MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

SENIOR PROJECT

FALL 2007

FINAL DESIGN REPORT

ONUR AK 1394576 **ŞERİF ÇETİNER 1394832**

SADETTİN ŞEN 1395540 **MASHAR TEKİN 1395565**

TURKUAZ

UMUT EROĞUL

Index of Contents

1. Introduction

Nowadays, in lots of area the amount of the information stored rapidly grows and this information is used more and more. Patient records in hospitals, student records hold by government, records of law suits and record of customer complaints are some example about this case. Every day new records are added, the old records are used. A doctor may need an old report about a disease or a patient, old court decisions are reference for new decisions and a bank used customer complaints to furnish better service. However, in these cases having access to specific reports can not be done manually. There are thousands of patient records, law suits and customer complaints. This document is about a solution for similar problems in a medical area namely Radiology.

1.1 Problem Definition

In the medical area Radiology, every day lots of reports are written about patients using their radiographs and they are stored. Later, this stored information can be used in researches, statistical analysis, medical practice and teaching. Doctors may need to follow patients' past and future reports to see responses for treatment, or they want to see all patients' results for a specific treatment. Obviously, this will be a valuable resource on doctors' researches about treatments. However, a large percentage of reports is unstructured in a form of free-text, therefore difficult to search, sort, analyze and summarize. Since, only structured data are amenable to advanced causal, spatial, temporal, and evolutionary database modeling techniques that are now being developed in the fields of medical informatics and computer science. The implications for teaching files and data collection for retrospective research studies are obvious. However, structuring reports and using them is a difficult task. There are several important points that we must find a solution. Firstly, we must extract technique used for treatment and inspection. Although this part is separated with a subtitle in reports, there may be more than one technique under this title. While defining techniques there may be conjunctions, abbreviations that we must eliminate and find their expansion. Secondly, we must extract disease with its name, place that it arises, diagnosis for this disease. This part generally told under a subtitle, contains several sentences. Here we need eliminate inessential words and determine the words tasks in sentences.

Lastly, we must extracting results of treatment whether it is positive or negative. Which is similar to second part and extra need is determining meaning of predicate. For finding valuable information we need a Turkish language processing tool. The only open source language processing tool for this is Zemberek which has restricted features. It can not analyze medical terms, and it only deals with the words' root, type, prefix and suffixes. It does not analyze words' tasks in sentence. Therefore, we must pre-process all reports and use additional Natural Language Processing techniques on the outputs of Zemberek. After accomplishing these tasks our solution will include that the valuable information in the reports is extracted, such as patient name, the organ that is examined and the diagnosis. This extracted information, too smaller with respect to whole report, is stored and the search is done using these words. Also in this solution, search can be done using reports.Valuable information in a given report is extracted and the search is done using these extracted words. Therefore, you can find all related reports.

1.2 Project Scope

Our Project will include three functionalities. First functionality is that user can search reports with respect to doctor name, patient name or a disease. Second one is searching related reports using given reports. This part includes the information retrieval from input report. Last one is adding a report, newly written, to database.

2. Design Constraints

Here we told our constraints under 4 subtitles: Resource Constraint, Time Constraints, Integrity Constraints and Performance Constraints.

2.1 Resource Constraints

Because of platform independence and having a large library that simplifies our work we use Java language as a development tool. We can easily find libraries and documents that we need. Also accordance is another point that we bear in mind. Since Zemberek, a NLP tool for Turkish, is coded in Java using Java is more suitable for us.

2.2 Time Constraints

We have to finish the project before the end of second semester. Machine learning and information extraction parts will be done in second semester and other parts need improvements. During second semester we will need to analyze a huge database of reports, determine important structures contain valuable information. Also we will connect to database of Hacettepe University from our program for reaching reports of Radiology Department of university. Doing these works in second semester will be difficult; we have to use our time efficiently.

2.3 Integrity Constraints

In our Project, we have divided the problem into sub problems and each member is responsible for some sub problems. And some sub problems are dependent on other sub problems to be solved. Such as in Morphological Analysis one group member must accomplish to analyzing medical terms, if not next problem, part of speech tagging, and then information extraction can not be achieved. Therefore, if the previous step that should be done before a sub problem is not finished and integrated to our program that would be a great loss of time.

2.4 Performance Constraints

Currently there are thousands of reports that must be analyzed and stored. After this, our program will be carrying out a search on information that is extracted from these reports. Since our database will grow later, this search may become inefficient. Therefore, we must bear in mind the speed of our program and we escape from inessential information while storing and operations during search.

Another performance is about the accuracy of words that we extract and their classifications. Actually this is more important than speed. Since these reports are about human life we must select information carefully. We must make correct classification such as name of disease, place of disease and its result. At the end of second term we aim 80% of accuracy on selection and classification of correct words.

3. What has been done?

At the end of the first semester, our prototype is finished with some basic functionalities of our projects. Below modules that we done all or some parts are told:

1. **GUI:** GUI of our project is finished for working parts of our project. Others will be added later. In next chapters design interface is told in more detail.

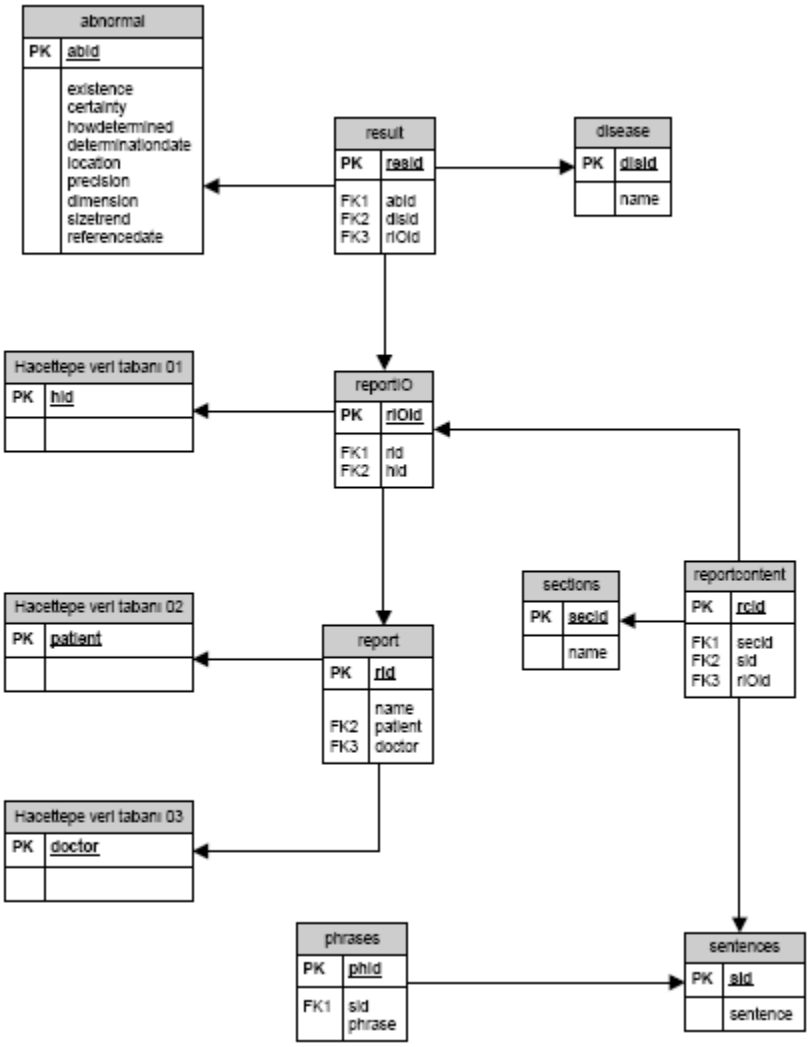
2. **SEARCH:** Since in our reports there is not patient name and determining disease involves all functionalities of Text Miner Module these two sub systems were not done. For search module we have done search by doctor name and search by other word parts.

3. **TEXT MINER:** For this module we implement some of subsystems. First of this subsystem is Tokenizer that we can separate reports to paragraph, sentences and words. Second subsystem is Morphological Analyzer. Except medical terms we can analyze words. Last subsystem that we have implemented is Part of Speech Tagging part. Here we can find prepositional phrases which contain objects and indirect objects.

4. **DATABASE:** We can use database for storing extracted information such as doctor names and all reports. In database there is also a table for storing results of Zemberek. We can do search on database and reach the report from there.

4. Database Design

4.1 ER Diagram of Database



4.2 Table Descriptions

Hacettepe veri tabanı 01	
PK	**hid**

This table holds radiology reports in database of Hacettepe University Hospital.

Hacettepe veri tabanı 02	
PK	**patient**

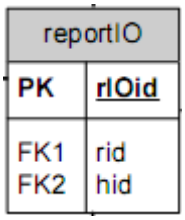
This table holds patients' datum in database of Hacettepe University Hospital.

Hacettepe veri tabanı 03	
PK	**doctor**

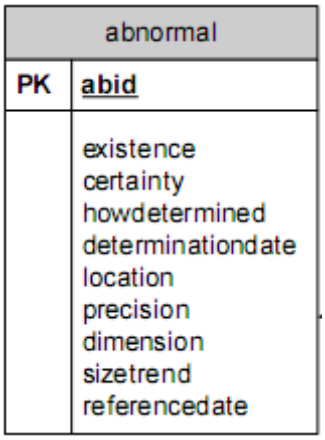
This table holds doctors' datum in database of Hacettepe University Hospital.

report	
PK	**rid**
FK2 FK3	name patient doctor

This table holds the report name and its related patient and doctor foreign keys to "Hacettepe veri tabanı 02" and "Hacettepe veri tabanı 03".

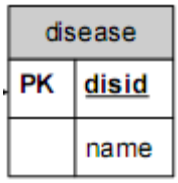


This table holds the report id, decides whether it is a new report added (foreign key to report) or it is an old report that exists in “Hacettepe veri tabanı 01” (foreign key to “Hacettepe veri tabanı 01”).

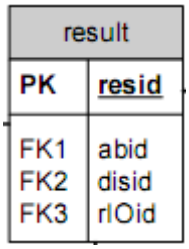


In this table, the abnormalities found in the report are stored.

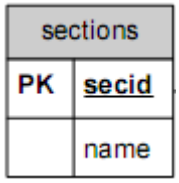
The properties of abnormal are existence, certainty, how it is determined, determination date, location, precision, dimension, the trend of size and reference date.



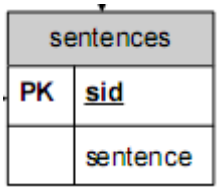
The diseases found are stored in this table.



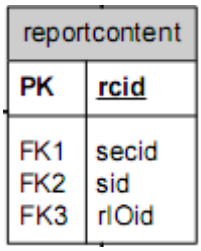
This table makes a connection between found diseases or abnormals with the report. All fields are foreign key (abid to abnormal,disid to disease, rIOid to reportIO) except resid (Primary key).



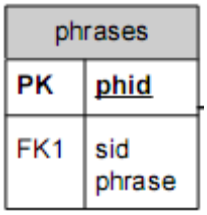
This table holds the names of the sections in the radiology reports.



This table holds the sentences that are found in information extraction module in the radiology reports. This table will be extracted later from the project as we develop in finding phrases.



This table holds a foreign key to sentences, a foreign key to sections, and a foreign key to reportIO.



This table holds the phrases (noun phrases, adjective phrases,…) that are found from the reports.

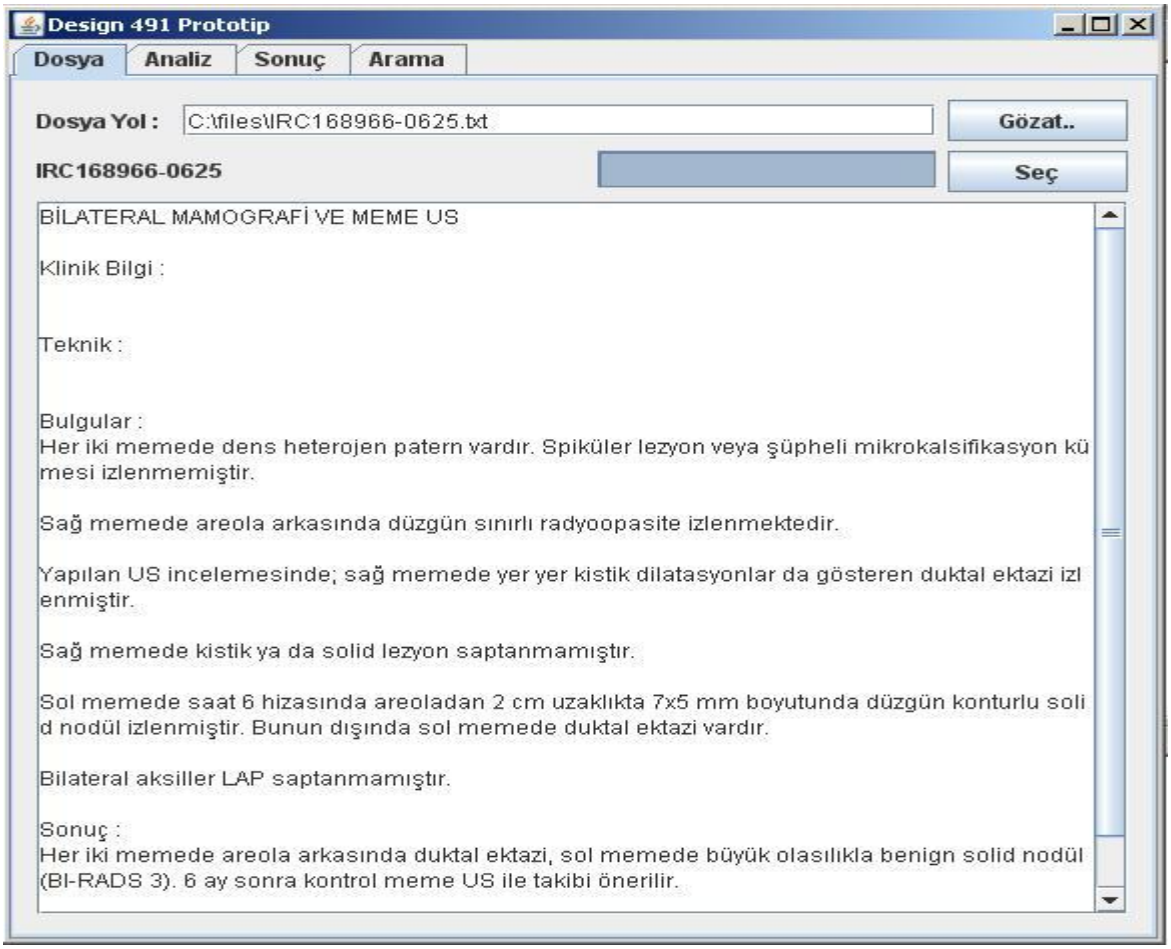
5. Graphical User Interface Design

Our Graphical User Interface consists of 4 main tabs, and the last tab also contains 2 tabs which are:

1. Dosya
2. Analiz
3. Sonuç
4. Arama
    a. Genel
    b. Eski Dosyalar

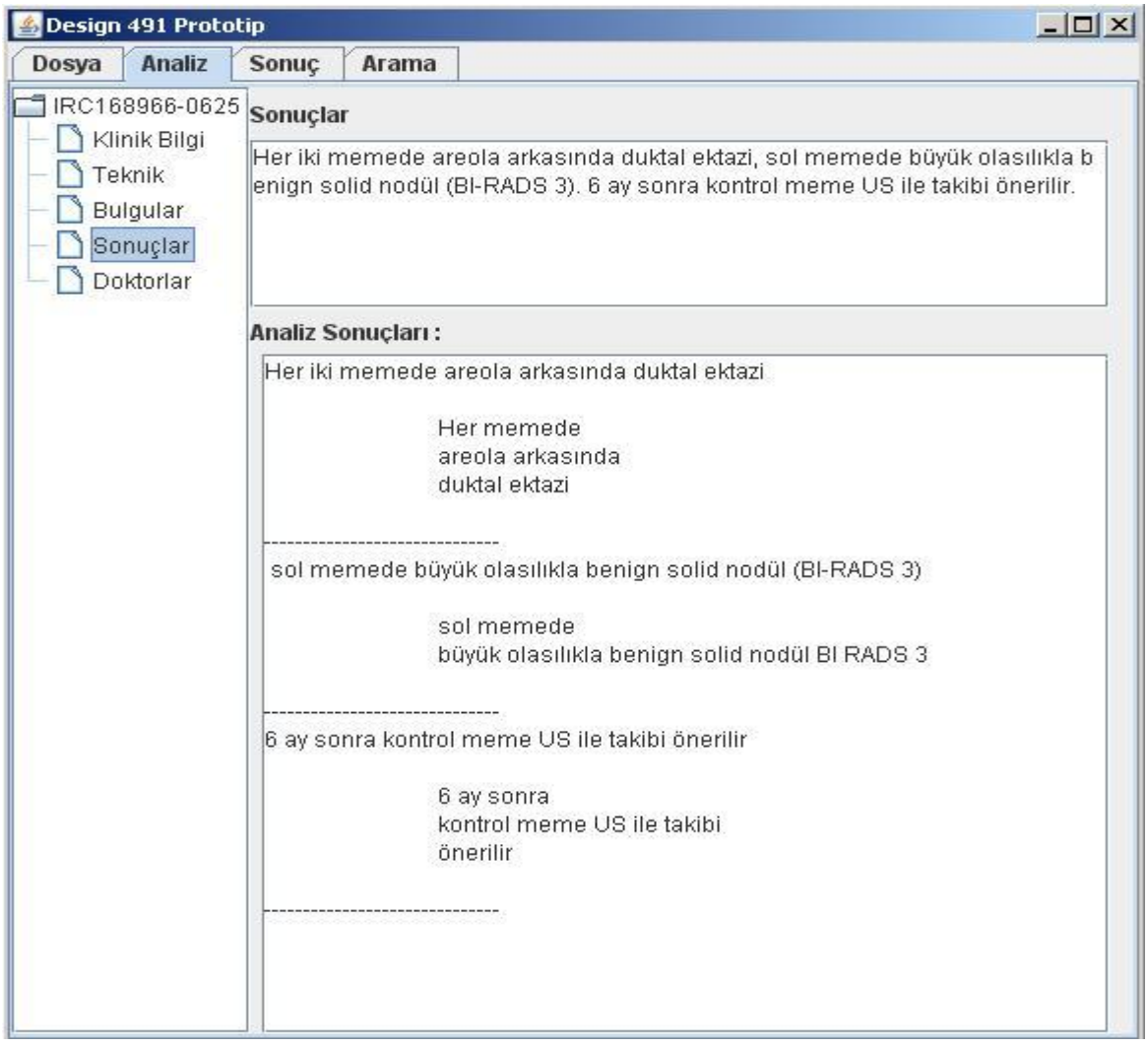
and can be seen in the following screenshots.

1) Dosya



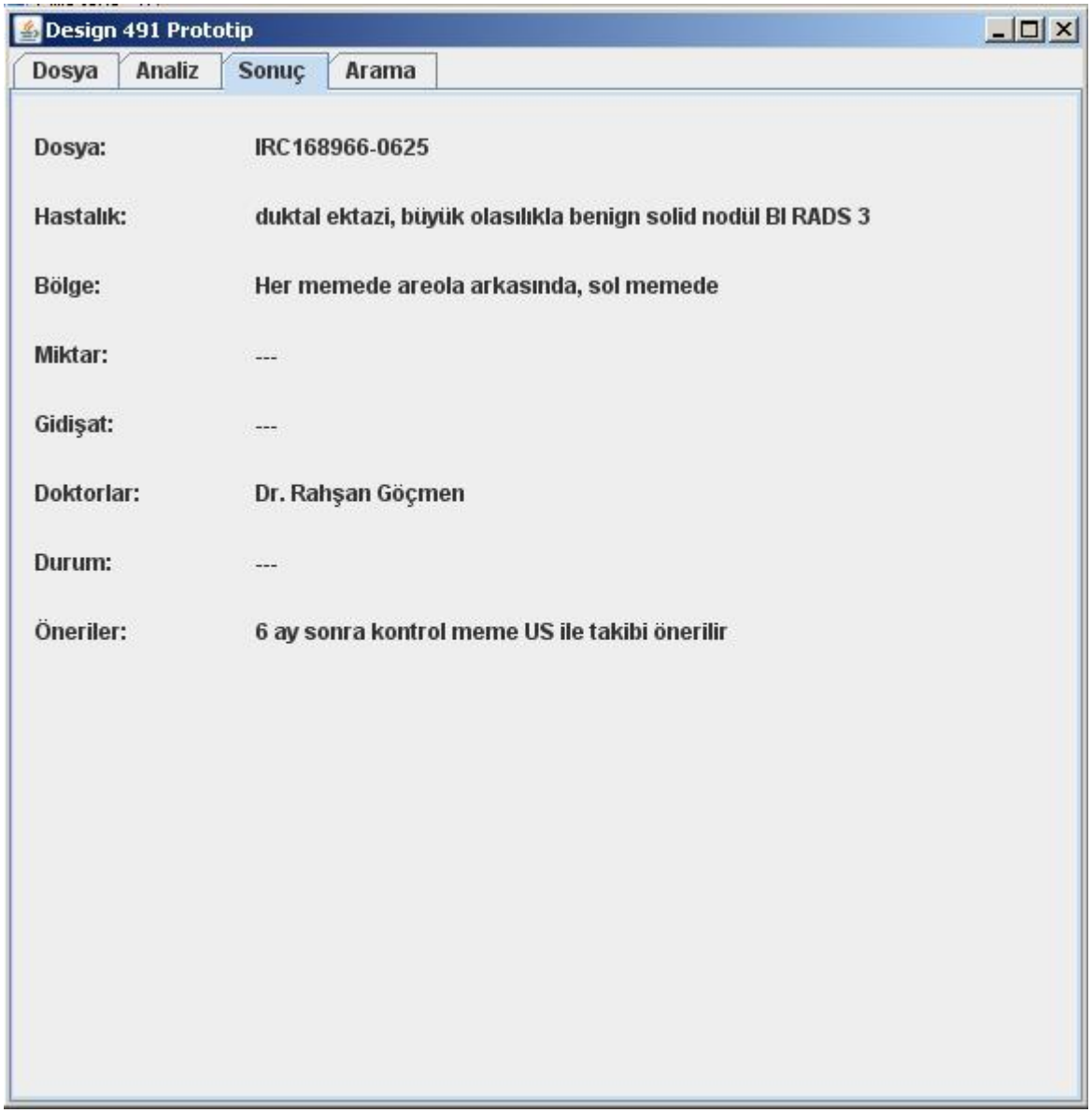
As you can see from the screenshot, there is a textField in the upper part of the window from which you can enter a file name or you can also choose a file by pressing the “Gözat” button. It opens a new window from which you can choose a file or folder. There is a filefilter while choosing the file so that we are ensured that we do not get a file with an unknown file format. After you choose the file the textField in the upper part of the window is automatically filled by the path of the chosen file or folder. After choosing the file if you press the button “Seç” it then looks up the file which is provided in the textField and then makes some evaluation about the report and the textArea in the bottom of the window is filled with the whole chosen report file.

2) Analiz



In this tab, the left part of the window is a tree which has the sections that can be provided in a report file. When you choose a node of this tree as an example “Sonuçlar”, which can be seen from the Screenshot above, two textAreas namely “Sonuçlar” and “Analiz Sonuçları” are filled. The Sonuçlar part which is located in the upper part of the window shows us the whole Sonuçlar section of the report. Then in the bottom part, the Analiz Sonuçları shows us the separated form of the section which is divided into sentences and for each sentence the separate valuable phrases. This tab is actually for to see our division of the sentences into phrases.

3) Sonuç

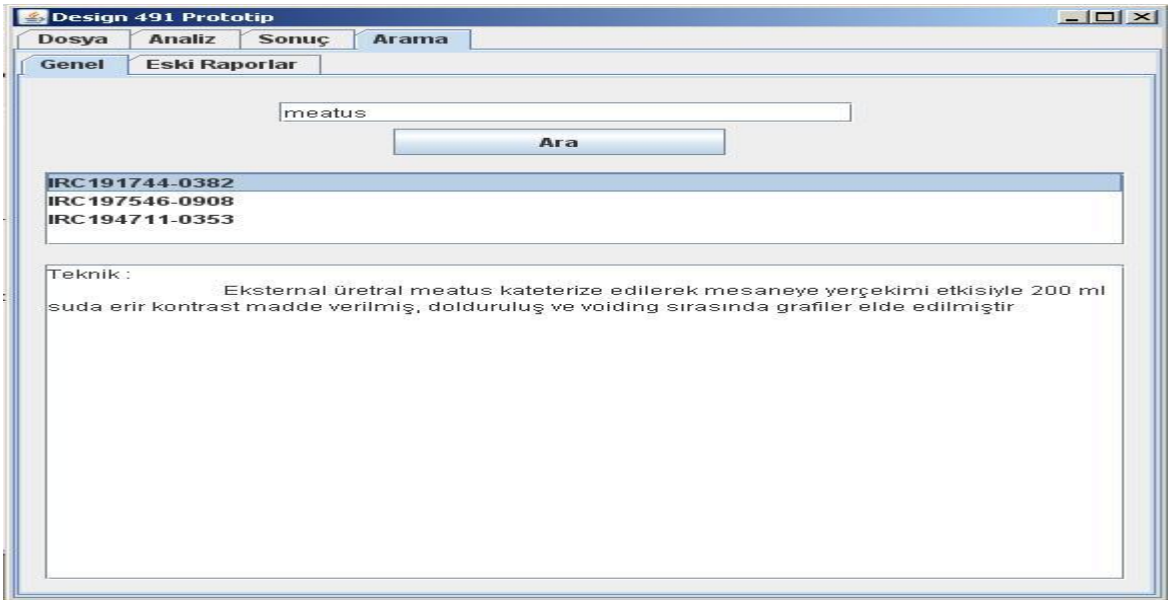


This tab is the main part of this Project in which one can see the disease , which is shown by Hastalık , the region of the disease , which is shown by Bölge , the size of the disease if it is applicable and provided , which is shown by Miktar , the situation of the disease with respect to before reports of the patient if there are any , which is shown by Gidişat , the condition of the disease , which is shown by Durum , the suggestions , which is shown by Öneriler and usually suggests a patient to come back for an examination after a time period and also there are doctor names , which is shown by Doktorlar , and file name , which is shown by Dosya . There should be a patient name field but since there are no data provided in the reports for privacy there is none yet. There will be a further information field which will include further information extracted from not only Sonuçlar section of the report but also from other sections of the report.

4) Arama

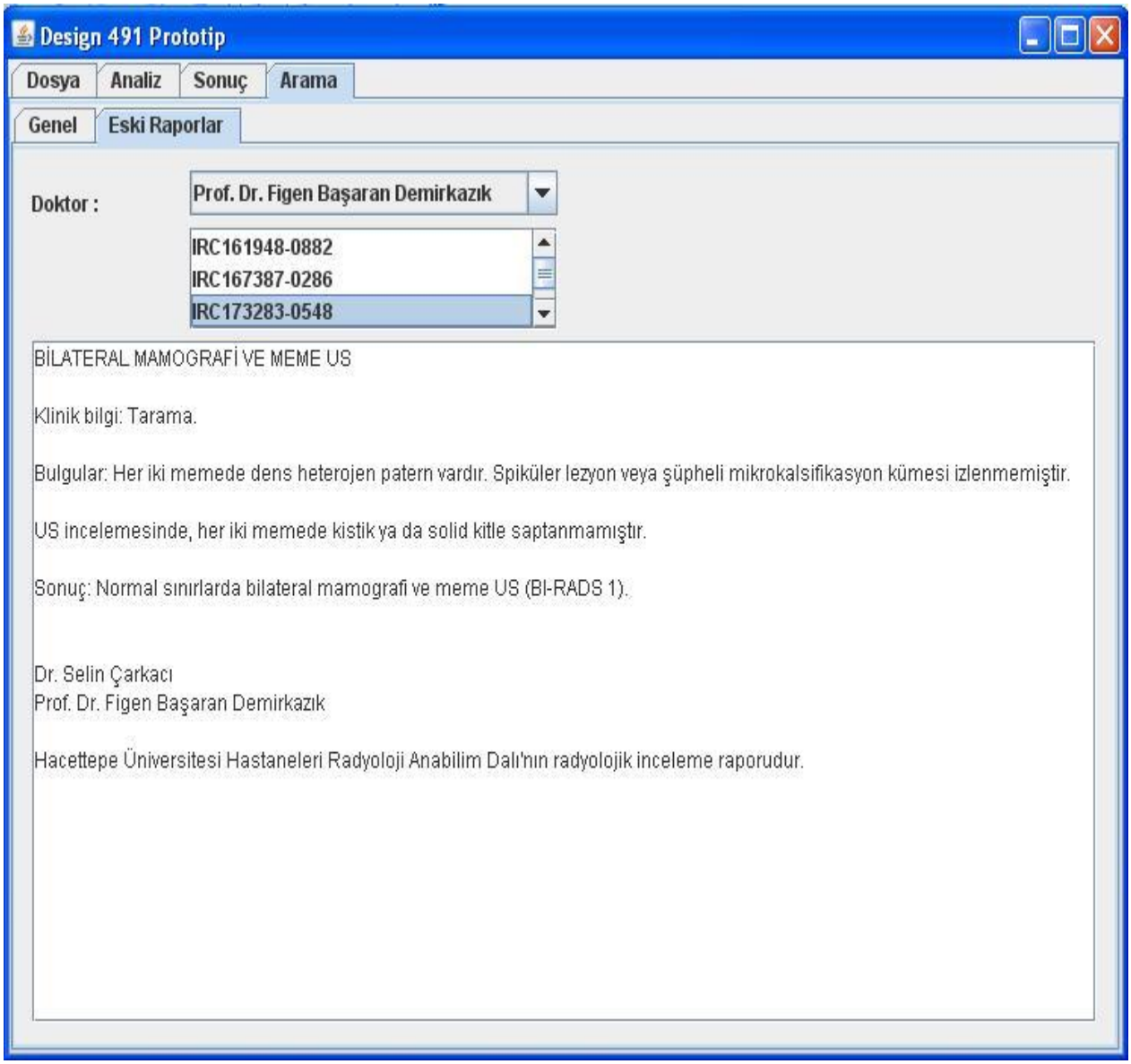
This tab is designed for search operations. We have implemented general search and finding reports of a doctor which can be chosen from a comboBox. There will be also search by a patient name, and finding related reports.

a. Genel



In this tab, you can enter a search value to search from our database and when you press “Ara” button the word(s) are searched throughout our database and the if there are any matched report then they are printed in the list below the “Ara” button. And after you choose a file name of a report, the section in which the searched word included is shown in the textArea in the bottom of the window.

b. Eski Dosyalar



In this tab, you can choose a doctor from the comboBox and after you select a doctor, his/her reports are shown in the list below the comboBox and if you select a report from here, the contents of the reports are shown in the below textArea.

6. Future Work

Currently we find only phrases of reports not its subjects, objects or predicates since we can not analyze medical terms. Next term we will achieve finding the subjects, objects, indirect objects and predicates of sentences. Each of this, except predicate, is part of phrases. Therefore, phrases that are found will be base for finding these. For determining these we will use affixes in Turkish. There are two main affix types in Turkish. These are "Yapım Ekler (YE)", "Çekim Ekleri(CE)".YE are important in morphologic analysis, but they are not important in making relations among words in a sentence. Therefore, YE are not as significant as CE in natural language applications and we will focus on CE. Analyzing of medical terms will be done in second term. Currently we are planning to use a medical term dictionary contains about 5000 term. If SNOMED is supplied we will use it for medical terms.
After finding word groups like subject, predicate and objects in a sentence we will make rule based extraction. Predetermined structures are used for determining several kinds of information like disease, place and treatment. For example if there is an indirect object, disease name will follow it. Therefore, finding indirect objects give the place of the disease name in sentence.

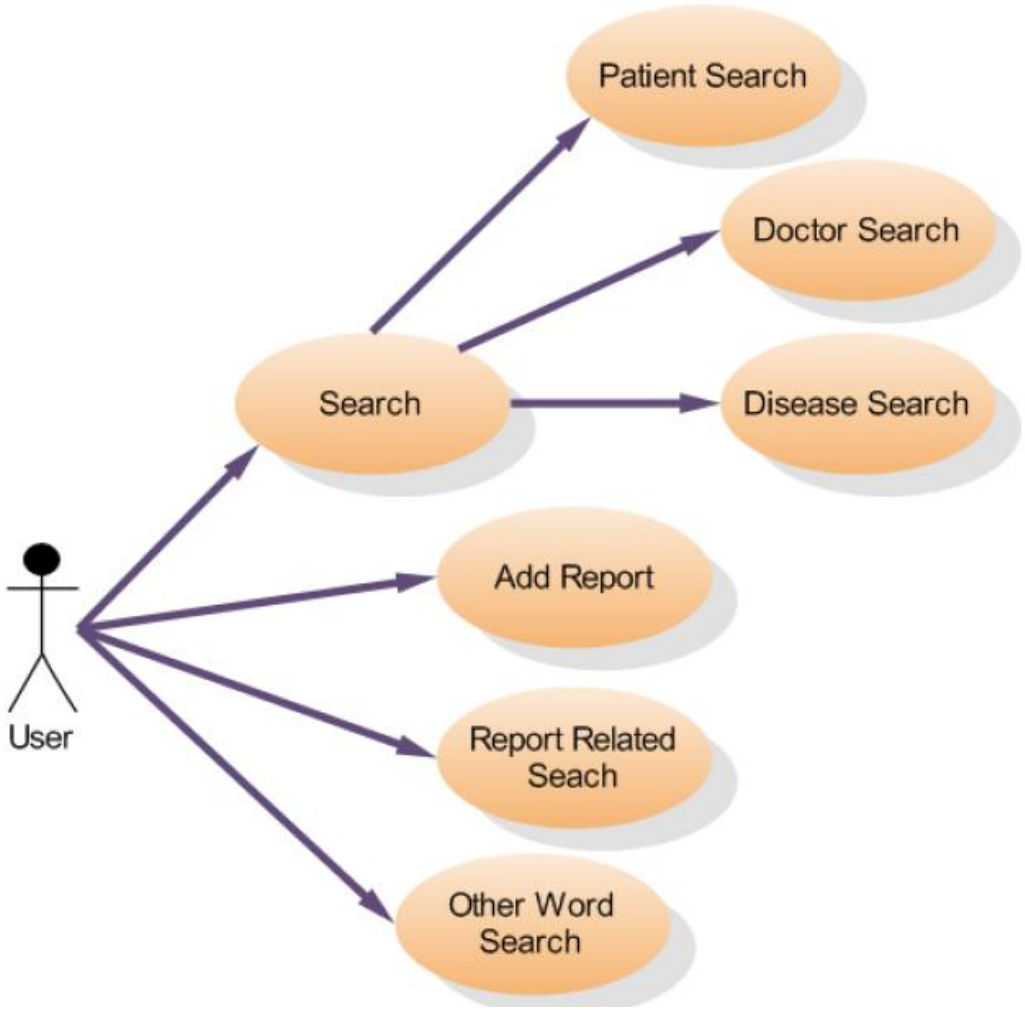
Next term we will develop our search module that it will contain search by patient name (gives all reports of patient), search by doctor name (gives all reports of doctor), disease name (gives all reports related with this disease) and the search functionality for other words.

Machine Learning will be done next term. We will determine several kinds of feature set and we will manually extract some information form reports for training. In this part we will use both tagging results and words.

Last part spell checking will be done in second term. We will use it for words that Zemberek that can not recognize. Although Zemberek has functionality that suggests words instead of unrecognized, we will define new algorithms for medical terms, because of their occurrence in Latine, English and Turkish languages.

7. Architectural Design

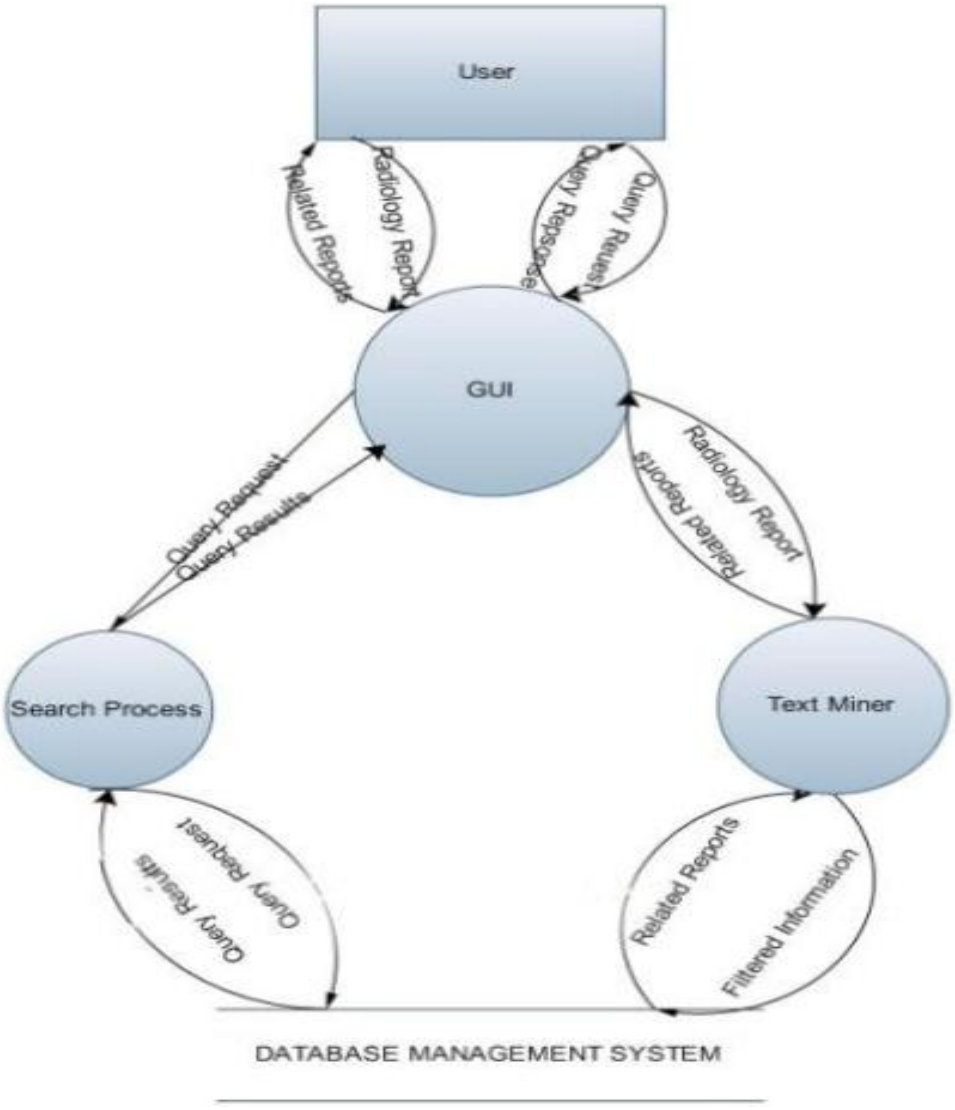
7.1.Use Case Diagram



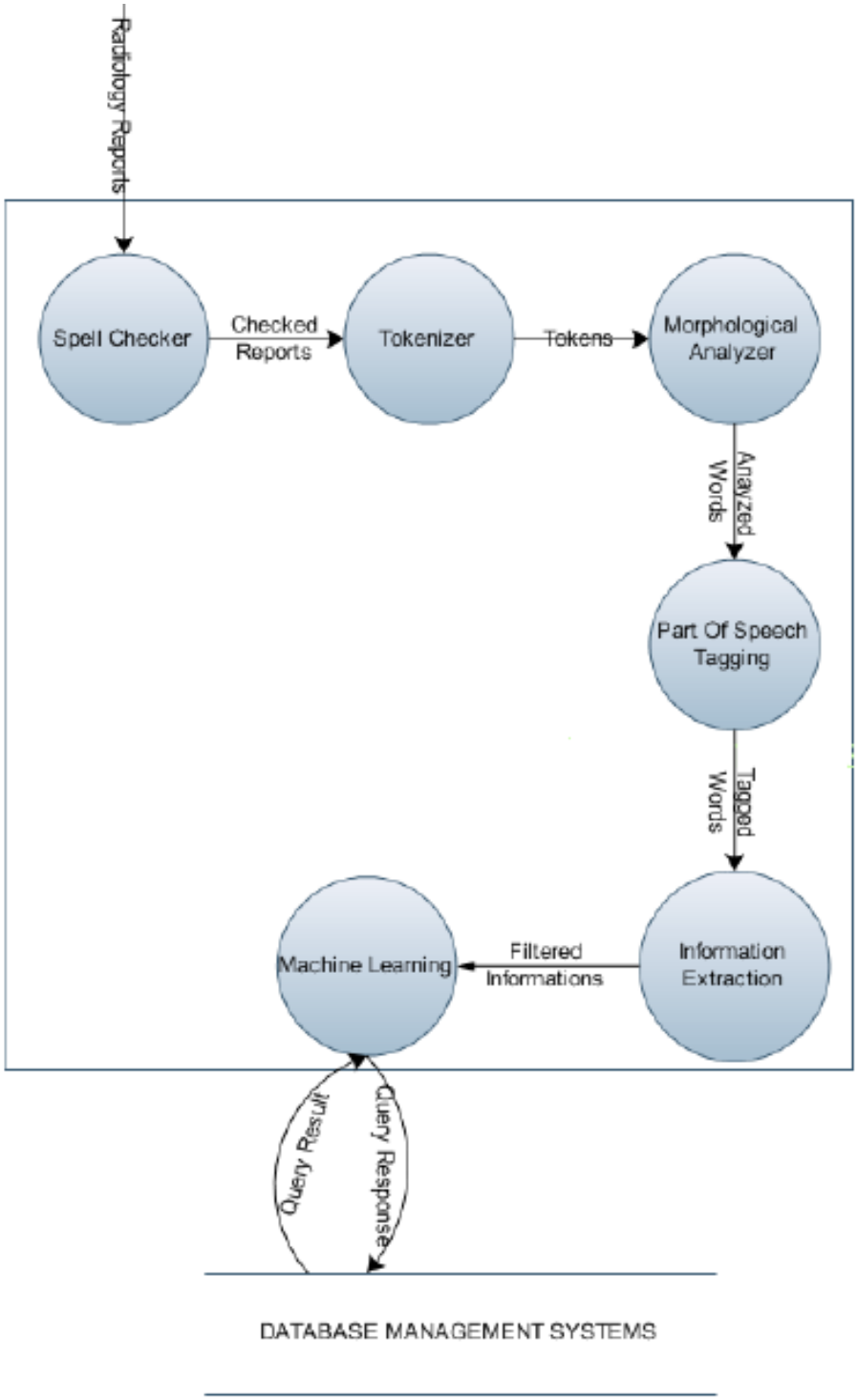
7.2Data Flow Diagram

LEVEL 0

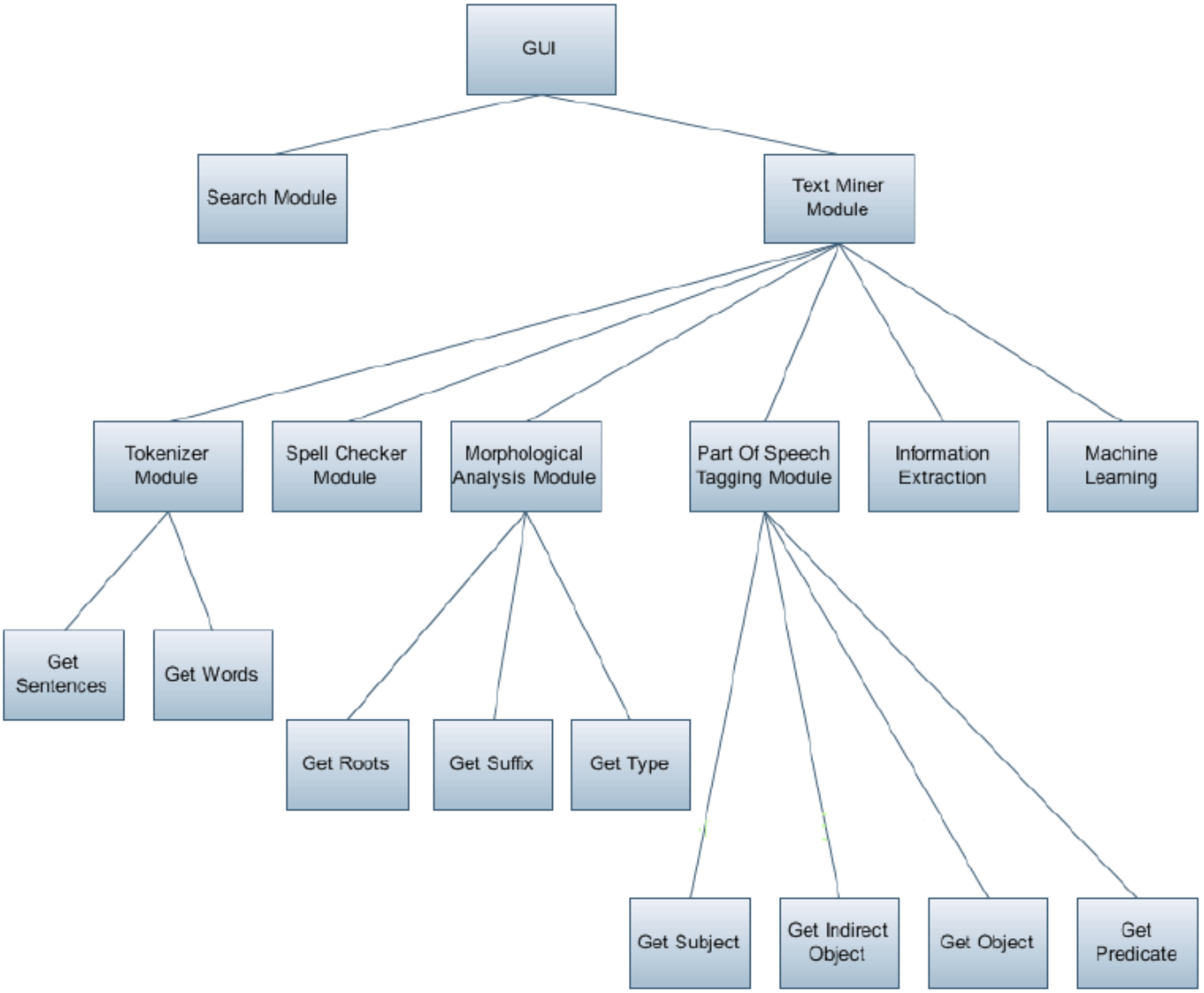
LEVEL 1



LEVEL 2 TEXT MINER



7.3.STRUCTURE CHART



7.4.MODULES

1. Search Module

This module is used for search with respect to doctor name, patient name or disease name. While disease names are extracted after text mining, others are not need to any text mining operation. They are in pre-determined structure in reports. In this module, simple queries will be used to retrieve reports containing given words.

2. Text Miner Module

The main module of our project is Text Miner Module. All information extraction, database update and report related search tasks are done in this module. The sub modules of this module accomplish the information retrieval from unstructured free text.

2.1 Tokenizer Module

In this module we find sentences in the free text and the words of this sentence. In Zemberek any given text is taken as a cluster of words, not any sentence structure is used, and only the words features are analyzed. However in Part of Speech Tagging we need to know words belong to same sentence. So we use sentence and words structure from a given text.

2.2 Spell Checker

In this module tokens are checked and corrected if there is an error. Medical terms are checked separately since they can be in Latin, English or Turkish.

2.3 Morphological Analyzer

This module is the part that we use pre-processing and Zemberek. We use pre-processing since Zemberek can not recognize and analyze the structure of medical terms commonly not Turkish. To solving this problem we use a medical terms dictionary and we will add this dictionary to database of Zemberek so that these terms can be recognize and analyzed. Using Zemberek we find roots and suffixes of words. Also we can determine the types of words.

2.4 Part of Speech Tagging

Analyzed words are input of this module and Machine Learning techniques are used for tagging. After this module the tasks of the words in a sentence are determined and valuable information is extracted using these tasks. For accomplishing this part we determine 4 important parts of speech. We use methods in master thesis of M. Oğuzhan Külekçi for determining these parts.

2.4.1 getSubject

As an important part of sentences we determine subjects. Generally reports contains passive sentences however there may be active sentences that the subject is the place of disease.

2.4.2 get Predicate

Mostly predicates give the results of disease and development of disease. We determine predicates according to their types and positions in sentence.

2.4.3 get Object

Generally diseases and treatments are objects in sentences. Mostly suffixes are used for determining objects.

2.4.4 get Indirect Object

These are the places of diseases arise. We use suffixes for determining indirect objects.

2.5 Information Extraction

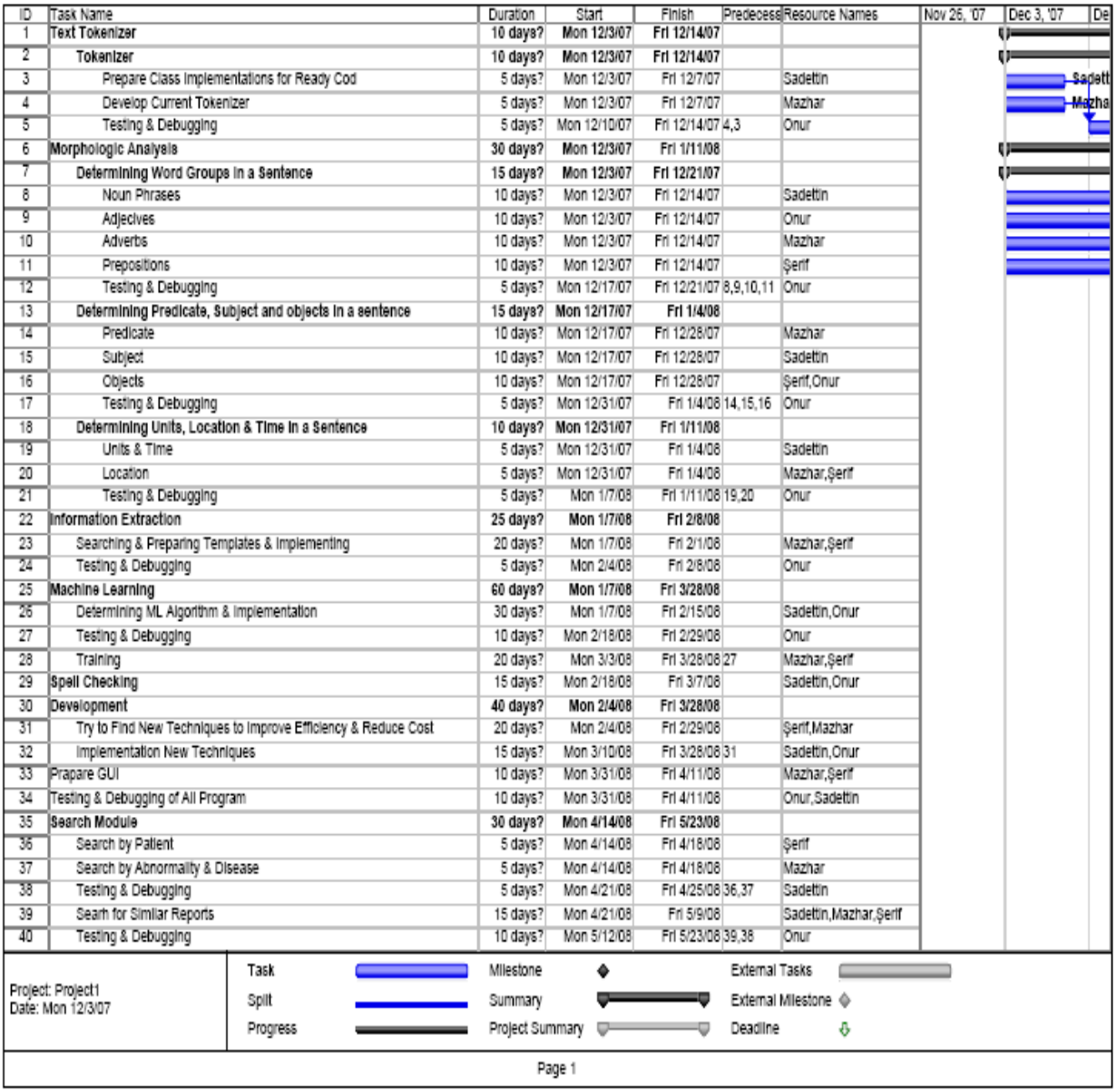
In this module, some of valuable information is determined. We use pre-determined structures to extract valuable information. These are inserted to Database if a report is added or given to Machine Learning module if any search is done.

2.6 Machine Learning

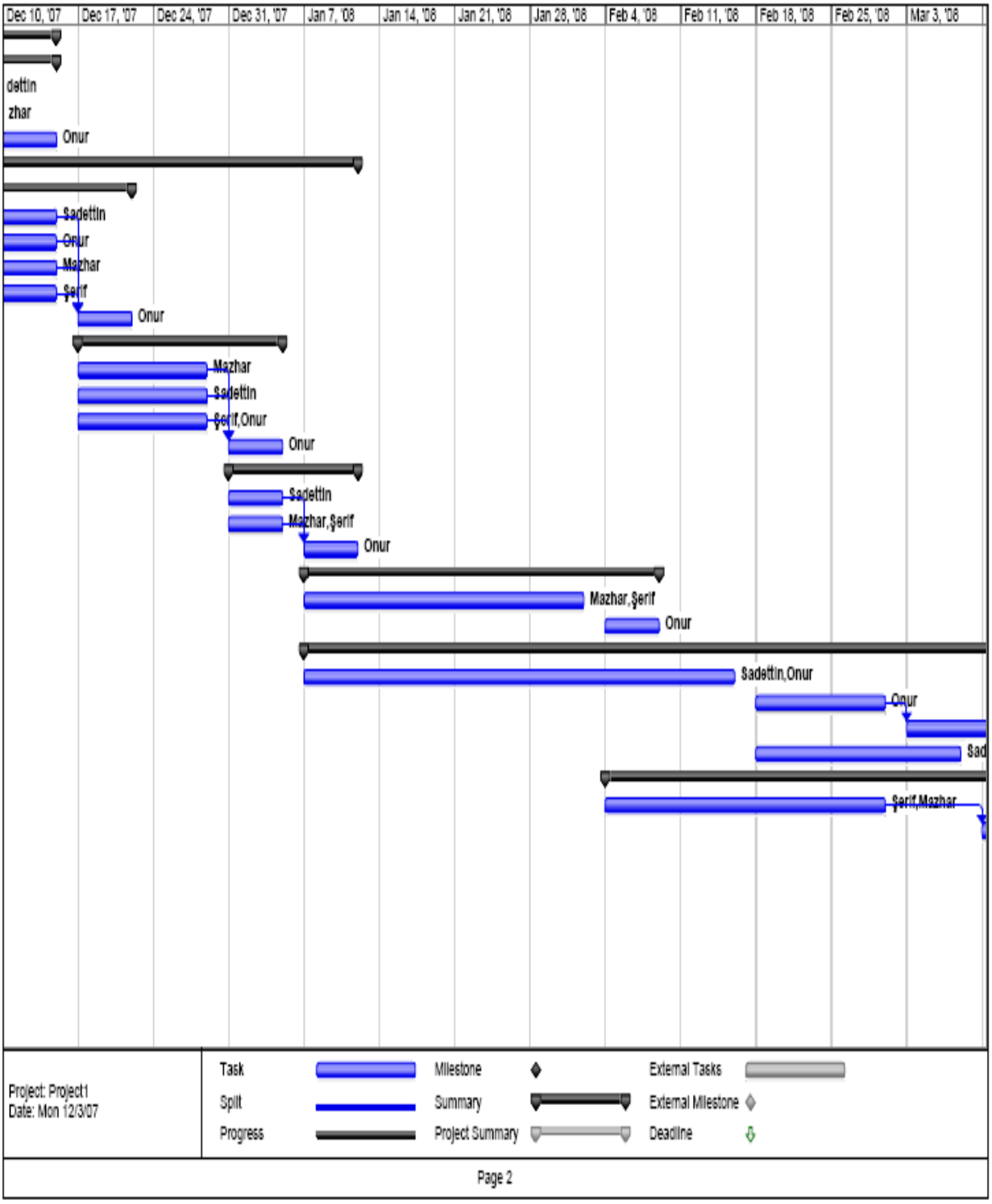
In this part we use machine learning to extract more information. A set of reports with extracted information will be used for training. We will use both part of speeches and words for training. Later each report that is passed from information extraction module will also passed from this

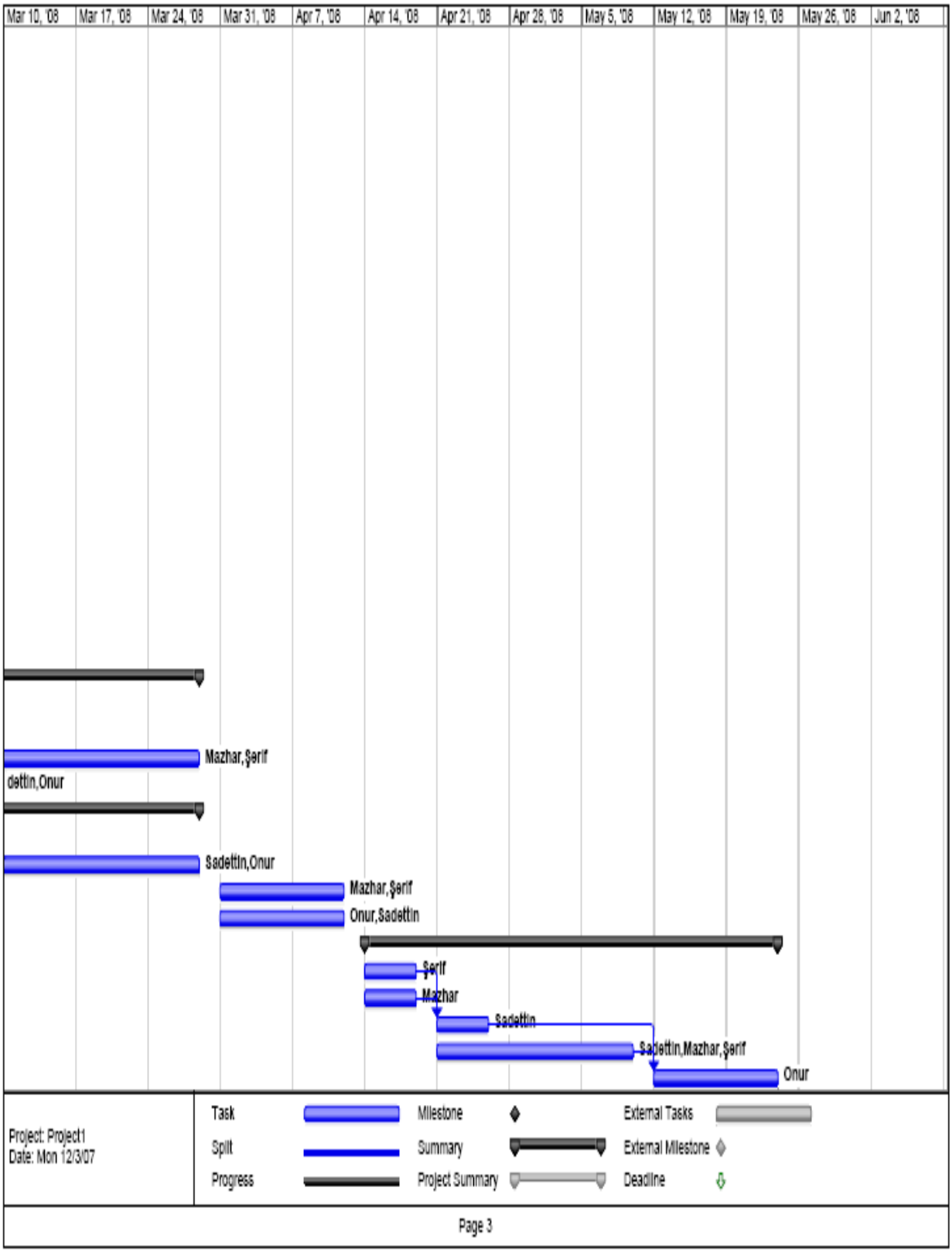
machine learning module and more information will be extracted. This is the last module of Text Miner, after this part all information are added to database.

8.Gantt Chart



ID	Task Name	Duration	Start	Finish	Predecess	Resource Names
1	**Text Tokenizer**	**10 days?**	**Mon 12/3/07**	**Fri 12/14/07**		
2	**Tokenizer**	**10 days?**	**Mon 12/3/07**	**Fri 12/14/07**		
3	Prepare Class Implementations for Ready Cod	5 days?	Mon 12/3/07	Fri 12/7/07		Sadettin
4	Develop Current Tokenizer	5 days?	Mon 12/3/07	Fri 12/7/07		Mazhar
5	Testing & Debugging	5 days?	Mon 12/10/07	Fri 12/14/07	4,3	Onur
6	**Morphologic Analysis**	**30 days?**	**Mon 12/3/07**	**Fri 1/11/08**		
7	**Determining Word Groups in a Sentence**	**15 days?**	**Mon 12/3/07**	**Fri 12/21/07**		
8	Noun Phrases	10 days?	Mon 12/3/07	Fri 12/14/07		Sadettin
9	Adjectives	10 days?	Mon 12/3/07	Fri 12/14/07		Onur
10	Adverbs	10 days?	Mon 12/3/07	Fri 12/14/07		Mazhar
11	Prepositions	10 days?	Mon 12/3/07	Fri 12/14/07		Şerif
12	Testing & Debugging	5 days?	Mon 12/17/07	Fri 12/21/07	8,9,10,11	Onur
13	**Determining Predicate, Subject and objects in a sentence**	**15 days?**	**Mon 12/17/07**	**Fri 1/4/08**		
14	Predicate	10 days?	Mon 12/17/07	Fri 12/28/07		Mazhar
15	Subject	10 days?	Mon 12/17/07	Fri 12/28/07		Sadettin
16	Objects	10 days?	Mon 12/17/07	Fri 12/28/07		Şerif,Onur
17	Testing & Debugging	5 days?	Mon 12/31/07	Fri 1/4/08	14,15,16	Onur
18	**Determining Units, Location & Time in a Sentence**	**10 days?**	**Mon 12/31/07**	**Fri 1/11/08**		
19	Units & Time	5 days?	Mon 12/31/07	Fri 1/4/08		Sadettin
20	Location	5 days?	Mon 12/31/07	Fri 1/4/08		Mazhar,Şerif
21	Testing & Debugging	5 days?	Mon 1/7/08	Fri 1/11/08	19,20	Onur
22	**Information Extraction**	**25 days?**	**Mon 1/7/08**	**Fri 2/8/08**		
23	Searching & Preparing Templates & Implementing	20 days?	Mon 1/7/08	Fri 2/1/08		Mazhar,Şerif
24	Testing & Debugging	5 days?	Mon 2/4/08	Fri 2/8/08		Onur
25	**Machine Learning**	**60 days?**	**Mon 1/7/08**	**Fri 3/28/08**		
26	Determining ML Algorithm & Implementation	30 days?	Mon 1/7/08	Fri 2/15/08		Sadettin,Onur
27	Testing & Debugging	10 days?	Mon 2/18/08	Fri 2/29/08		Onur
28	Training	20 days?	Mon 3/3/08	Fri 3/28/08	27	Mazhar,Şerif
29	**Spell Checking**	15 days?	Mon 2/18/08	Fri 3/7/08		Sadettin,Onur
30	**Development**	**40 days?**	**Mon 2/4/08**	**Fri 3/28/08**		
31	Try to Find New Techniques to Improve Efficiency & Reduce Cost	20 days?	Mon 2/4/08	Fri 2/29/08		Şerif,Mazhar
32	Implementation New Techniques	15 days?	Mon 3/10/08	Fri 3/28/08	31	Sadettin,Onur
33	Prapare GUI	10 days?	Mon 3/31/08	Fri 4/11/08		Mazhar,Şerif
34	Testing & Debugging of All Program	10 days?	Mon 3/31/08	Fri 4/11/08		Onur,Sadettin
35	**Search Module**	**30 days?**	**Mon 4/14/08**	**Fri 5/23/08**		
36	Search by Patient	5 days?	Mon 4/14/08	Fri 4/18/08		Şerif
37	Search by Abnormality & Disease	5 days?	Mon 4/14/08	Fri 4/18/08		Mazhar
38	Testing & Debugging	5 days?	Mon 4/21/08	Fri 4/25/08	36,37	Sadettin
39	Seam for Similar Reports	15 days?	Mon 4/21/08	Fri 5/9/08		Sadettin,Mazhar,Şerif
40	Testing & Debugging	10 days?	Mon 5/12/08	Fri 5/23/08	39,38	Onur


Dec 10, '07
Dec 17, '07
Dec 24, '07
Dec 31, '07
Jan 7, '08
Jan 14, '08
Jan 21, '08
Jan 28, '08
Feb 4, '08
Feb 11, '08
Feb 18, '08
Feb 25, '08
Mar 3, '08
dettin
zhar
Onur
Sadettin
Onur
Mazhar
Şerif
Onur
Mazhar
Sadettin
Şerif,Onur
Onur
Sadettin
Mazhar,Şerif
Onur
Mazhar,Şerif
Onur
Sadettin,Onur
Onur
Şerif,Mazhar
Project: Project1
Date: Mon 12/3/07
Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline
Page 2


Mar 10, '08
Mar 17, '08
Mar 24, '08
Mar 31, '08
Apr 7, '08
Apr 14, '08
Apr 21, '08
Apr 28, '08
May 5, '08
May 12, '08
May 19, '08
May 26, '08
Jun 2, '08
Mazhar,Şerif
dettin,Onur
Sadettin,Onur
Mazhar,Şerif
Onur,Sadettin
Şerif
Mazhar
Sadettin
Sadettin,Mazhar,Şerif
Onur
Project: Project1
Date: Mon 12/3/07
Task
Split
Progress
Milestone
Summary
Project Summary
External Tasks
External Milestone
Deadline
Page 3