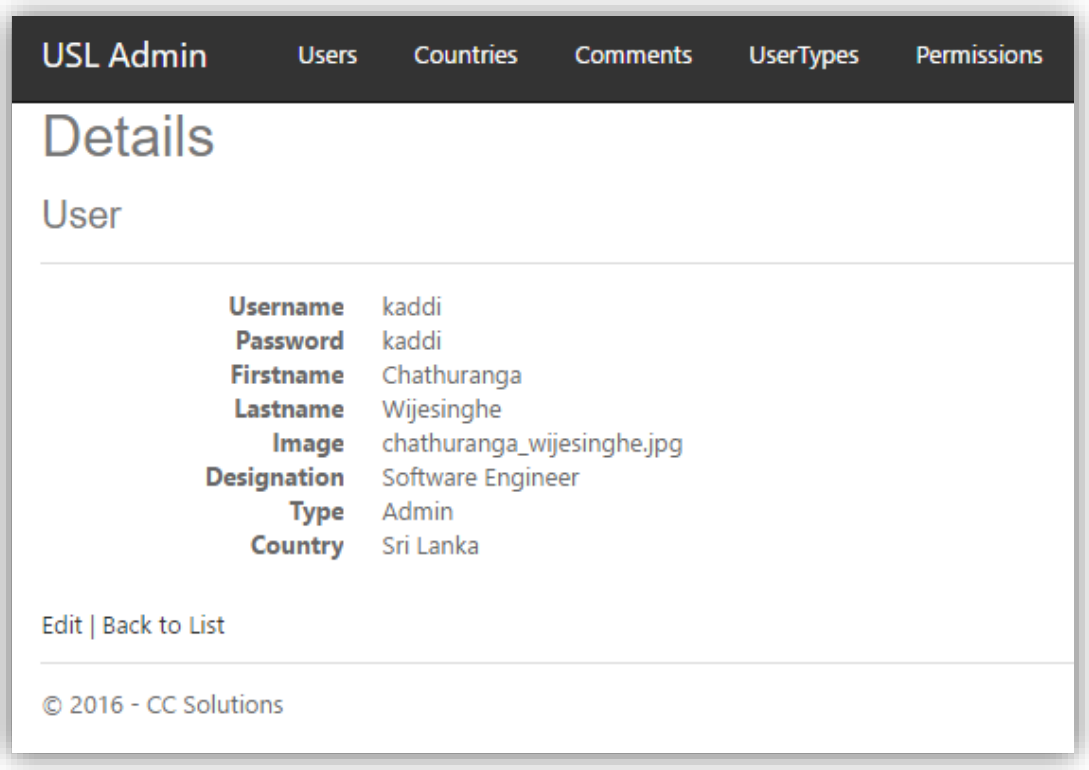
USL Admin Users Add Page



USL Admin Users Edit Page

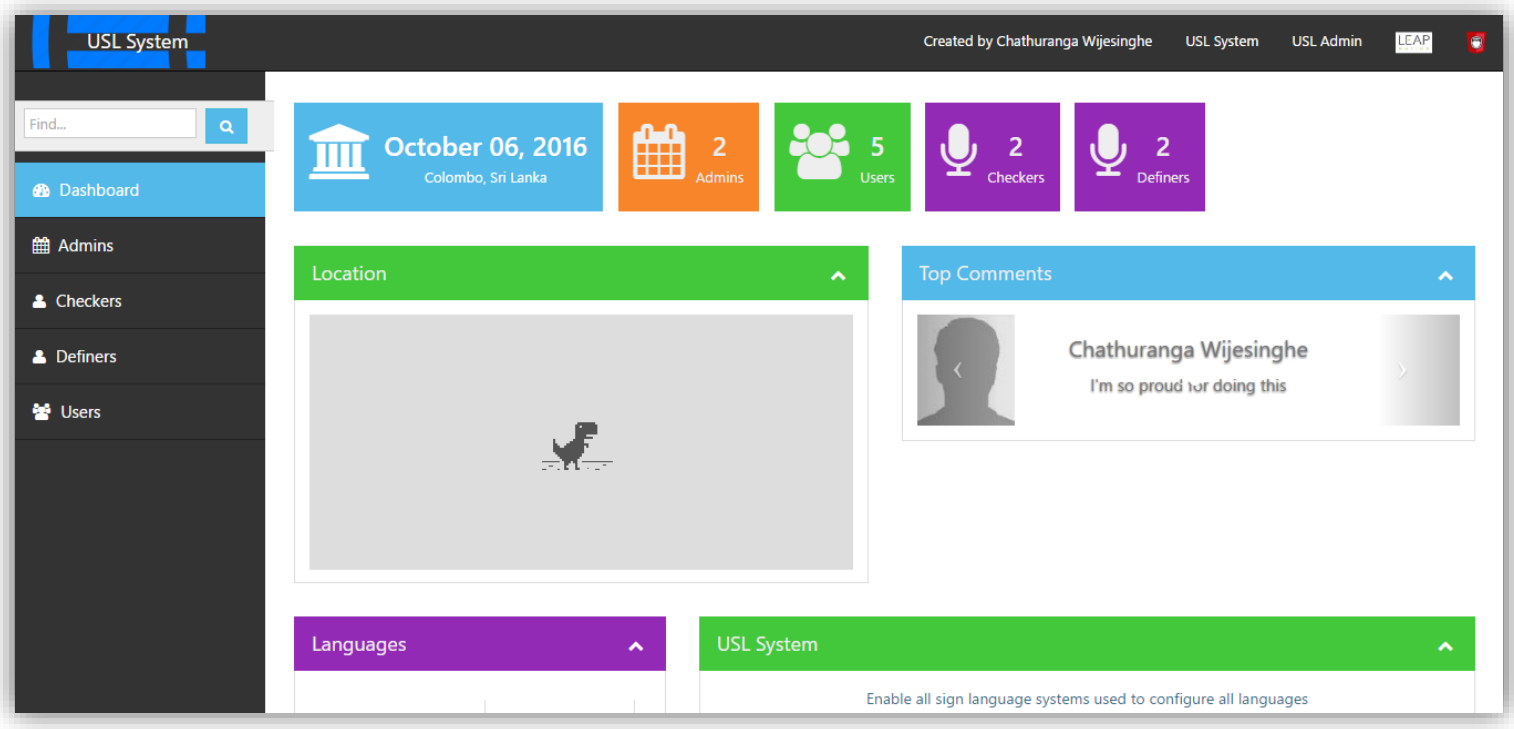


USL Admin Users Delete Page

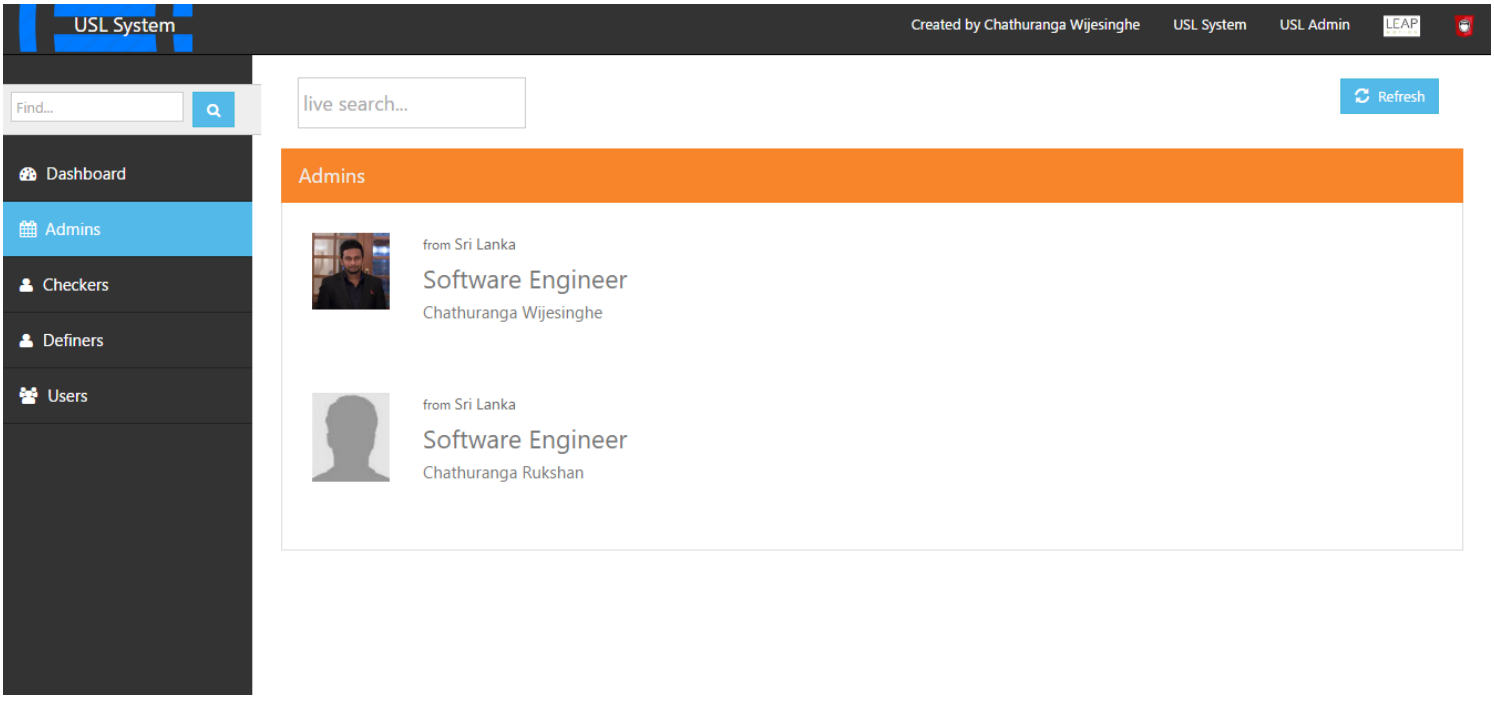


USL Admin Users Details Page

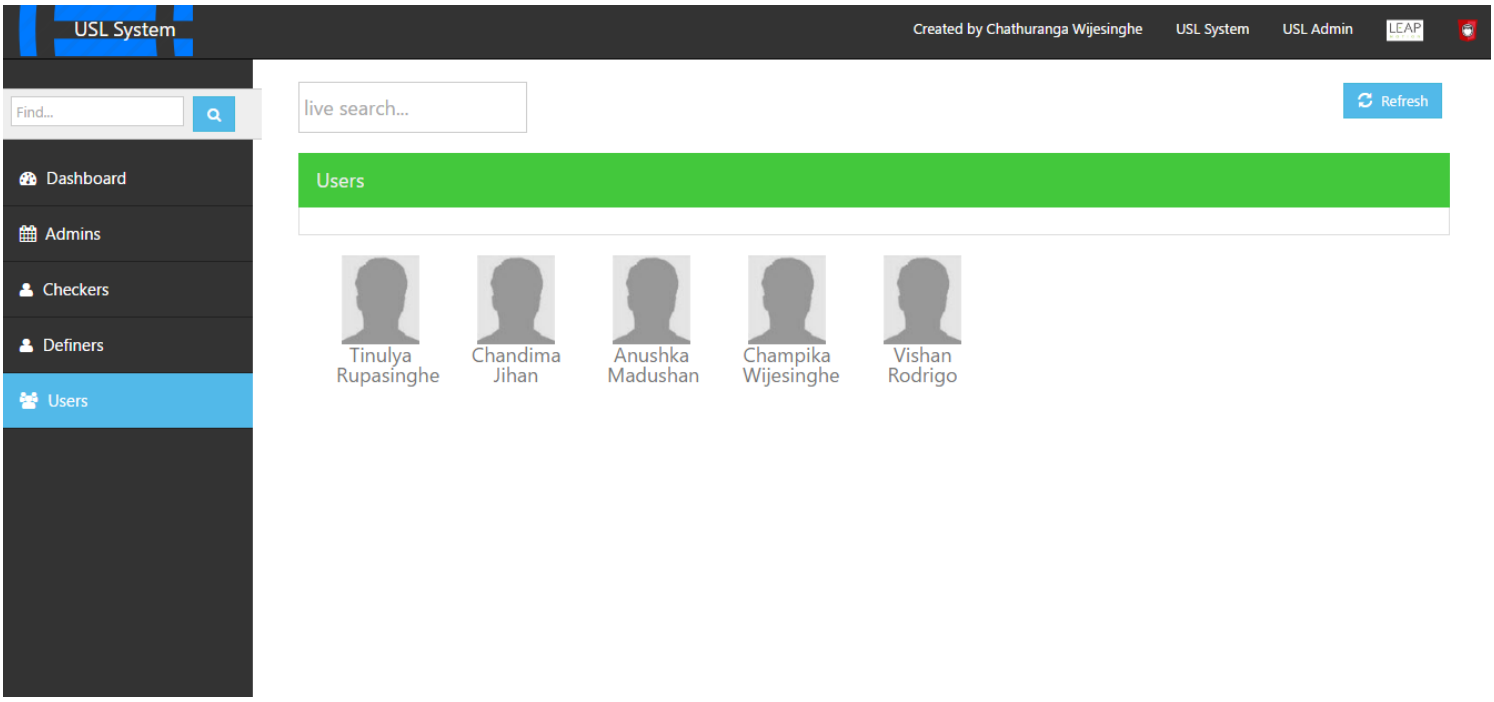
Appendix B – Dashboard Module Screen Shots



USL Dashboard Home Page



USL Dashboard Admin Page



USL Dashboard Users Page

Appendix C – Test Cases

Test Case	Scenario	Input Data	Expected Result	Actual Result	Status	Success Rate
1	Create a user	User details	Successfully save user data	Successfully save user data	Pass	100%
2	Edit a user	User edit details	Successfully save edit data	Successfully save edit data	Pass	100%
3	View users list	NA	Show users grid view	Show users grid view	Pass	100%
4	View user details	NA	Show user details view	Show user details view	Pass	100%
5	Delete a user	NA	User deleted from the list	User deleted from the list	Pass	100%
6	Create a language	Language details	Successfully save language data	Successfully save language data	Pass	100%
7	Edit a language	Language edit details	Successfully save edit data	Successfully save edit data	Pass	100%
8	View languages list	NA	Show languages grid view	Show languages grid view	Pass	100%
9	View language details	NA	Show language details view	Show language details view	Pass	100%
10	Delete a language	NA	Language deleted from the list	Language deleted from the list	Pass	100%
11	Create a permission	Permission details	Successfully save permission data	Successfully save permission data	Pass	100%
12	Edit a permission	Permission edit details	Successfully save edit data	Successfully save edit data	Pass	100%
13	View permissions list	NA	Show permission grid view	Show permission grid view	Pass	100%
14	View permission details	NA	Show permission details view	Show permission details view	Pass	100%
15	Delete a permission	NA	Permission deleted from the list	Permission deleted from the list	Pass	100%

16	Create a country	Country details	Successfully save permission data	Successfully save language data	Pass	100%
17	Edit a country	Country edit details	Successfully save edit data	Successfully save edit data	Pass	100%
18	View countries list	NA	Show countries grid view	Show countries grid view	Pass	100%
19	View country details	NA	Show country details view	Show country details view	Pass	100%
20	Delete a country	NA	Country deleted from the list	Country deleted from the list	Pass	100%
21	Create a comment	Country details	Successfully save comment data	Successfully save comment data	Pass	100%
22	Edit a comment	Country edit details	Successfully save edit data	Successfully save edit data	Pass	100%
23	View comments list	NA	Show comments grid view	Show comments grid view	Pass	100%
24	View comment details	NA	Show comment details view	Show comment details view	Pass	100%
25	Delete a comment	NA	Comment deleted from the list	Comment deleted from the list	Pass	100%
26	Create a user type	User type details	Successfully save user type data	Successfully save user type data	Pass	100%
27	Edit a user type	User type edit details	Successfully save edit data	Successfully save edit data	Pass	100%
28	View user type list	NA	Show user type grid view	Show user type grid view	Pass	100%
29	View user type details	NA	Show user type details view	Show user type details view	Pass	100%
30	Delete a user type	NA	User type deleted from the list	User type deleted from the list	Pass	100%
31	USL Admin home page	Username and password	User should be able to successfully login	Not implemented		

32	USL Dashboard home page	NA	User should be able to view dashboard page	Dashboard page shown	Pass	100%
33	USL Dashboard admins page	NA	User should be able to view the admins page	Dashboard admins page shown	Pass	100%
34	USL Dashboard definers page	NA	User should be able to view the definers page	Dashboard definers page shown	Pass	100%
35	USL Dashboard checkers page	NA	User should be able to view the checkers page	Dashboard checkers page shown	Pass	100%
36	USL Dashboard users page	NA	User should be able to view the users page	Dashboard users page shown	Pass	100%
37	USL Dashboard home page statistics	NA	All counts should equal with details pages	Count are equals with details pages	Pass	100%
38	Redirect to USL System module	NA	Redirect to USL System home page	Redirected to USL System home page	Pass	100%
39	Redirect to USL Gesture Definer module	NA	Redirect to USL Gesture Definer home page	Redirected to USL Gesture Definer home page	Pass	100%

Appendix D – Entity Data Model Diagram