Middle East Technical University
Department of Computer Engineering

Software Requirement Analysis Report

PAPAĞAN

by

Korsan Yazılım

November, 2005

Contents

1 Introduction **2**
1.1 Project Background . . . 2
1.2 Project Goal . . . 2
1.3 Project Scope . . . 2
1.4 Not in Project Scope . . . 2
1.5 Project Objectives . . . 2

2 Research **3**
2.1 Literature Review . . . 3
2.2 Customer Survey . . . 8
2.3 Contacts . . . 9

3 Requirements **11**
3.1 System Requirements . . . 11
3.1.1 Hardware Requirements . . . 11
3.1.2 Software Requirements . . . 11
3.1.3 Development Requirements . . . 11
3.1.4 Product Requirements . . . 12
3.2 Functional Requirements . . . 12

4 Data Modeling and Analysis **26**
4.1 System Data Flow . . . 26
4.2 Entity Relationship Model . . . 28

5 Project Planning **33**
5.1 Team Organization . . . 33
5.2 Process Model . . . 33
5.3 Major Constraints . . . 34
5.4 Timeline . . . 35

6 Appendix **36**

References **36**

1 Introduction

1.1 Project Background

According to the website [1] of Türk İşaret Dili, UNESCO reports that there are more than 2.5 millions deaf people in Turkey. Despite this huge number, there is a big insufficiency in deaf education. One important deficiency here is the absence of institutions and materials for TİD education. TİD has not been taught at related schools for about fifty years and the efforts to make TİD widespread are not efficient. Besides, there are not any available interactive educational programs to train hearing-impaired people in TİD. So that, there is a huge need for an interactive tool in TİD education.

1.2 Project Goal

In this project, our purpose is to develop a TİD education tool for hearing-impaired children, their parents and who else interested in learning TİD. Especially, our goal is to extend the user's TİD vocabulary by providing animations of TİD expressions supported with various lessons and games.

1.3 Project Scope

- Animations of TİD expressions
- Categorized TİD vocabulary dictionary
- Lessons and Games
- Multiple choice tests

1.4 Not in Project Scope

- Turkish to TİD direct translator (Free text analyzer)
- Speech to text translator

1.5 Project Objectives

- Present TİD expressions with 3D animations
- Develop an Editor to create animations of TİD expressions that will be used by developer team and volunteers

- Develop a categorized vocabulary dictionary presenting the animations of TİD expressions with related pictures and videos
- Develop an editor that is capable of creating vocabulary lessons from the available context.
- Create lessons, such as Basic Mathematics
- Develop vocabulary games, such as Initials and Ladders
- Develop an editor that is capable of creating multiple choice tests

2 Research

2.1 Literature Review

When we decided to implement TİD in our graduate project, first we had to make a study, especially on its basic structure and feasibility to be modeled.

During this study, it was figured out that there are neither valuable resources nor deep researches about TİD. Therefore, it will not be reasonable to expect any products serving for TİD education. Another interesting point encountered is that, sign language training was stopped in Turkey at 1953 in order to promote oral communication [1]. However, nowadays people including some academicians in Koç University tend to restart it and have already made some researches about TİD.

As a first impression, sign language sounds appropriate to be simulated in a 3D virtual environment, mainly for the following reasons:

- Facial expressions are not densely used
- Actions of signs are quite slow
- Structure of statements are simple such that you do not have to use possession suffixes for nouns or actions
- Conjunctions are not used
- There are some fixed signs for some widely used suffixes such , as ”-cı”, ”-ci”, ”-cu”, ”-cü” (the suffixes that we use to indicate occupations, e.g. ”ayakkabıcı”).

After all, we decided to analyze the software products available related to sign language education and communication in order to see what the capabilities, difficulties and deficiencies are.

1. *Vsign* [2]

   It is a 3D animation tool implemented in Macromedia ShockWave, created by EMMA (European Media Masters of Art). It models the sign animations by means of an editor.

   Vsign consists of two parts.

   - *Vsign Builder:* The editor that provides facilities to set the beginning, end and intermediate states of signs, with separate modeling for hands, body and arms. The animation is saved to text files.
   - *Vsign Player:* Plays the animation loaded form the text file.

   Vsign is a good example for our project since it contains the 3D capability, and the sign language editor. It does not have an extra hardware requirement. Furthermore, the simple file format that Vsign keeps its animations in makes Vsign an interesting project.

   The main drawbacks of the program are: the lack of education aspect, problems with user interface, and unrealistic 3D viewing.

2. *The DePaul University American Sign Language Project* [3]

   It is a large scaled and professional academic 3D project that aims to translate English to American Sign Language (ASL).

   In order to improve quality of the animations, the project emphasizes on shadows and naturalness. Shadows and different light sources are implemented to make animations look meaningful. To achieve naturalness, they repeated every animation hundreds of times in order to avoid sharp and unrealistic movements. Furthermore, project aims a successful finger spelling. Thus, transitions from every letter to other are kept in videos as AVI files, collisions are detected and fixed.

The animations produced seem elaborated and realistic. However, there are only a few sample animations in the website which does not give a concrete idea about the educational aspect and user interface of the project.

3. *eSIGN* [4]

   The eSIGN project aims to provide sign language on websites with a small software installed to clients. It uses both 3D animations and videos as the expression medium. It animates original BBC news simultaneously with a cute avatar near the news video. Besides, eSIGN provides a user friendly interface.

   One important thing about the animations is that they are created with an intelligent sign language editor. The animations are based on motion-capture data and so they are more realistic than synthetic ones. Nevertheless, the hand shapes are not caught easily since the avatar is small.

   There are some question marks about the product. There is not sufficient information about its requirements since no downloads are available on the web. Furthermore, the information about its intelligent sign language editor is not more than some screenshots which seem exciting.

4. *Reading Power* [5]

   Reading Power is a K-8 educational software product for native signers focused on literacy and reading comprehension. The software includes story telling, interactive conversation, and tools to build comprehension and vocabulary. Reading Power uses 3D signing characters to unlock the power of reading and make learning fun.

   Reading Power also includes teacher support materials, activities, a starter dictionary and ideas for integrating technology into learning.

Actually, Reading Power is very close to what we are planning to develop. Reading Power uses 3D signing characters which neglect the disadvantages of video based applications. Although Reading Power has a big plus with its 3D virtual environment, it has some shortcomings that need to be improved.

5. *SignGenius* [6]

SignGenius is a fast, interactive and cost effective software package to learn Sign Language developed by Moving Hand Enterprises and accredited by DEAFSA (South African National Council for the Deaf). It uses video clips to demonstrate sign language.

SignGenius is composed of six sections

- *Tips:* Overview of the basic hand shapes and movements that a user may need to know in order to use sign language correctly.
- *Tutor:* 2197 video clips grouped into 65 categories.
- *Test:* Monkey puzzle style test feature to test the ability in associate the video clips with the correct words.
- *Score:* For parents, teaches and students, measure progressing medium.
- *Info:* Comprehensive list of addresses of Deaf organizations, support groups etc.
- *Game:* A built-in Hangman game.

Even though SignGenius is not a derivative of what we are trying to do, it has some inspirable features. Advanced search function, user friendly interface, good categorization for tutor and tips parts are the features of which SignGenius has successful implementations.

However, SignGenius should be considered twice because of some shortcomings: low video quality, limited entertainment, insufficient education perspective.

6. *Ready Set Sign* [7]

   Ready Set Sign (RSS) is an online portal for teaching sign language, but the main products are published and sold via CDs. The portal (and disks) has many lessons and many video clips for each lesson.

   The courses are organized as if it is teaching a foreign language. The site is sincere and it is easily understandable. Iconic explanations are widely used in videos which are helpful for reminding the words.

   The site and the product uses video clips for showing motions and icons but as the quality increases it gets harder to process and download those videos. The portal has many lessons but the range is not wide, however the site is educationally well organized. These games and sincere lessons can help people learn and enjoy simultaneously.

7. *Tessa* [8]

   Tessa is a prototype for VisiCast, that is a project for translating the teletext into sign language. Tessa is used in post offices for preventing the deaf people from facing problems.

   A speech recognition tool, converts the sound into text and if desired it outputs the text on the screen. The text found can also be translated into British Sign Language (BSL).

   This application can give very realistic animations because it will be simulating a real human, since it is using motion capture. However there are no previews available, it is a predicton.

8. *Personal Communicator* [9]

   Personal communicator is a tool for Learning and Communicating in American Sign Language (ASL) developed by Comm Tech Lab in MSU. Personal Communicator uses digital video and compression technology for presenting sign language features. It has four components: a word processor, a text to sign and speech converter, an English-ASL dictionary and the ASL playroom.

The target audience is not only the people who wants to learn new signs but ones that look for fun along their way. Interaction with the objects is provided in a user friendly and cute manner.

At first glance; Personal Communicator's properties like low system requirements, high quality of its learning materials, incentive features for learning (such as classroom and futuristic screen backgrounds) stand out.

Unfortunately, these positive attributes are offset by notable shortcomings such as: lack of information about various sections of the program, the low quality videos which cannot express handshapes very well, the difficulty in interaction with the objects in the Playroom. Moreover, Personal Communicator is limited in some isolated words, extending its features is not possible.

Personal Communicator is a complete tool for sign language education and communication, although this tool is digital video oriented and not a good example for a project based 3D environment. However some of its educational properties include creative points.

Examining these eight notable products, we had a good overview over the market. After this investigation, it is useful to compare the products in a table. In the last row of the table, the features which will projected in PAPAĞAN are listed in Table 2.1.

2.2 Customer Survey

We created our survey by an online survey provider [10] and published it on the web [11] at 30^{th} of October. While working on the questions, we especially focus on the educational side of PAPAĞAN and try to reveal the views of our potential customers about this issue.

- Participant Profile

  We announced our survey at *turkisaretdili* mailing list, of which members are mainly dumb or deaf people. In addition to our hearing-impaired friends, there are also some academicians working on TİD in the group.

	3D Anim.	Video Based	Sign Editor	Educ. Tools	Dictio -nary	Free Text Analyzer	Speech Recog.
Vsign	✓	-	✓	-	-	-	-
ASL Project	✓	-	✓	-	✓	-	-
Reading Power	✓	-	-	✓	✓	-	-
Sign Genius	-	✓	-	-	✓	-	-
Personal Commun.	-	✓	-	✓	✓	✓	-
eSIGN	✓	✓	✓	✓	✓	✓	-
Ready Set Sign	-	✓	-	✓	✓	-	✓
Tessa	✓	-	-	-	-	✓	✓
Papağan	✓	-	✓	✓	✓	-	-

Table 1:

- Comments

It's not meaningful to comment on our survey results since an adequate number of participants are not reached for the time-being. However, this survey will not be wasted as we make use of it when a higher attendance is achieved.

2.3 Contacts

- *with Veysi İşler:*

As our project's subject is announced as 3D education tool, we all agreed to talk to our graphics course instructor Veysi İşler. In our first meeting, he asked what we have in mind. We stated that our priority is the realistic look of our model. He advised us to use motion capture, but also admitted that it is not feasible and easy. Then we asked Aykut Erdem about the subject and he discouraged us for it to be extremely hard and expensive to do. Returning to our meeting with Veysi İşler, he suggested us to use RenderWare for our project. It is a widely used, strong tool for designing games. Having his word for another meeting, we investigated and came up with new ideas. After explaining the disheartening motion capture issue, we asked his ideas about the skeletal animation and inverse kinematics. After sharing valuable ideas; with his feedback, we give up the idea of using RenderWare. He suggested us to look for an open source graphics library, which has an easily reachable documentation and support. Those meetings were golden opportunities for us to clear our minds about the blur areas, and reach valuable information.

- *with Engin Arık:*

  Engin Arık is a psychologist working on sign languages, currently studying his PhD in Purdue University. We noticed his name in contributors part of TİD website; contacted him via e-mail and requested his advices especially about the education aspect of our project. He replied our e-mail cordially, promised to help and wanted our proposal in order to provide us better support. Since then, we have communicated with him, explained our ideas and got useful feedbacks. The main ideas acquired from these correspondences that enlighten us in the project can be summarized as follows.

  - There is no successful project in this area so far.
  - There is no certain methodology in sign language and deaf education.
  - Although there is no certain method to teach sign language, it is a good idea to start with basic concepts like pronouns, simple sentences, numbers and basic arithmetic.
  - To get accurate sign language data, the people who are actually deaf must be chosen as models. The bilingual people which are not native signers (such as children of deaf people) are not appropriate.

  We keep our contact with him and try to make use of his experiences as much as possible.

- *with Müge Sevinç:*

  Müge Sevinç is currently preparing a master thesis on TİD at Cognitive Sciences, METU. We contacted her with the help of Dr. Cem Bozşahin and asked for an appointment in order to make things clear in our mind. It was such a beneficial gathering for us that we learned many things about the researches on TİD and realized it's very hard to implement a Turkish-to-TİD free-text analyzer, especially without a grammar. Actually, it was the day we were convinced to give up free-text analyzer facility of PAPAĞAN and started to think on a more fascinating educational atmosphere. We keep our contact with her even today and request her advices about every aspect of our project.

3 Requirements

3.1 System Requirements

3.1.1 Hardware Requirements

The only hardware requirement of the project is a standard PC the specifications of which can be described roughly as follows: Pentium 4 main processor, 512 MBs of RAM, GeForce Ti4200 Graphics Card.

3.1.2 Software Requirements

PAPAĞAN is mainly projected to run under Windows systems. There are no additional software requirements to run standard features of the project.

Macromedia Flash Player will be needed to view the lessons and a media player will be required to play supplementary videos.

Portability to other platforms such as UNIX/LINUX is not a major concern in our project, but the platform tools will be preferred as much as possible.

3.1.3 Development Requirements

The technical requirements that will be needed during the development can be stated as follows:

- *A comprehensive development platform:* Visual Studio.NET 2003 is planned to use in the project.
- *An appropriate programming language for graphical applications:* Since, C++ language meets this requirement; the project will be implemented in C++.
- *A 3D Rendering Engine with skeletal animation capability:* OGRE seems the best choice, however OpenSceneGraph and RenderWare are still among alternatives in case of a problem
- *3D Modeling Tools:* 3D Studio Max or Maya can be used for modeling. MilkShape3D or any similar tool can be used for skeleton modeling

3.1.4 Product Requirements

By the end of the project, the product has to satisfy the following requirements:

- Clear and realistic animations
- Friendly and easy-to-use interface
- Extendible dictionary and lectures

3.2 Functional Requirements

- Whole System

  The diagram above (Figure 1) gives a brief description of how our project communicates with its users.

  1. User shall select Dictionary
  2. User shall select Vocabulary Lessons
  3. User shall select Games
  4. User shall select Lessons
  5. User shall select Evaluation
  6. User shall select Editor

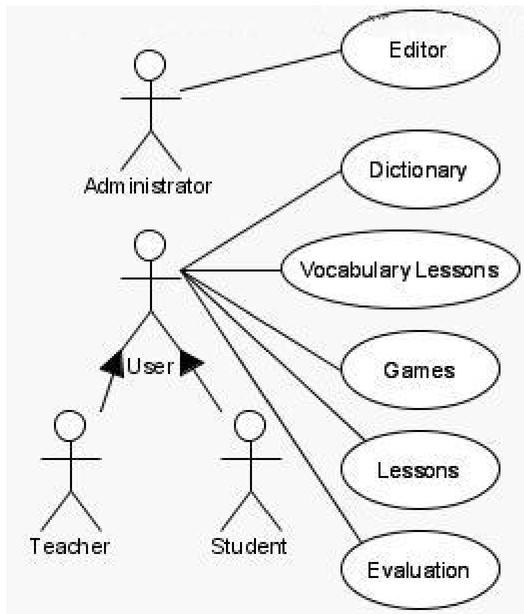

Figure 1: Main Use Case Diagram

- Editor

Editor is a development tool which will be used by the project team (Also, anyone who is interested in this study) to animate the 3D character model and insert the animation into the database with related pictures, video files and a label defining the sign language expression. For convenience, Editor will be described with the help of a GUI. First a conjectural GUI will be presented, and then the features will be specified according to this GUI.

The 3D character model visualization window (Actually it is the Player with a few additional features) is at the upper left corner. The main feature of this window is to display the model, and the animation that is being created. It is possible to select a joint for modification by clicking on a red spot on the model. Also, the model can be roughly modified by dragging and dropping from these red spots according to the inverse kinematics' rules. Model can be viewed in any direction

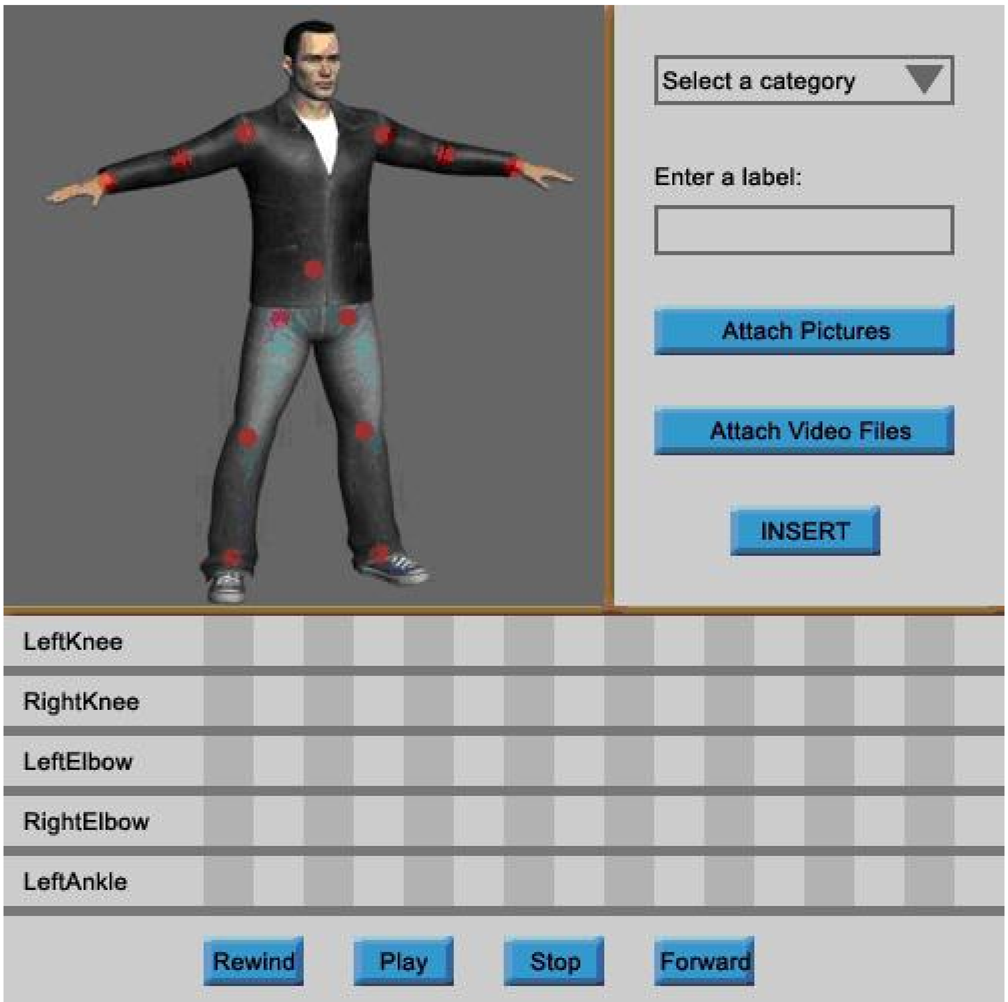

Figure 2: GUI of the Editor

(direction can be changed by left clicking on an empty space and moving the cursor) and distance (distance can be changed by right clicking on an empty space and moving the cursor).

Right below the *Model Visualization Window*, timelines for each of the joints (there are many other joints that are not represented in

Figure 2) reside. A timeline represents the changes in the properties of a joint with respect to time. Of course it is not just for representation, durations of the actions can be modified by dragging and dropping the time marks in the timeline. Actually, timeline feature is not a new idea, it is also used in Macromedia Flash.

Under the timelines, there are the *Rewind*, *Play*, *Stop*, *Forward* buttons which control the Player.

And finally, the insertion panel which consists of *Category Dropdown Menu*, *Label Text Field*, *Attach Pictures* button, *Attach Video Files* button, and *INSERT* button is at the upper right corner. From this panel, an animation that is created with the Editor can be inserted into the database with a label describing it. Also, it is possible to add several pictures and video files related to the animation.

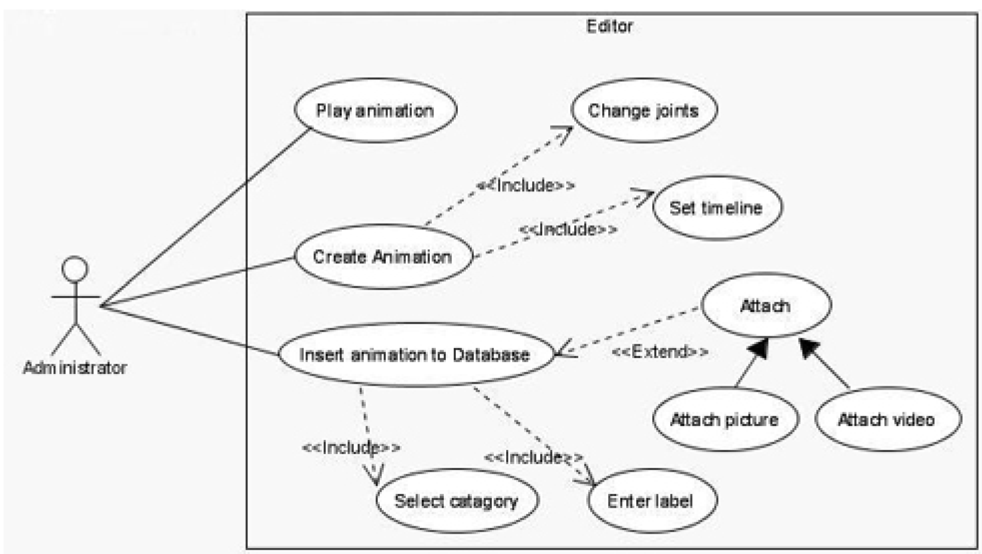

Figure 3: Editor Use Case Diagram

1. User shall select a joint of the skeleton.
2. User shall view the properties of a selected joint.

3. User shall change the properties of a selected joint.
4. User shall change the pose of the model by selecting a joint and then changing the properties of the selected joint.
5. User shall change the pose of the model by dragging and dropping a joint of the skeleton.
6. User shall create an animation by moving the time pointer forward and then changing the pose of the model.
7. User shall drag and drop the time marks on the joint timelines.
8. User shall play the animation visualization by pressing the *Play* button.
9. User shall stop the animation visualization by pressing the *Stop* button.
10. User shall rewind the animation visualization by pressing the *Rewind* button.
11. User shall forward the animation visualization by pressing the *Forward* button.
12. User shall change the camera view.
13. User shall select a category for the animation from the *Category Dropdown Menu* or create a new one.
14. User shall write a label for the animation.
15. User shall attach pictures to the animation by pressing the *Attach Pictures* button.
16. User shall attach video files to the animation by pressing *the Attach Video Files* button.
17. User shall insert an animation into the database with the specified label and related pictures, video files by pressing the *INSERT* button.

- Dictionary

Dictionary of PAPAĞAN can be thought as a basic vocabulary guide for our users who wants to learn TİD. Actually, what we mean by dictionary is a slightly modified version of ordinary ones with additional features that we hope, to create a better education atmosphere. For convenience, a conjectural GUI (Figure 4) will be presented now, in order to present the big picture.

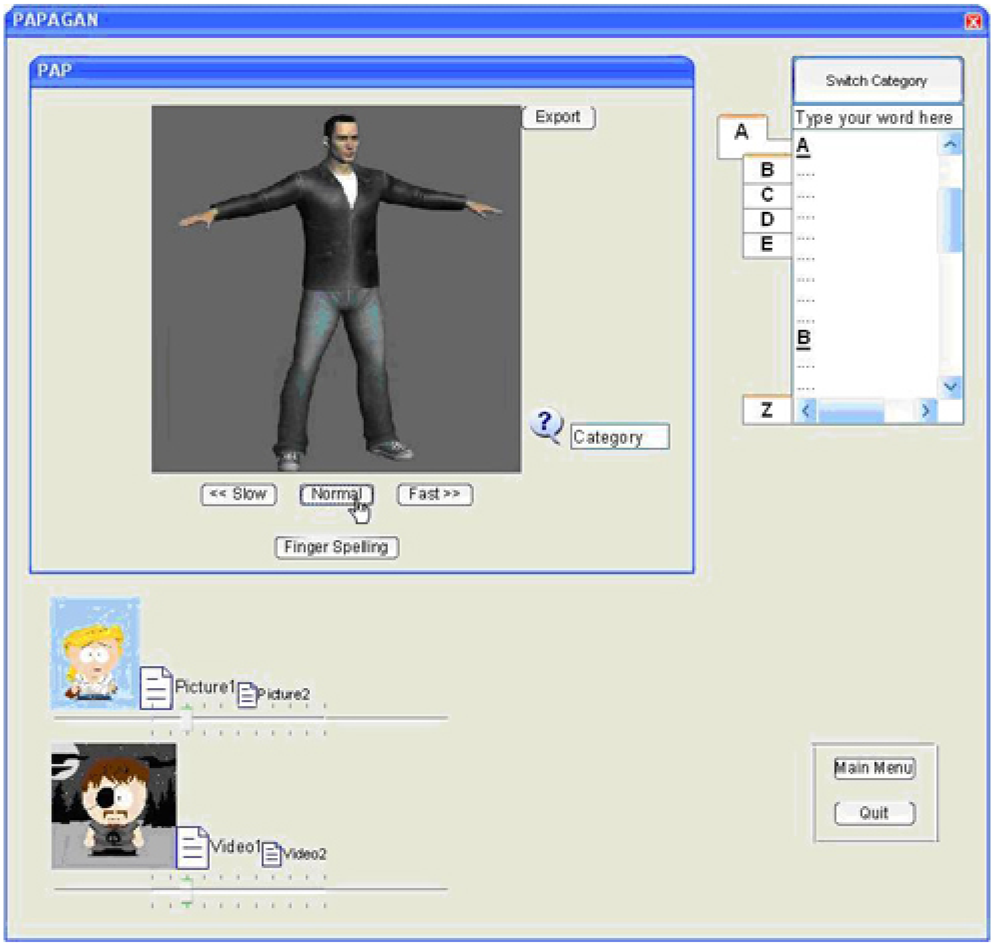

Figure 4: GUI of the Dictionary

The interface of words or phrases is at the upper right corner. User searches for the desired animation here and that desired animation is played in PAPAĞAN Animation Player (PAP) as it is shown with a dummy model. In addition to the original animation, user can watch the finger spelling of selected word. Figures, below the PAP, are the relevant pictures and videos attached to that word or phrase in order to facilitate remembering for users. As it's mentioned later, animations can be exported a video file and can be used in different applications such as in a Macromedia Flash animation or Microsoft Power Point presentation. The category box shown in PAP, near the question mark icon, is just for giving information about the currently animated word or phrase.

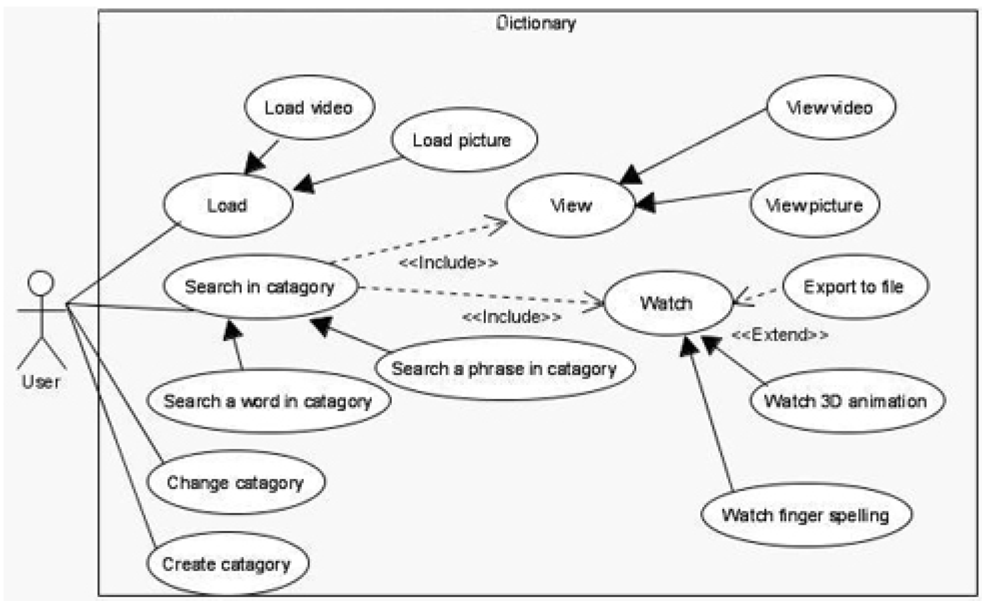

Figure 5: Dictionary Use Case Diagram

1. User shall search a word or a phrase.
2. User shall view the animation of selected word or phrase.
3. User shall change the speed of selected animation in PAP.
4. User shall view the related pictures or videos of selected word or phrase, if available.
5. User shall view the finger spelling of selected word.
6. User shall export the selected animation to a video file.
7. User shall view a warning message with an animation in PAP, when the desired word or phrase is not found.
8. User shall switch from alphabetical order to categories and vice-versa.

9. User shall create a new category.
10. User shall add the desired words or phrases under the new category.
11. User shall add new pictures or videos for a desired word or phrase.
12. User shall bind a word or phrase to an old category.
13. User shall delete a category and free the words or phrases under that category.
14. User shall return to main menu.

- Lessons

It's essential to have built-in lessons in an educational tool. For this purpose, we decide to develop some kind of a classroom atmosphere that includes *Basic Mathematics* lesson. However, this atmosphere does not truly implemented by us because we will use Macromedia Flash to embed our exported animations and create a more professional lesson concept. One important point about this choice is to have portability such that anyone who wants to contribute to this section can easily create lessons and share it on the web. Actually, as it is mentioned above, what left to the user is to play our *Basic Mathematics* lesson.

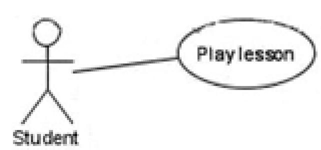

Figure 6: Lesson Use Case Diagram

1. User shall play the lesson.
2. User shall quit the lesson.
3. User shall return to main menu.

- Vocabulary Lessons

  Vocabulary Lessons are the lessons, the material of which comes from the dictionary items directly. The aim of Vocabulary Lessons is doing vocabulary training in an organized way supported with videos and pictures.

  One major difference from the Dictionary is Vocabulary Lessons are kept in separate files rather than the main database. The point in doing so is making the lessons portable from one computer to other.

  Vocabulary Lessons concept includes both manipulating (creating / editing / viewing) lessons and viewing them.

  Vocabulary Lessons are created/modified through *Vocabulary Lesson Editor* and viewed through *Vocabulary Lesson Viewer*.

  – *Vocabulary Lesson Editor*

    Through this tool, user can see either available words in entire dictionary or words grouped according to their category. User cans select a word and add it to the lesson, append some or all of the video and pictures of the added word to the lesson. Can move up / down the word in the lesson and remove a word from the lesson.

    User can name a lesson and add a brief description for it. After preparing the lesson user can save it to a file.

  – *Vocabulary Lesson Viewer*

    In the viewer the user views the items in the Vocabulary Lesson one by one. User can click to view sign animation fingerspell of the item through the animation player, see its pictures and play its movies if available. User can move to previous or next item in the lesson, rewind the lesson jump to any item in the lesson.

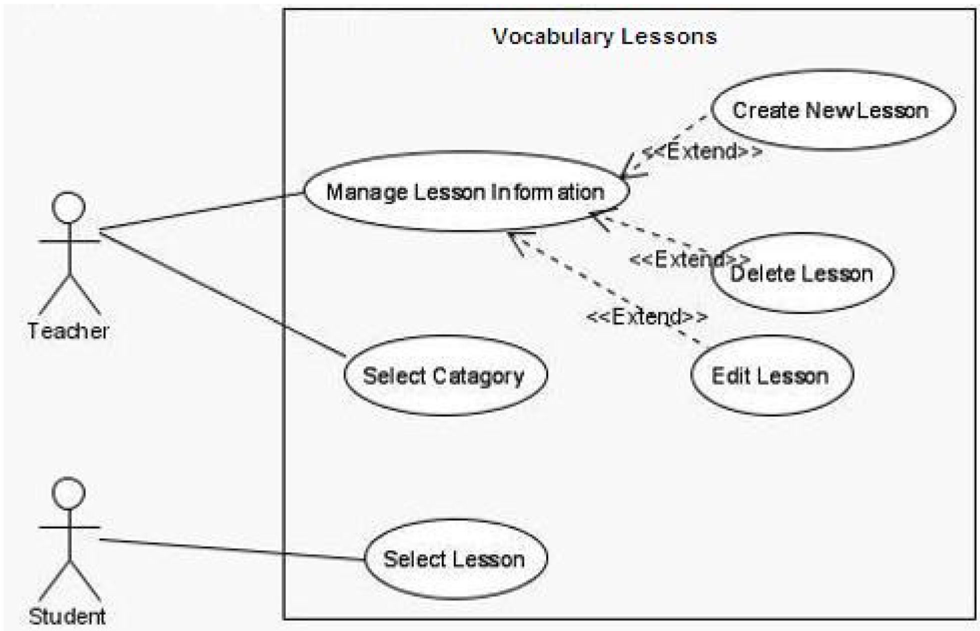

Figure 7: Vocabulary Lesson Use Case Diagram

An important point here is that user can use all the functionalities of PAP.

– Vocabulary Lesson Viewing Main Screen:

1. User shall select a lesson from available lessons.
2. User shall view the data of the selected lesson.
3. User shall proceed to the *Vocabulary Lesson Viewer*

– Vocabulary Lesson Viewing Tool:

1. User shall view the available pictures, videos, fingerspell options of the word.
2. User shall click on the word to view the see its sign language animation through PAP.
3. User shall select an available video of the lesson to view.

4. User shall select an available picture to view.
5. User shall click the option to see the fingerspell animation through PAP.
6. User shall control the functionalities PAP.

- Vocabulary Lesson Editor Main Screen:

1. User shall create a new lesson.
2. User shall edit a lesson.
3. User shall delete a lesson.

- Vocabulary Lesson Editing Tool:

1. User shall select a category or entire dictionary to view.
2. User shall select a word in the vocabulary list.
3. User shall view the related videos of the selected words in the integrated environment.
4. User shall view the related pictures of the selected word.
5. User shall view related fingerspell animations of the word through PAP.
6. User shall view the sign animation of the word through PAP.
7. User shall add the selected word to the lesson.
8. User shall select and add videos, pictures and fingerspell animations of the word to the lesson.
9. User shall delete a word from the lesson.
10. User shall delete a picture, video, fingerspell of the word from the lesson.
11. User shall add title to a lesson.
12. User shall add description about the lesson.
13. User shall save a lesson.

- Evaluation

PAPAĞAN is considered to be a complete sign language education tool. For achieving this aim, we decided to develop an evaluation system for users. There are two main parts of our system; test based interactive exams for evaluating student's sign language knowledge level and exam editor for teachers. Indeed, we provide flexibility for teachers. A teacher may want to create his own exam that has a content parallel to his lectures. He can enhance exam content with animations, images and videos taken from database.

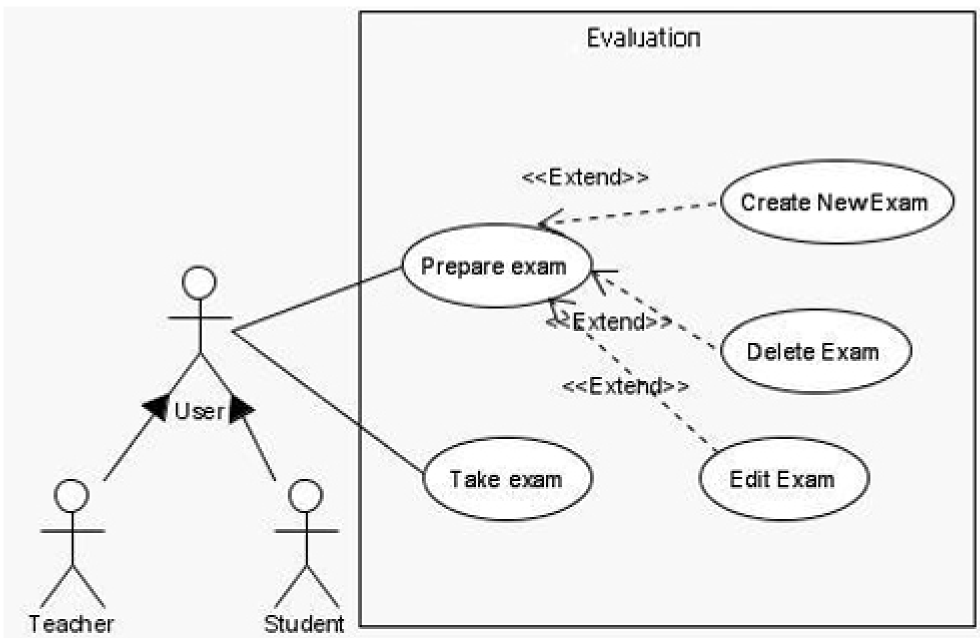

Figure 8: Evaluation Use Case Diagram

1. User shall go to prepare exam window or take an exam from the main evaluation window.
2. In prepare exam window, user shall specify exam duration and number of questions in the exam.
3. In prepare exam window, user shall select question template for each question.
4. In prepare exam window, user shall specify the question content after selecting question template for a question.
5. User shall specify 3D animation content by using PAP in the template question.

6. In prepare exam window, user shall save new prepared exam.
7. In prepare exam window, user shall load and edit a previously prepared exam.
8. In take an exam window, software shall display all existing exams.
9. In take an exam window, user shall select and start an existing exam.

- Games

What we are planning to do as games are mainly vocabulary training tools. They are supported by animations which helps the player guess a required word. These games should not only be easily played but also have a ranking list for making people compete.

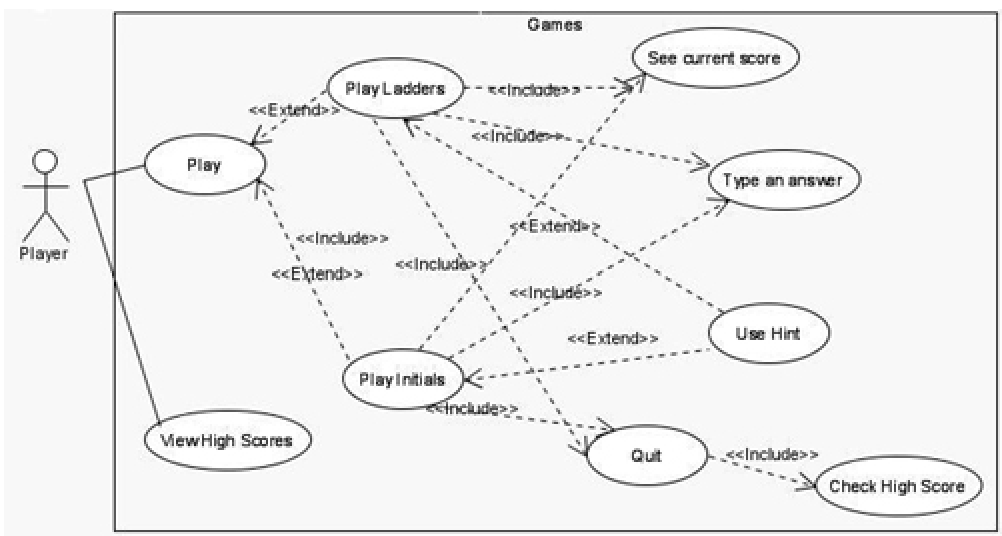

Figure 9: Games Use Case Diagram

– *Initials:*

This game is triggered with a letter and gets a word with that initial and the following word must start with the latter's last letter. There should be timer counting down, when a correct guess is made, more time should be awarded. The more time left, the more the points gained. If a hint is used points should decrease. Also no word should be observed twice in a game. If no letter exists with the last words last letter, a random word should be chosen and the user should be warned about that, and should be given the initial of that word (or the game can be over). If the user has a high score he must give his name to the application for updating the high score before exit.

– *Ladders:*

  A random letter from a word burst into a new word perpendicular to that word. There is a grid like a spreadsheet and the word fits on that grid letter by letter. If the user has a high score he must give his name to the application for update before exit.

1. User shall open a new game.
2. User shall view the high scores.
3. User shall view the animation of the awaited answer.
4. User shall guess the answer.
5. User shall use the first hint, which is the category of the word.
6. User shall use the second hint, which is the fingerspell of the word.
7. User shall see if he succeed or not.
8. User shall see his scores.
9. User shall enter a nickname if he has a high score.
10. User shall quit the game.

4 Data Modeling and Analysis

4.1 System Data Flow

- Level 0 DFD

Level 0 DFD of PAPAĞAN, shown in Figure 10, is just the big picture of our system. Major external entities, *User* and *Administrator*, with data stores, *Dictionary Data Store* and *Lessons Data Store*, are specified in this level. Here, *User* refers to the end-user who can run the program and *Administrator* refers to the person who can design animations. One important point about data stores is, while *Dictionary Data Store* is a real DBMS, Lessons Data Store is composed of files,

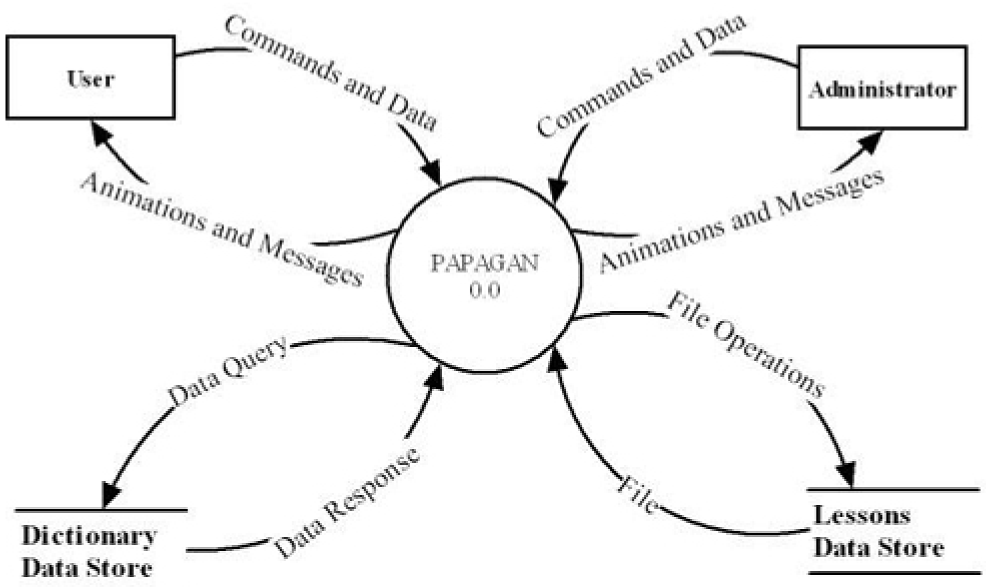

Figure 10: Level 0 Data Flow Diagram

- Level 1 DFD

In this section, as it can be seen from Figure 11, we detail the underlying system of PAPAĞAN with its building blocks, namely its modules:

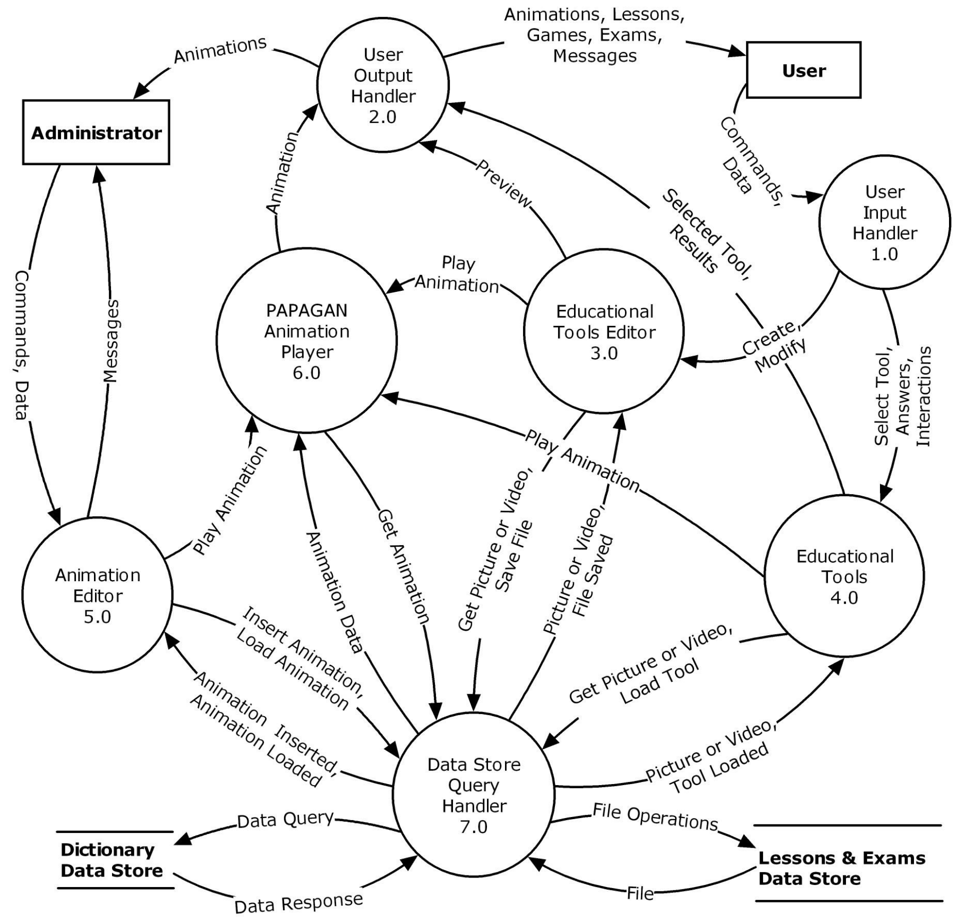

Figure 11: Level 1 Data Flow Diagram

- User Input Handler
- User Output Handler
- Educational Tools Editor

- Educational Tools
- Animation Editor
- PAP
- Data Store Query Handler

4.2 Entity Relationship Model

The simple ER diagram below (Figure 12) shows the entity-relationship model of PAPAĞAN. Attributes of the entities are depicted in separate parts in order to keep the visualization clear.

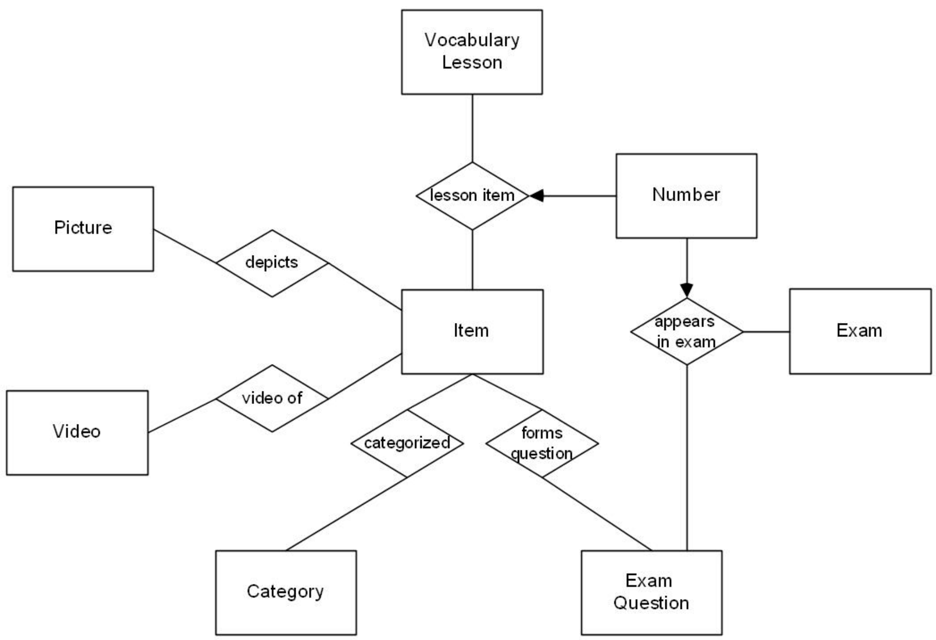

Figure 12: Entity Relation Diagram

Skimming through the entities, we observe:

- *Item*

  Item stands for anything that has a sign language equivalent. It may be a word, a phrase, a number, a complete sentence or a single letter.

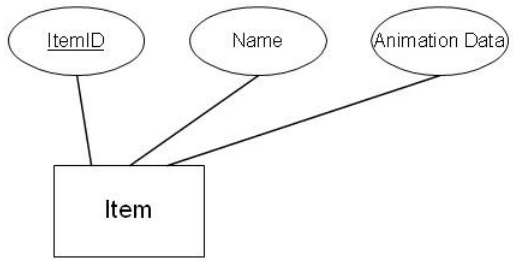

Figure 13: Item Entity

Each item has a unique id, a name that corresponds to its real meaning. It is important to mention that *name* attribute is not unique, since a word has different meanings expressed with different signs. For example the sign equivalent of the word *carry ("taşımak")* has different signs, one stands for *carrying* in *carrying a baby*, other stands for *carrying* in *carrying a bag*.

Each item has a animation data equivalent which can be thought as a text (VARCHAR in a DBMS), describing the animation in a vector form.

- *Picture*

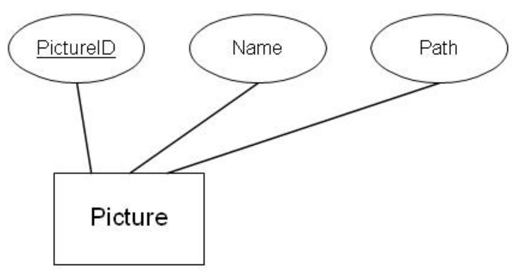

Figure 14: Picture Entity

Picture entity is the representation of pictures used to support visualization of items. Each *Picture* has a unique *PictureID*, a name and a path showing the location of the file.

- *Video*

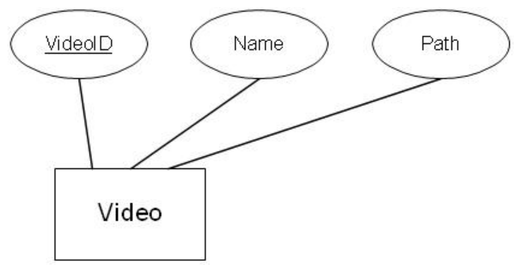

Figure 15: Video Entity

Video entity is the representation of video used to support visualization of items. Each *Video* has a unique *VideoID*, a name and a path showing the location of the file.

- *Category*

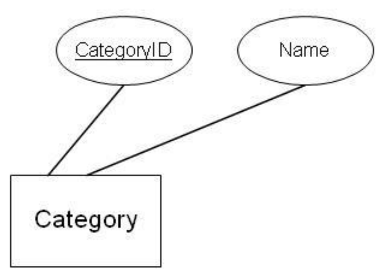

Figure 16: Category Entity

Category entity is used to group items. Each category has a unique *CategoryID*, and a name.

- *Vocabulary Lesson*

  Vocabulary Lesson is the entity, that encapsulates the relevant information about the Vocabulary Lessons . Each Vocabulary Lesson has a unique *LessonID*, a name, and a description that contains a brief text information.

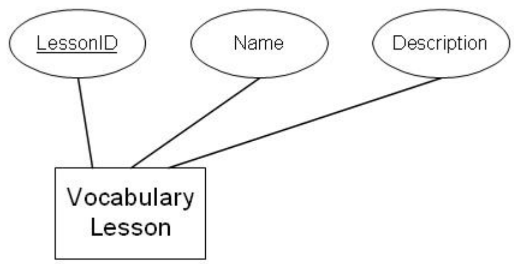

Figure 17: Vocabulary Lesson Entity

- *Number*

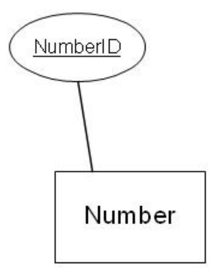

Figure 18: Number Entity

Number is nothing but an enumeration for the *Item*s which are included in a Vocabulary Lesson.

- *Exam Question*

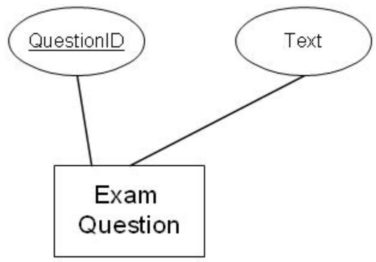

Figure 19: Exam Question Entity

Exam Question encapsulates *QuestionId* and *QuestionText*

- *Exam*

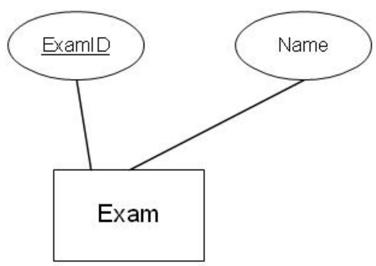

Figure 20: Exam Entity

The *Exam Name* and *Id* are kept here.

All in all, there are some remarks to remind:

- The relations are kept many-to-many, not to bind a picture or video to a single item, or an item to a single category.

- The entities and relations including Vocabulary Lesson and put in the diagram only to describe the ER model of the project. As mentioned before, they will be kept in separate files rather than a database.
- Fingerspells are also data but they are not considered separately, since every fingerspell is a sequence of animation data of individual letters.

5 Project Planning

5.1 Team Organization

Korsan Yazılım has a Democratic Decentralized team structure. There is not any permanent leader; instead, every month one of the team members gets organization responsibilities. Decisions are given by all team, each member has exchangeable roles during the project and there is always horizontal communication among members.

5.2 Process Model

The process model of PAPAĞAN Project cannot be classified into one of the predefined process models because of the constraints of the project schedule. The deadlines of analysis and design procedures are predefined in the course syllabus. Each phase can be considered separately in order to explain our process model better.

- Analysis:

  Analysis phase is strictly linear, since the deadline is fixed in the syllabus and there is no planned return to this phase. Hence everything in analysis has to be thought and planned carefully.

- Design:

  Design phase also has strict deadlines and returns are limited, but it can be asserted that this phase is more evolutionary since the phase is divided into two. After the initial design, the design can be refined by the feedbacks of academic staff and reviews of the team members.

- Implementation:

  This part is far from being linear. Starting with prototype implementation, in this phase new modules are implemented, tested and revised continuously in an evolutionary manner.

5.3 Major Constraints

- *Project Team Constraints:*

  The profile of the Korsan Yazılım team is a major constraint in the project. All members are senior students and they have no specialization neither in software engineering nor in computer graphics programming. This directly affects the analysis, design and implementation

- *Schedule Constraints:*

  Since this is a graduate groject, the deadlines are set definitely. This is a constraint in scheduling analysis, design and implementation and determining the process model.

- *Hardware Constraints:*

  In the early stages of the project, motion capture techniques were considered to model sign language animations, but the hardware facilities needed were not available, so we were urged to change our modeling strategy to pure software techniques.

- *Software Constraints:*

  Since this is the first project of the team in this area, the software tools used have to be well documented and easy to learn. Hence, open source tools will be used primarily to obtain better documentation. For this reason, we project to use an open source rendering engine such as OGRE or OpenSceneGraph rather than a commercial one like RenderWare.

- *Subject Constraints:*

  Starting the project we wanted to create a text to sign translator for TİD , but the studies on TİD were not sufficient to make such a kind of application. Therefore our project is limited to a dictionary based approach.

5.4 Timeline

It was a very difficult task for us to determine the work packages and scheduling of these in a realistic manner. What we pay attention to during this process is achieving modularity and sticking to syllabus. After some noteworthy effort, we decided on project timeline as you can see, of course not the final version, in Appendix. For clarity, work packages of PAPAĞAN can be stated as follows:

- *WP1: Project Management*

  This is our all inclusive package that strictly sticks to the syllabus.

- *WP2: Animation Editor Development*

  This is the crucial part of PAPAĞAN, so it forms most of our implementation work. Besides, educational tools such as Dictionary and Lessons highly depend on the completion of Animation Editor.

- *WP3: GUI Design & Implementation*

  We decide on and detail this package in order to have a presentable prototype.

- *WP4: Educational Tools Development*

  It's thought unrealistic to detail this package so, only very rough time estimations are given.

- *WP5: Finalizing Package*

  It's thought unrealistic to detail this package so, only very rough time estimations are given.

6 Appendix

We could only be able to represent our Gantt Chart in two pages. First half is on the 37^{th} and the second part is on the 38^{th} pages.

References

[1] http://turkisaretdili.ku.edu.tr/en/tid.aspx

[2] http://www.vsign.nl

[3] http://asl.cs.depaul.edu

[4] http://www.sign-lang.uni-hamburg.de/eSIGN

[5] http://www.vcom3d.com/ReadingPower.htm

[6] http://www.signgenius.com

[7] http://www.readysetsign.com

[8] http://www.visicast.co.uk/news/Tessa.html

[9] http://commtechlab.msu.edu

[10] http://www.freeonlinesurvey.com

[11] http://freeonlinesurveys.com/rendersurvey.asp?id=124883

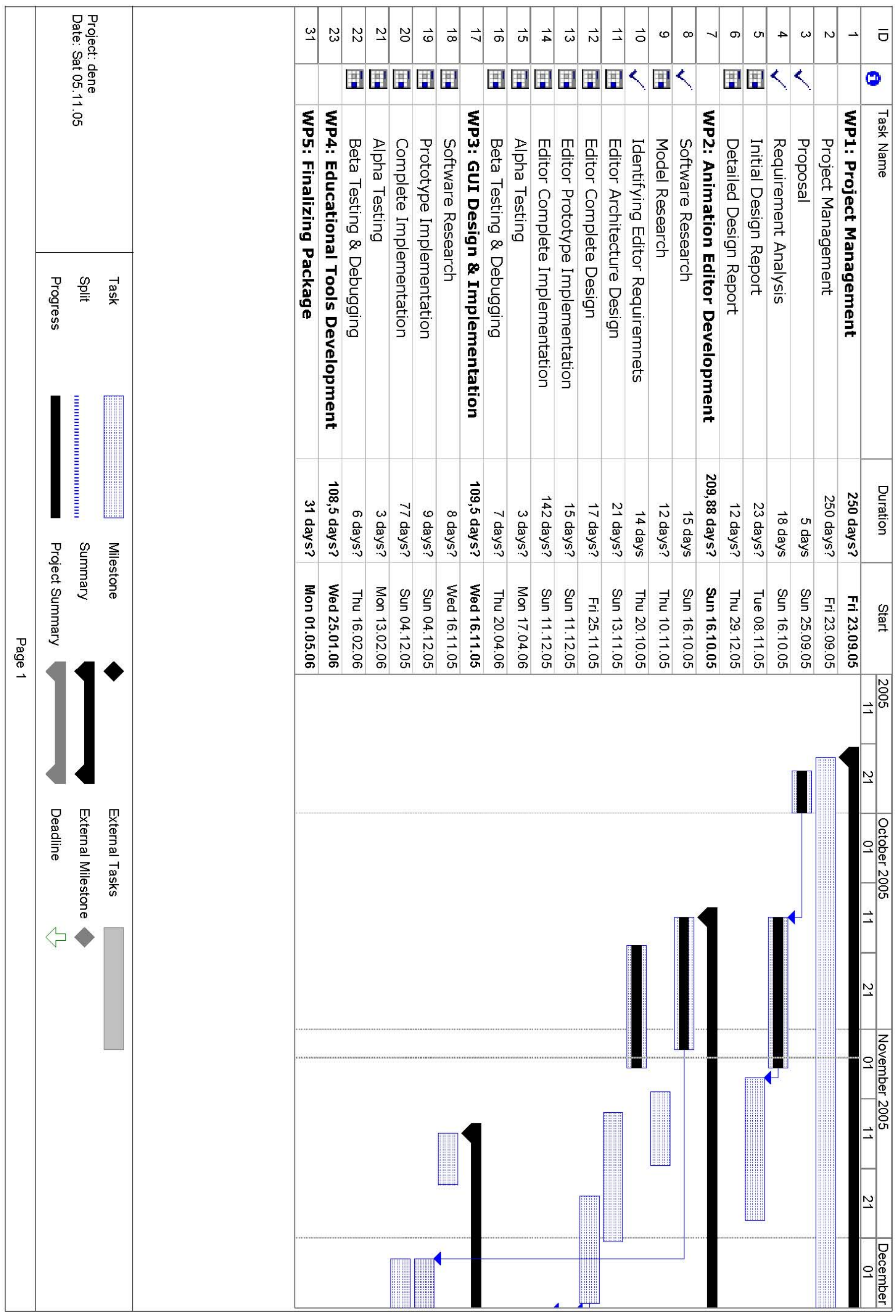

Figure 21: First Half of Gantt Chart

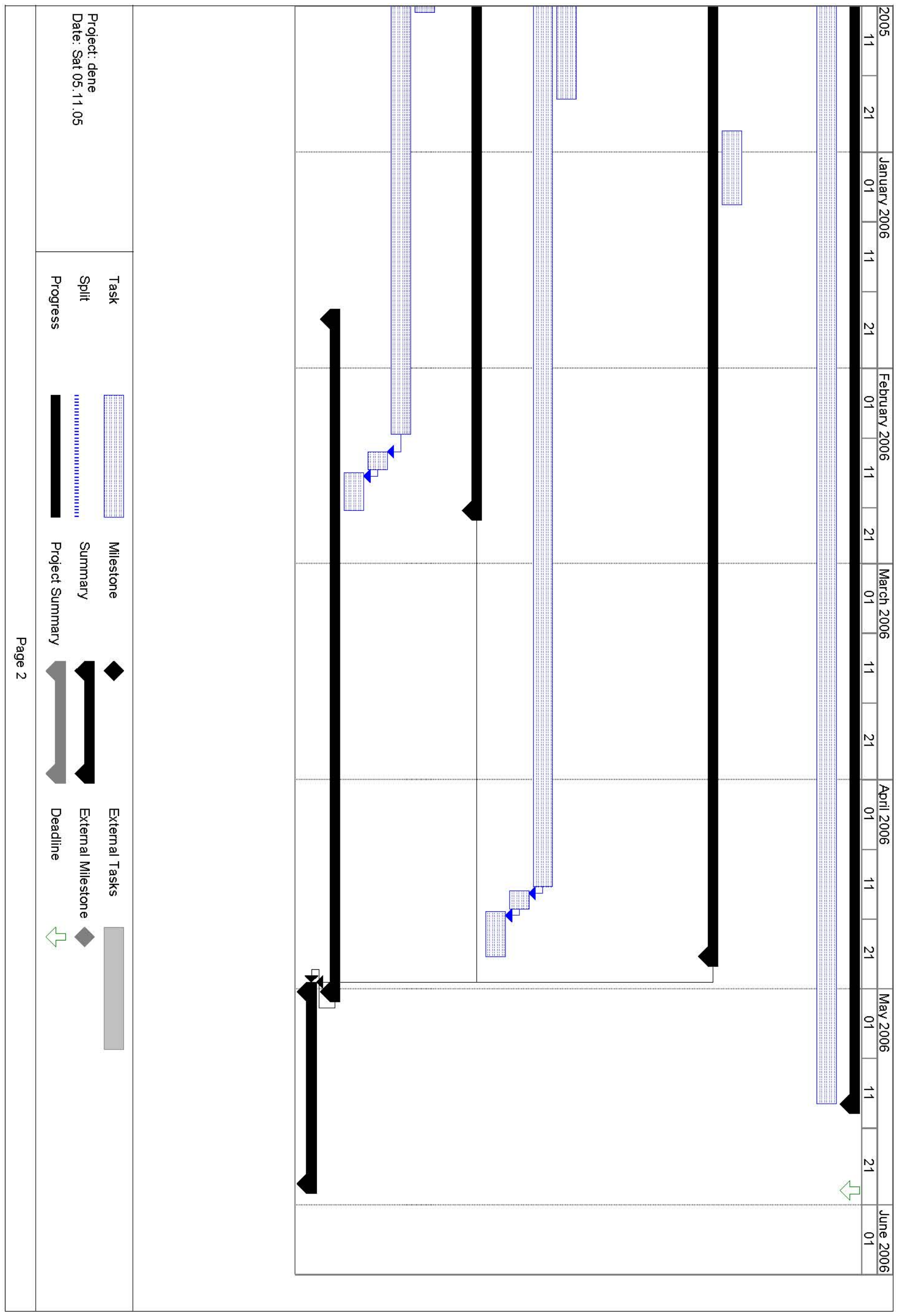

Figure 22: Second Half of Gantt Chart