20 December 2011

CENG 491

INITIAL DESIGN REPORT

Think-e
Wink-e

İbrahim Kıvanç ERTÜRK	1560192
Süleyman YILMAZ	1631215
Gizem ÖĞRÜNÇ	1631316
Aykut ŞİMŞEK	1631415

Sponsored by: innova future ready solutions

PAPAĞAN

Turkish Sign Language Recognition Using Microsoft Kinect

INDEX

1. INTRODUCTION

1.1. Problem Definition

The aim of the project is to implement an interface working through Microsoft Kinect that can recognize and understand Turkish Sign Language (TSL). We plan to divide our implementation in two main dimensions. First, by using the information we inferred from Kinect, we will be translating the meaning to text(including various languages), audio and visuals (such as videos and pictures) to provide a bidirectional communication between speech disabled people and people who doesn’t know TSL, by creating a platform that they can communicate. The second dimension of our project is to design an interactive education tool for TSL.

The communication platform will be a tool that maps basic TSL phrases to Turkish Language, and then transfers it to the second party in text, audio or visual object formats, depending on the desires of the parties.

With the educational tool we are planning to implement, we are willing to increase the number of TSL speakers, which will be containing both the handicapped people who want to communicate with the handicapped people such as their relatives/friends etc.

1.2. Purpose

According to statistics the percentage of handicapped people in Turkey is 11%. Furthermore, 36% of the total handicapped people are unable to read or write, where the number is 13% among non-handicapped people.[1] So, in order to handicapped people to communicate, either the handicapped party has to know how to write or both parties has to know the sign language. With our project we are willing to help those people by both translating their language to natural language and spreading the triteness of TSL with our educational tool in Turkey.

1.3. Scope

In software requirements specifications report, we explained the requirements that will be needed to design our project Papağan. This document shows how the software system will be structured to satisfy the requirements that specified in SRS document.

Briefly, this document includes architectural design, data design, procedural design, design constraints, time schedule etc.

1.4. Overview

In the following chapters, we will be giving information about;

- General overview of our system,
- System design considerations,
- Data design,
- The architecture of our system
- User interface,
- Libraries and tools that we used in design
- Timeline

1.5. Definitions, Acronyms and Abbreviations

TSL: Turkish Sign Language

SL: Sign Language

FANN: Fast Artificial Neural Network

SRS: Software Requirements Specifications

PAPAĞAN: Name of our product

GUI: Graphical User Interface

1.6. References

[1] T.C. Aile ve Sosyal Politikalar Bakanlığı Özürlü ve Yaşlı Hizmetleri Genel Müdürlüğü. (n.d.). Türkiye Özürlüler araştırması temel göstergeleri. Retrieved from http://www.ozida.gov.tr/arastirma/oztemelgosterge.htm

[2] Gazi Universitesi Kulak Burun Boğaz Ana Bilim Dalı, Odyoloji Bilim Dalı. (n.d.). Ülkemizde İşitme engellilerin durumu. Retrieved from http://www.odyoloji.gazi.edu.tr/isitme engellilerin durumu.htm

[3]Christopher Grant. (Dec 20th, 2010). Kinect hacks: American sign language recognition. Retrieved from http://i.joystiq.com/2010/12/20/kinect-hacks-american-sign-language-recognition/

[4]http://www.kinecthacks.com. (May 20th, 2011).Kinect sign language in arabic. Retrieved from http://www.kinecthacks.com/kinect-sign-language-in-arabic/

[5]Max Eddy. (May 19th, 2011).Hacked kinect understands sign language. Retrieved from http://www.geekosystem.com/kinect-sign-language/

[6] IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

2. SYSTEM OVERVIEW

Due to the technical capabilities of Microsoft Kinect and the amount of work can be produced by 4 people in two semesters, for now we will only be covering the basic TSL phrases, which are mostly reflected by arm movements. For the rest of the language, where the finger movements are also included, an extra and more professional approach will be required. So, for our graduation project we will design a prototype for 6 this implementation, which will mostly be handling basic arm phrases of TSL, and leave it as a generic product which can be improved by INNOVA later on.

3. DESIGN CONSIDERATIONS

Our system will be designed with an Object-Oriented paradigm. Each module will be represented by a class. Each Module will be able to communicate each other. Since our system will user interactive, we should also think about the user when we are designing Papağan. User should be able to use easily.

3.1. Design Assumptions, Dependencies and Constraints

3.1.1. Time Constraints

Pağapan has to run synchronously with the Kinect device and the user. Other modules should finish their jobs before Kinect send new data. Since user motions will be input for our systems we should be able to process 10-15 frames per second.

3.1.2. Resource Constraints

In our project we are using Kinect devise. To able to use the device Microsoft Kinect SDK should be installed on the device. It also constrains the operating system we are forced the use Microsoft's operating system since we are using Kinect SDK.

3.1.3. Performance Constraints

When the system is activated, system will be in a state that interaction users will be the most frequent event; thus, as it is stated before system should process 10-15 frames/sec for a proper recognition of the gestures.

3.1.4. Hardware Constraints

The system needs a powerful processor in order to make complicated pattern recognition and stereo matching algorithms, like processing 10-15 frames per second. We think any new generation core due processors will do the work. Also system memory consumptions never exceed 20% of the total memory. Minimum memory to system work properly is 2GB.

3.1.5. Software Constraints

The system will be implemented by using c# programming language. Kinect device will be provided by out sponsor INNOVA. GUI will be implemented by using Visual Studio 2010. Database Module interacts with a MySQL DBMS will be used to appropriate connection between the system and the database. As mentioned above operating system should be Windows 7 because we are using Microsoft Kinect SDK.

3.2. Design Goals and Guidelines

Due to Papağan's aimed purpose and targeted audience, the system mainly will be evaluated by its accuracy, usability and coverage. As the development team we will set those properties as our priorities through the system design. The explanation of those priorities can be found as below;

Papağan will be considered as successful as its accuracy to interpret TSL. The system cannot create user confidence, unless it fulfills a reasonable accuracy level. It is clear that with the increasing number of TSL moves covered by the language, the accuracy level will drop, as we can see from the similar projects [1]. For this reason, aiming 100% accuracy level will be a too optimistic perdition. But, in order to earn the user confidence, the targeted accuracy level should be above at least 90%.

As the targeted audience will include people with various disabilities, the user interfaces and system functionalities should be consistent, clean, easy to use and easy to understand. Also considering the speech and hearing disabled people, the audible interactions and materials should be kept at minimum. For points where audible materials add value to the system (such as resulting interpretations), the information should be available in alternative formats as well.

Once Papağan reaches to a desired level of performance, accuracy and can be considered as complete. The rest of the implementation will focus on enlarging the coverage of TSL by Papağan. But, for initial design the system coverage will cover the basic phrases in TSL.

4. DATA DESIGN

4.1. Data Description

This section describes data objects that will be used and managed in Turkish Sign Language Interpreter. First, we define the data objects and their attributes. Then there is a complete diagram which shows the relations between the classes.

4.1.1. Data Objects

Motion: This class is used to describe the motions of the user. The attributes and methods with their explanations are shown below.

Motion Table:

Field Name	Data Type	Decription
start position	tuple	The starting coordinates of a body-part movement. It holds three different values for x, y and z directions
end position	tuple	The ending coordinates of a body-part movement.It holds three different values for x, y and z directions
angular velocity	float	the value of angular velocity of motions
feedback	file	result of a motion in interpreter mode, and it is used as feedback message for the user
isTrue	boolean	result of a motion in interpreter mode, and it is used as feedback message for the system

Skeleton: This class is used to create the basic skeleton form of user body.The attributes and methods with their explanations are shown below.Tha data types :Head,Neck,etc; are predefined openkinect library.

Skeleton Table:

Field Name	Data Type	Decription
head	Head	It represents the user's head. It maps the head to the sensor to be tracked by kinect
neck	Neck	It represents the user's neck. It maps the head to the sensor to be tracked by kinect
torso	Torso	It represents the user's torso. It maps the head to the sensor to be tracked by kinect
leftshoulder	Shoulder	It represents the user's shoulders. It maps the shoulder's left to the sensor to be tracked by kinect
rightshoulder	Shoulder	It represents the user's shoulders. It maps the shoulder's right to the sensor to be tracked by kinect
leftelbow	Elbow	It represents the user's elbows. It maps the elbow's left to the sensor to be tracked by kinect
rightelbow	Elbow	It represents the user's elbows. It maps the elbow's right to the sensor to be tracked by kinect
leftwrist	Wrist	It represents the user's wrists. It maps the wrist's left to the sensor to be tracked by kinect
rightwrist	Wrist	It represents the user's wrists. It maps the wrist's right to the sensor to be tracked by kinect
lefthand	Hand	It represents the user's hands. It maps the hand's left to the sensor to be tracked by kinect
righthand	Hand	It represents the user's hands. It maps the hand's right to the sensor to be tracked by kinect
leftfingertip	FingerTip	It represents the user's fingertips. It maps the left of fingertip to the sensor to be tracked by kinect
rightfingertip	FingerTip	It represents the user's fingertips. It maps the right of fingertip to the sensor to be tracked by kinect

Sign: This class holds the attributes that are needed for input types. In other words, it is used to do main job on the system. The attributes and methods with their explanations are shown below.

Sign Table:

Field Name	Data Type	Decription
motion	Motion	It is used to detect user movements and pass arguments to the classes and activities that will need
audio	AudioStream	It keeps the audio stream of relevant sign
text	Text	It keeps the text form of the word of Turkish sign language
video	VideoStream	It keeps the video stream of the word of Turkish sign language. It is used in teaching mode
user	User	It keeps the information of the user that is used to track of body motion, hand sign language, to translate in text

User: This class is used to hold attributes that are info about the user. The attributes and methods with their explanations are shown below.

User Table:

Field Name	Data Type	Decription
skeleton	Skeleton	It is an object of Skeleton. It keeps the value of user body joints locations and transmit the info to the sign class over itself
mode	integer	It keeps the general info about the user is in which mode on the kinect system

4.1.2. Relationships

This part described the relationship between each class.

Sign – Motion (Has relationship):

In sign class, the motion is described as an entity. Also, it is used to translate motion to sign and then text. This relationship is needed for translation. The relationship is shown below.

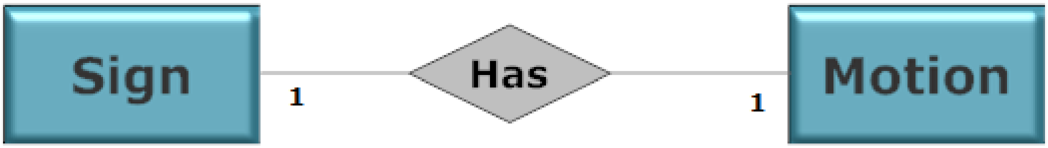

Figure 1-Sign and Motion Objects Relationship

Sign – User (Uses relationship):

In sign class, the user's information must be specified to calculate the motions' relevant meaning. So, sign class uses user class and its info. The relationship is shown below.

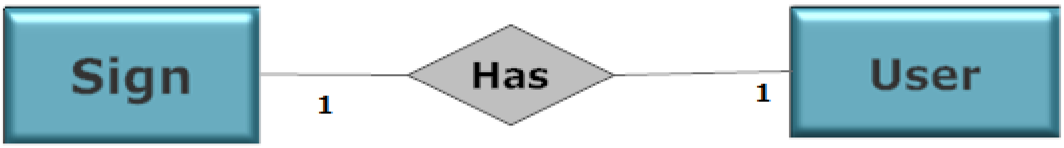

Figure 2-Sign and User Objects Relationship

User- Skeleton (Has relationship):

In user class, the user has a skeleton to recognize him to Kinect. Because of the protection of simplicity, we divided skeleton class and info to another class named Skeleton. The relationship is shown below.

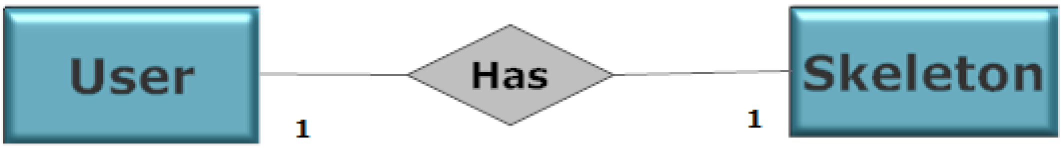

Figure 3-User and Skeleton Objects Relationship

4.1.3. Complete Data Model

Complete data model is shown below.

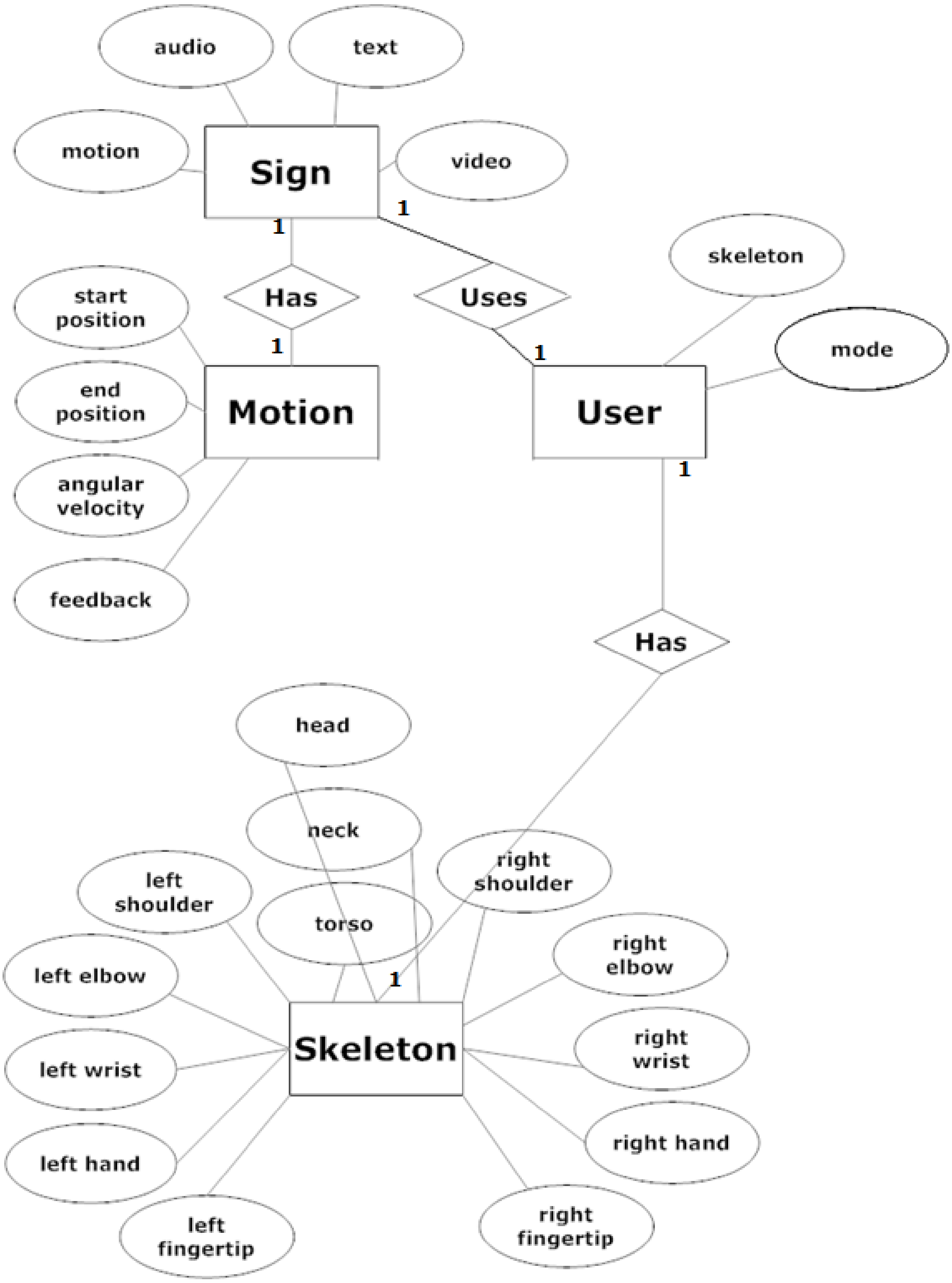

Figure 4-Complete Data Model

4.2. Data Dictionary

Skeleton: skeleton object is used to keep the basic skeleton form of user's body. The joints are used in sign language are attributes of this object. The virtual vision of object is shown below.

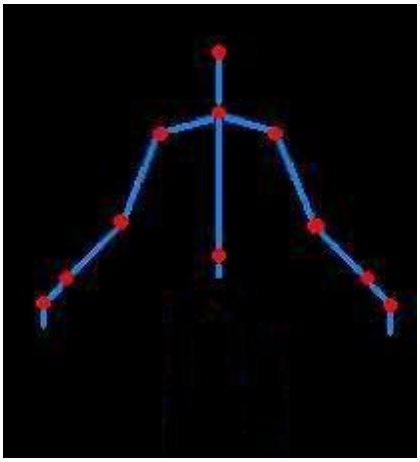

Figure 5-Basic Skeleton Model

Angular velocity: this entity holds a vector quantity which specifies the angular speed of an body part (in skeleton object) and the axis about which the object is rotating.

5. SYSTEM ARCHITECTURE

5.1. Architectural Design

Papağan will include 3 main modules that interacts each other user and database. These modules are GUI Module, Database Module and Hand Sign Recognizer Module.GUI Module is an intermediate module to provide connection between the user and Papağan. The module processes user request and passes it between the user, database, and Hand Sign Recognizer Module. Database module will hold the gestures and their videos which are handled in Papağan. This module will communicate with both hand sign recognition module and GUI module. Final module, Hand Sign Recognizer Module is the basis module that connects the user to the system. It is connected to the Kinect device in one end and to database on the other end. It receives the data of a movement on the Kinect device and interacts with the database. How these modules are connected together is presented in the following component diagram.

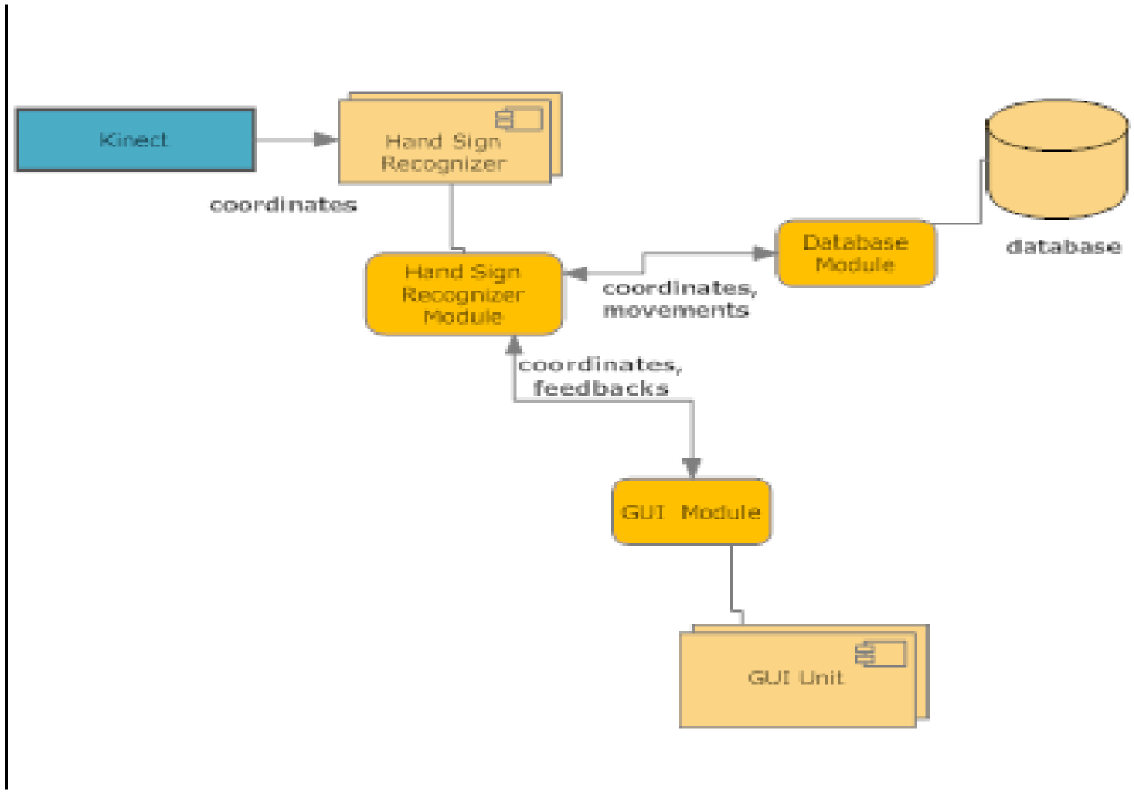

Figure 6- UML Diagram

5.2. Description of Components

5.2.1. Graphical Interface Module

5.2.1.1. Processing Narrative

GUI Module is an intermediate module to provide connection between the user and Papağan. The module processes user request and passes it between the user, database, and Hand Sign Recognizer module. GUI modules responsibilities over other modules and parties are, receiving input or providing output to the user, invoking Hand Sign Recognizer module for incoming user input TSL sign input through Kinect, and accessing the database module for retrieving information for Educational Tool Mode, where an access to Kinect or Hand Sign Recognizer module is not required.

5.2.1.2. Interface

User inputs will be coming to GUI Module from user via GUI. These inputs indicate user preferences about the screen content. The module will be outputting the screen content, according user preferences, and informing the user through the processes.

5.2.1.3. Detailed processing

The user input will be received via mouse clicks over interface buttons. Depending on the process the mouse click can activate the Kinect device and Hand Sign Recognizer module for incoming TSL movement, navigate the user to another to a different interface in its own module or retrieve user requested data from database module.

5.2.2. Database Module

5.2.2.1. Processing Narrative for Database Module

Database module will hold the gestures and their videos which are handled in Papağan. This module will communicate with both hand sign recognition module and GUI module. We choose to use this module because speed is the main issue for the Papağan. We could easily write the gestures to files and read whenever they are necessary; however, this method will slow the project. In interpreter method which we translate any given gesture if we think the worst case scenario the system may require the compare all the recorded gestures. It make sense the use of database module in that case.

5.2.2.2. Database Module Interface Description

Database module will be working with the both GUI and Hand Sign Recognition module.

GUI module can give an id for specific gesture video as input . This is input will be send when the user is in teaching mode. User selects a gesture to learn and the GUI module sends the input the database module.

Hand Sign Recognition module ask for the some gestures or all the gestures in database. That will change according to our recognition algorithm. if we can develop and algorithm that first make a preprocessing before comparing the gestures to narrow the possibilities. It will speed up the Papağan.

5.2.2.3. Database Module Processing Detail

Generally database module will process to kinds of data which are gestures and videos.

In each condition a new data is requested from the data related object for the request will be created and filled with data gathered from the database. These object will be used other modules.

5.2.2.4. Dynamic Behavior of Database Module

Dynamic behavior of the database module is explained in data design section. A data flow sequences relations and usages can be seen.

5.2.3. Hand Sign Recognizer Module

This module is the basis module that connects the user to the system. It is connected to the Kinect device in one end and to database on the other end. It receives the data of a movement on the Kinect device and interacts with the database. According to the data, it makes some database operations like comparison, insert etc. and gives feedback to the user. The data which received from Kinect device is the coordinates of the joints of a body frame by frame.

5.2.3.1. Processing narrative for Hand Sign Recognizer Module

This module receives the coordinates of the body joints for every frame. After that, the module interacts with the database. There is a comparison operation which compares the movement just received with the movements that stored in the database. According to the feedback of this comparison operation, it activates other modules on the system.

5.2.3.2. Interface Description for Hand Sign Recognizer Module

In the module, first the mode selection screen is come out with the selections of training mode and the interpreter mode. Both of these selections make this module to interact with the database.

Training mode: In this mode, there are some movements in the screen which can be learned using Papağan. After this choice, according to the choice of way which the training will occur (a video or n audio), there is a teaching video/audio come put on the screen.

Interpreter mode: In this mode, Papağan waits he make a move that will be translated. So there is just a waiting screen on interface.

5.2.3.3. Processing Detail for Hand Sign Recognizer Module

This module waits for the user to make a movement. Then the Kinect device transmits this movement coordinates to this module. Then, these coordinates are compared with the ones that stored in the database in runtime. If there is a match, it gives the meaning of the movement. Otherwise, it waits another move to interpret.

5.3. Design Rationale

We have chosen three-part module decomposition. We followed decoupling principle [2] through this selection. As the principle requires, we divided the system in three modules that should not depend on each other. Of course those modules are highly related through the processes, but the flow is through one module to another and they do not work simultaneously. Additional modules could have been represented, but to keep it simple and understandable a division of three would provide a perfect balance between decoupling the system for debugging and development and simplifying it for design and understanding issues.

5.4. Traceability of Requirements

In this section we are going the relations between our use case operations to our descriptions components.

5.4.1. GUI Module and Use Case Relations

Use case relations with the GUI Modules are shown in the table below.

Design Elements	Functional Operations	GUI Module Relation
Login Mode	Logging in the PAPAĞAN	User logins the system
	Select Mode Operation	User selects a mode
	Logout from PAPAĞAN	User logouts the system
Interpreter Mode	Returning the Main Screen	User gives input by pressing the return button
	Select Output Type	User chooses the output type from GUI

Design Elements	Functional Operations	GUI Module Relation
Teaching Mode	Returning the Main Screen	User gives input by pressing the return button
	Selecting an Gesture to Teach	User selects a gesture from the GUI.
	Play Video and Play Video Again	User presses the play button via GUI
	Taking a Test	User takes a test by using GUI
	Results and Get Feedback	GUI shows the result gathered from Hand Sign Recognition Module

5.4.2. Database Module and Use Case Relations

Use case relations with the Database Module are shown in table below

Design Elements	Functional Operations	Database Module Relation
Interpreter Mode	Translate a Phrase	Gesture datas are requested from database module.
Teaching Mode	Selecting an Gesture to Teach	Related Video about the Gesture is requested from database module
	Taking a Test	Related videos and gestures are requested from the database module

5.4.3. Hand Sing Recognition Mode and Use Case Relations

Use case relations with the Hand Sing Recognition Mode are shown in table below

Design Elements	**Functional Operations**	**Hand Sing Recognition Mode Relation**
Interpreter Mode	Translate a Phrase	The given input are processed and compared datas received from Database Module
Teaching Mode	Taking a Test	Answers are processed by Hand Sign Recognition Module Translate a Phase operation applied to each answer.

6. USER INTERFACE DESIGN

6.1. Over of User Interface

In the project one of the targeted user class is handicapped people, so to ease the usage of the program, the interface will be kept as easy as possible with limited, but appropriate functionality. The main interfaces can be summarized as followings;

6.2. Screen Images

6.2.1. Login screen

The login screen is required to calibrate the user with Kinect and the program. It does not require any kind of username/password verification, but identifies the body currently using Kinect. So, once the user is logged in, no other calibration is required through the other interactions with Kinect.

The fist image shows the initial login screen. Once the user is ready for calibration, he/she clicks on the button, and Kinect calibrates user body as shown in the following image.

When the calibration ends a confirmation message appears on screen, as shown in the last image, and the user is directed to Main Screen.

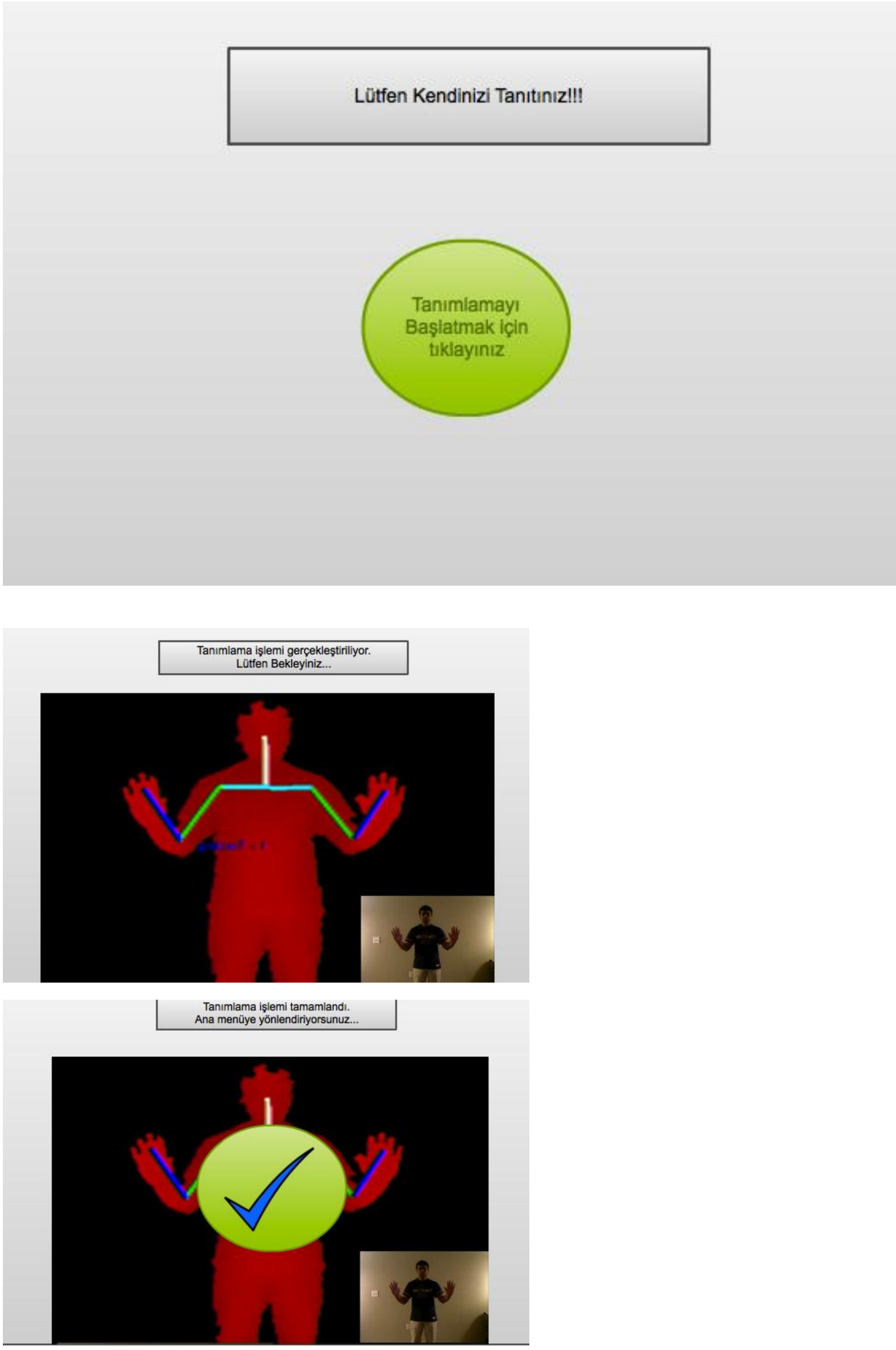

Figure 7 – Login Interfaces

6.2.2. Main Screen Menu

In Main Menu screen user is requested to select desired mode. If user selects Interpretation Mode (Çevirme Modu) he/she will be directed to a screen to input sign language movements to Kinect device. If the user selects Education Mode (Eğitim Modu) he/she will be directed to Education Mode Main Screen to select sign to learn and study.

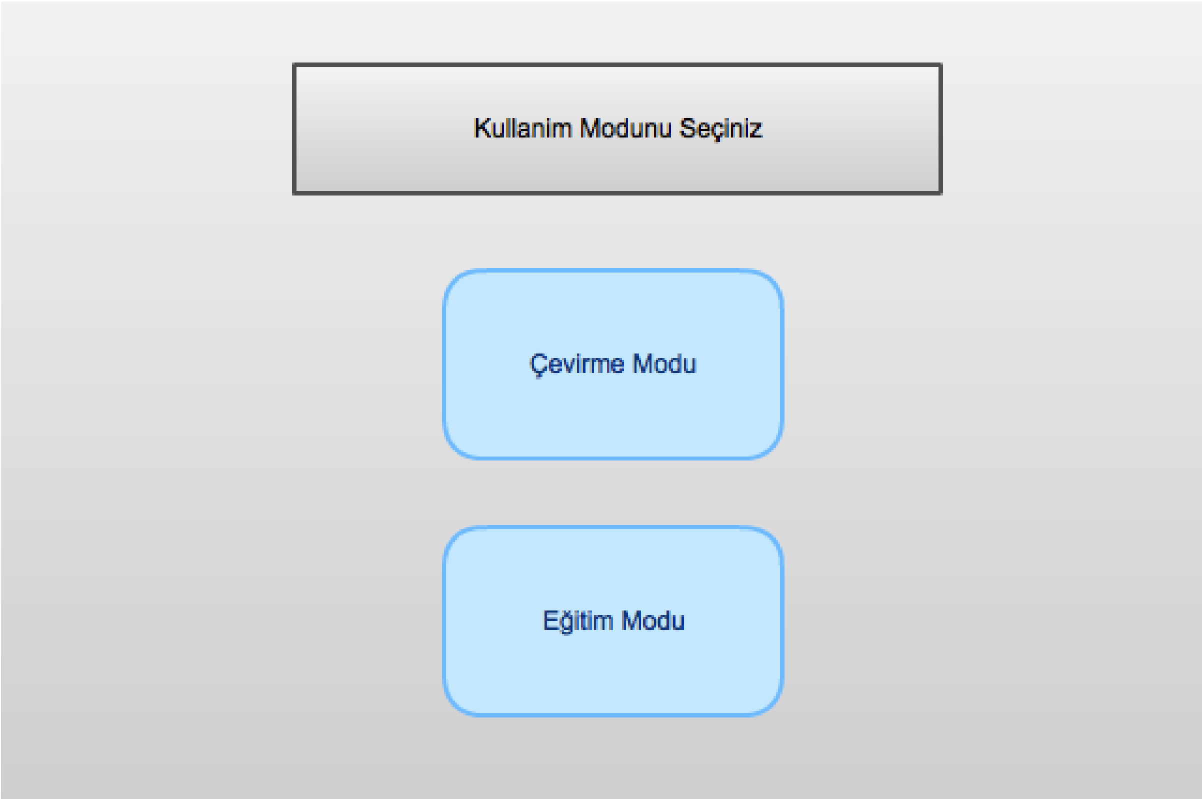

Figure 8-Main Menu Interface

6.2.3. Education Mode Main Screen

Education Mode Main Screen is the screen where user selects desired sign to exercise.

Figure 9- Education Mode Main Interface

6.2.4. Interpreter Result Screen

In Interpretation mode, after the user inputs the sign movement to Kinect, the interpreter recognizes the sign and outputs it in text format in Interpreter Result Screen. In this screen the user can hear the phrase in audio format or watch the video for it. Once the user is done, he/she can either navigate to the Main Screen or input a new sign by clicking on the relevant buttons.

Figure 10- Interpreter Result Interface

6.2.5. Educational Feedback Screen

In Education mode, once the user watches the video of the desired movement, and exercises the movement by him/herself, the user will be directed to the Educational Feedback Screen and can see how well did he/she managed to do the sign. Depending on users performance, he/she can keep practicing the same move or can navigate to the Main Screen.

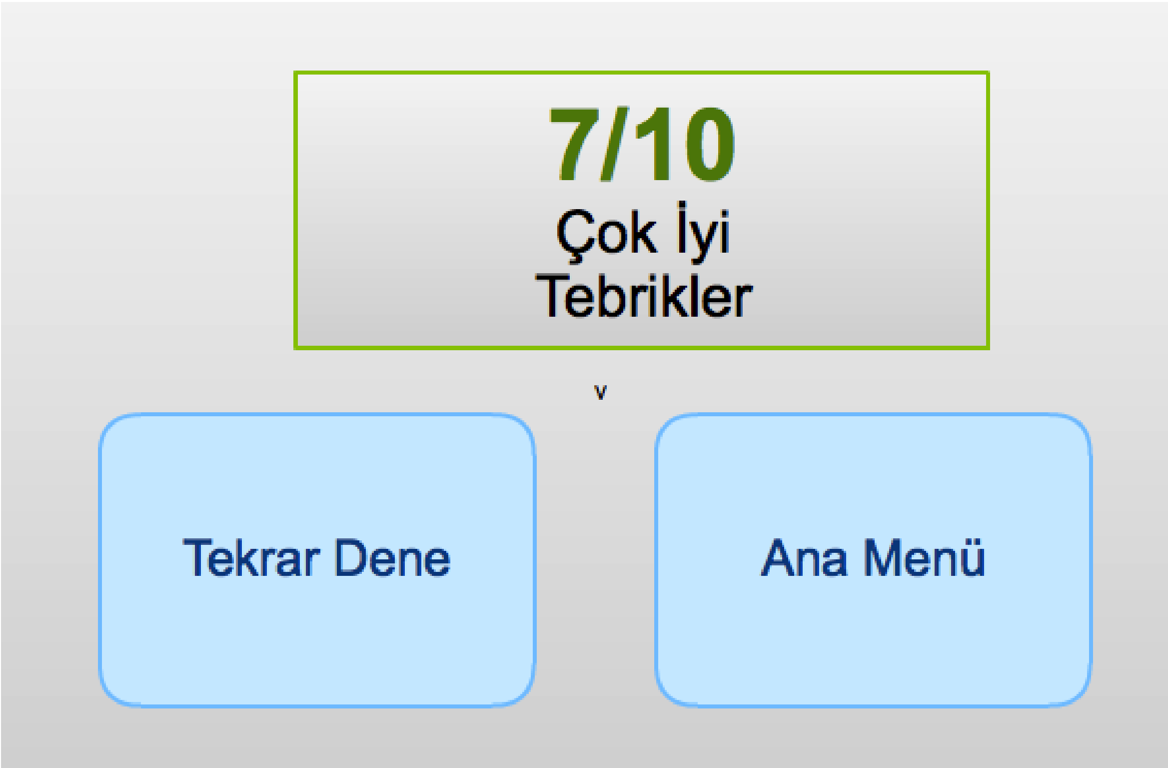

Figure 11- Education Feedback Interface

6.3. Screen Objects and Actions

When Papağan is started, there is an identification screen and a “identify yourself” object. This button is the start button because to make any movement, the Papağan first identify the user and his body joints respectively. After identification, there is the mode selection screen. In that screen, there are two objects for two modes respectively. These objects’ actions are to activate the corresponding modes in Papağan.

In the teaching mode, there are movement object that will be taught to user. Every object in the screen has the action to display the corresponding education video/audio. Then Papağan waits you to repeat your movement to be sure that you actually learned the motional meaning of the screen object. Related to your movement, there will be an opening feedback screen which evaluates your movement and there will be two objects that one returns to main menu and the other one request for another try.

In the interpreter mode, Papağan waits for the movement that will be interpreted to Turkish. If the movement is stored in the database, then the translation of this movement is reflected to the screen with two objects. One of the objects is return to main screen and the other one is waits for the new movement that will be interpreted.

7. LIBRARIES AND TOOLS

We are using Kinect device in our project. In order to get required data from Kinect we need install Microsoft Kinect SDK. This SDK can be installed only in Microsoft environment. This restricts the operating system choice to windows 7.

We are going develop our system with c# programming language. We will code in Visual Studio 2010 IDE. This IDE has required tools for GUI development. Mysql is chosen as database for our project. If we find a easy way to handle database operation via our IDE the database choice can change but since is not very critical mission it is not going to slow our development time span.

8. TIME PLANNING

In this chart whole semester is included.

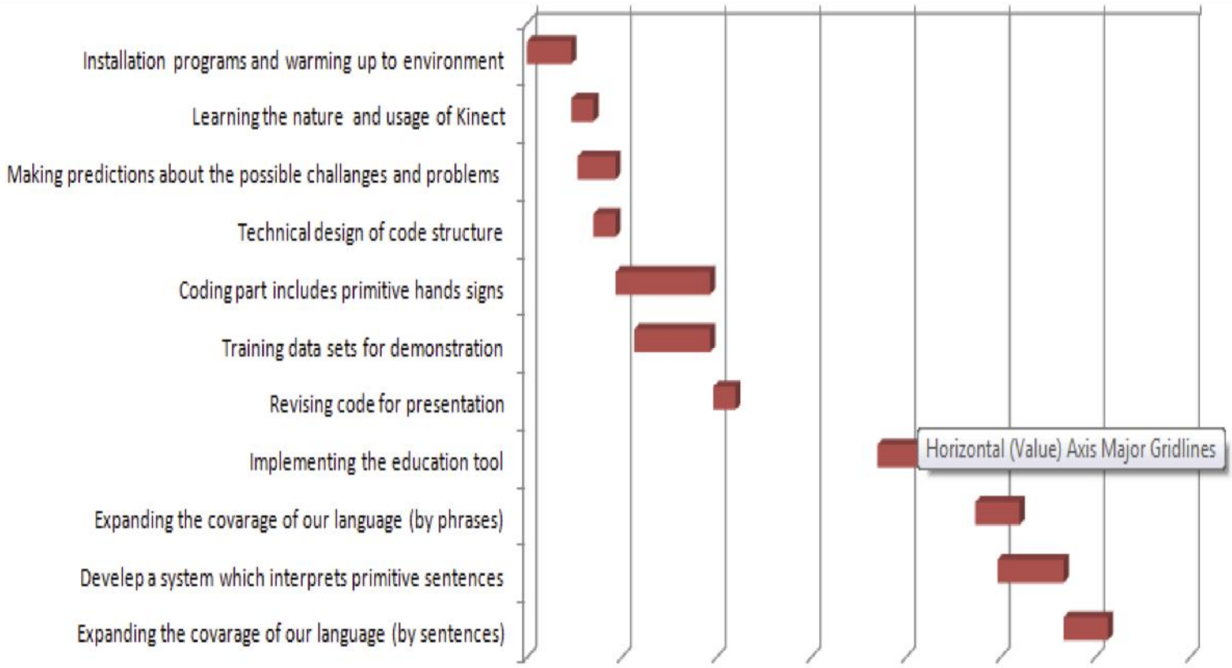

Figure 12- Gantt chart

9. CONCLUSION

This report is the Initial Design Report (IDR) document of PAPAĞAN Project. The Project purposes to understand TSL and by doing so building a bridge between handicapped people and others. It also aims to increase the usage of TSL with its educational features. The report generally aims to focus on the aspects which are considered as crucial.