Journal of the Islamic University of Gaza,Vol.11, No.2 P 126 –P145, 2003.

Distributed Electronic Health Care System (DEHCS)

Hatem M. Hamad Faculty of Engineering, the Islamic University of Gaza, B.O.Box 108 Gaza. Palestine

نظام الرعاية الصحية الإلكترونية الموزع

ملخص: إن الرعاية الصحية الإلكترونية هي الحل الأمثل لغلاء وبعد الخدمات الصحية التي لا تحجزهـا الحدود. ففي هذه الورقة ننوي تطوير وتحسين نظام الرعاية الصحية الإلكترونية في قطاع غزة اعتمـاداً على خدمات شبكة المعلومات الأنترنت، فمن خلالها نعرض لها حلاً برمجياً انطلاقاً من البنيـة التحتيـة المعتمدة على (CORBA) و لغة الجافا مقسماً على ثلاثة أجزاء أساسية هـي الوصـول إلـى قاعـدة البيانات من أي مكان وعرض الصور الطبية و اللقاءت الحية وهي التي تشكل اللب الأساسي لهذا الحل .

Abstract: A solution for the expense, distance, and barriers of health delivery is called telemedicine or electronic health care "E-health". In this paper we intend to improve the health care system in Gaza-Strip by proposing an E-health care system that is Web-based. This paper gives a software solution based on the distributed infra-structure based on CORBA and implemented by java programming language for the electronic health care in Gaza-Strip. Three main parts are available in the implementation of the system, remote access to database, medical image viewing, and video conferencing.

1. INTRODUCTION

E-health is very important for our country in the present situation. Nowadays citizen's movement is very difficult and dangerous and there is a shortage of physicians and some of them do not have the required experience to provide a medical quality of service. There is also a shortage in medical labs and equipments. This creates many problems in the service and an inefficient health care system. Electronic health "E-health" may represent the solution.

E-health refers to a health delivery system that provides health related activities at a distance between two or more locations using technology-assisted communications.

The World Health Organization (WHO) [1] uses the following broad definition for E-health (E-health is the delivery of health care services, where distance is a critical factor, by health care professionals using information and communication technologies for the exchange of valid information for diagnosis, treatment and prevention of disease and injuries, and for the continuing education of health care providers as well as research and evaluation, all of this is because of the interests of advancing the health of individuals and their communities). Although E-health is not new, recent years have seen a significant expansion of E-health applications associated

with rapid technological advances and dramatic reductions in costs [2,5,10].

Current E-health practice incorporates video conferencing for consultations, counselling, supervision, education, training and administration, medical image and data transfer, and access to databases and multimedia information [8].

E-health is used across a range of specialties, including mental health, radiology, pathology, paediatrics and renal medicine [9].

E-health, itself, is a broad area and it incorporates many disciplines; Thus, the glossary by necessity needs to encompass terms ranging from information technology and telecommunications to health information systems and management, and health services [6,7,11].

The next part talks about state of the art, part three talks about the problem, Part four discusses modelling in UML. The rest is about system packages, state diagram, system implementation and some screen shots.

2. DEHCS OVERVIEW

The Distributed Electronic Health Care System (DEHCS) serves to reduce the burden of hospitals and their staff. The purpose is thus to identify both functional and non-functional requirements [3,4,12,13].

2.1 Functional Requirements:

The DEHCS must provide the following functional properties:

- Allow doctors to retrieve patient records that have been previously sent from the DEHCS Patient. A patient record contains all personal information about the patient and all the conditions (past and present) that concern the patient.
- Allow doctors to retrieve meter readings that have been sent from the DEHCS Lab. A reading is the result (string) returned by the meter device that the patient is using.
- Allow doctors to retrieve messages that have been previously exchanged between the doctor and patient. A message is the text that was sent from one user to the other regarding the relevant issues between the doctor and the DEHCS Patient.
- Allow doctors to update medical information within patient records. This includes new conditions, dosages and medicines.
- Doctors should be able to generate reports based on the patient's readings and history.
- Allow the doctor to add a new patient into the system.

- Allow doctors to communicate with the DEHCS Patient.
- The doctor must be able to communicate their diagnosis and recommendations to the DEHCS Patients.
- Free text messages relating to relevant issues between the doctor and the DEHCS Patient.
- The DEHCS Doctor should allow the doctor to prioritise messages that are to be sent to the DEHCS Patient. This involves a message hierarchy between emergency messages and general queries.
- Allow doctors to receive communication from the DEHCS Lab.
- Doctors should be able to receive readings from various medical devices from the DEHCS Lab.
- The doctor should be able to receive any free text messages sent from the DEHCS Lab.
- The doctor must be able to change the critical emergency levels of the medical readings gathered by devices.
- The DEHCS Doctor must provide security to prevent unauthorized access to information and functions designated to a particular doctor. Thus doctors need to be authenticated.

2.2 Non-Functional Requirements:

The DEHCS must provide the following non-functional properties:

- The DEHCS must be portable between different platforms, Which then allows the DEHCS Patient and DEHCS Doctor components to communicate even if implemented in a different manner / programming language.
- The E-health care system is required to be distributed, thus it must be interoperable with existing hospital infrastructures. This will allow any new components to be added with minimal interruptions / downtime of the running E-healthcare system.
- The DEHCS Doctor should handle concurrency issues, as there will be potentially many users using the E-healthcare system. Doctors may handle many patients thus the DEHCS Doctor must be able to receive and send multiple messages concurrently.
- Latency of the DEHCS is crucial. Communication delays must be kept to a minimum and will vary according to priority.
- The security of data sent between the DEHCS Doctor and DEHCS Patients should be protected by restricted access in order to ensure doctor-patient confidentiality.
- The DEHCS must be user friendly and operated with minimal fuss.

- The system must manage and organize meter readings/messages received from the DEHCS Patient according to the priority specified by the DEHCS Patient.

2.3 Assumptions:

The following are some assumptions for DEHCS:

- All patient records are stored in a database monitored by the administration staff.
- Information about current patients is stored in the database.
- Each patient has a unique identifier.
- When a new patient is added he/she is provided with a unique identifier by the administration staff.

3. MODELING IN UML

UML stands for **U**nified **M**odelling **L**anguage; it is a standard language for specifying, visualizing, constructing, and documenting the artefacts of software systems, as well as for business modelling and other non-software systems [16]. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

The primary goals in the design of the UML were [16]:

1. Provide users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modelling language.

3.1 Types of UML Diagrams

Each UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML defines eight types of diagrams for the purpose of analysing any type of systems [17].

1. **Use Case Diagram:** displays the relationship among actors and use cases.
2. **Class Diagram:** models class structure and contents using design elements such as classes, packages and objects. It also displays relationships such as containment, inheritance, associations and others.

3. **Sequence Diagram:** displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects).
4. **Collaboration Diagram:** displays an interaction organized around the objects and their links to one another. Numbers are used to show the sequence of messages.
5. **State Diagram**: displays the sequences of states that an object of an interaction goes through during its life in response to received stimuli, together with its responses and actions.
6. **Activity Diagram:** displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. This diagram focuses on flows driven by internal processing.
7. **Component Diagram:** displays the high level packaged structure of the code itself. Dependencies among components are shown, including source code components, binary code components, and executable components. Some components exist at compile time, at link time, at run times well as at more than one time.
8. **Deployment Diagram:** displays the configuration of run-time processing elements and the software components, processes, and objects that live on them. Software component instances represent run-time manifestations of code units.

3.2 DEHCS Use case diagram

The use case diagram of the healthcare system describes what the system dose from the standpoint of an external observer; the emphasis is on what the system does rather than how Figure 1.

An actor can be a human, a system or an application that interacts with the system without being a part of the system. The main system actors are Healthcare professionals, Nurses, Technicians, Remote Consult and Diagnose (RCD) and Patients.

Each of these user groups need to interact differently with health related information in terms of information access, medical acts and security.

3.3 Acquisition, store and forwarded medical data

The use case diagram shown in Figure 2 and the sequence diagram shown in Figure 3 represent the sequence of events, which transpire between the remote X-ray technician and the radiologist reading the X-ray or any other medical image. The same scenario can be carried out for the results of medical tests or any medical information.

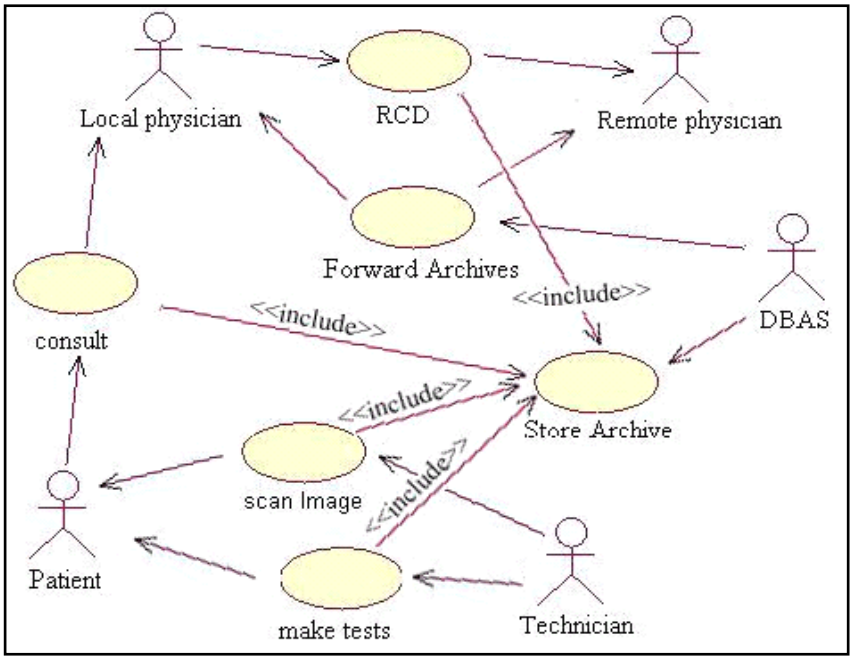


Figure 1: Use case diagram of the healthcare system

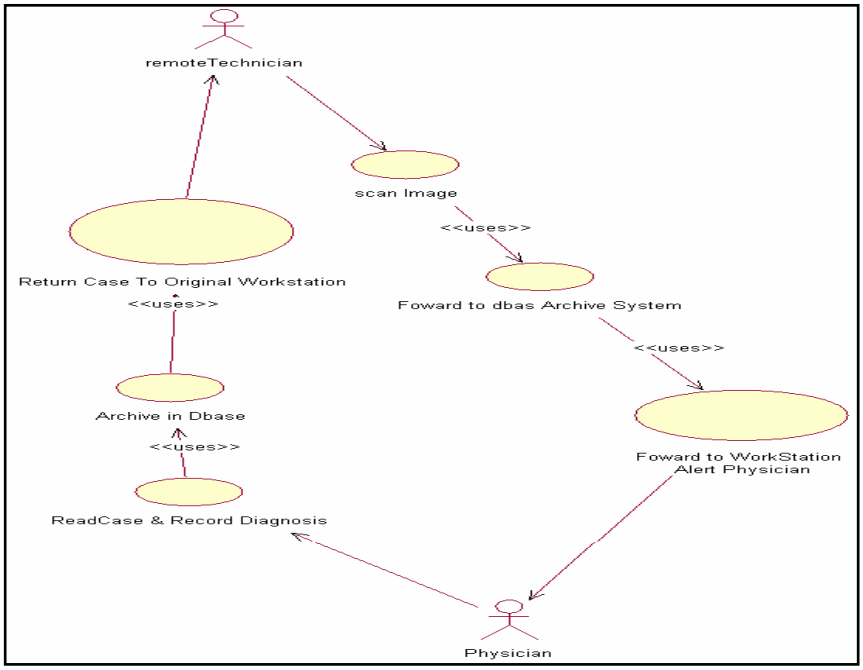


Figure 2: Use case diagrams for acquisition, store and forwarded images

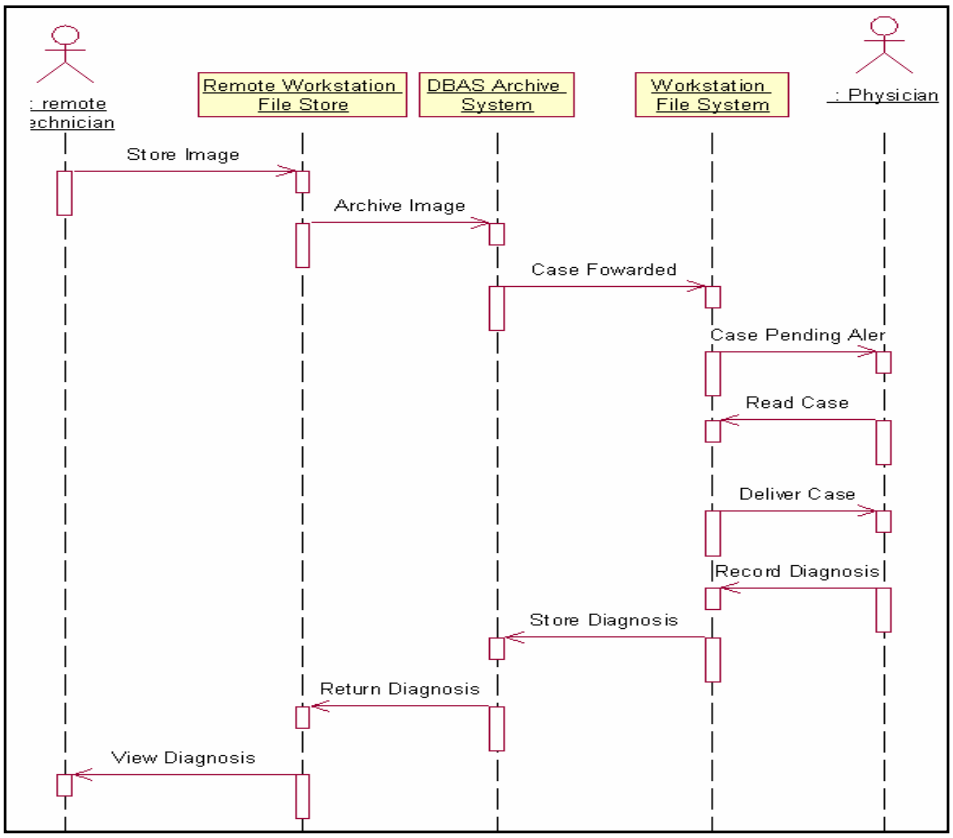


Figure 3: Sequence diagram for acquisition, store and forwarded images

3.4 Remote Consult and Diagnose (RCD)

The system enables physicians to perform remote consultation and diagnosis (RCD) over the Internet, or any private network. The multimedia RCD sessions can take place between two physicians in different geographical locations, including rural clinics. The use case and sequence diagrams depicted in Figure 4 and Figure 5 respectively describe the relation between two physicians.

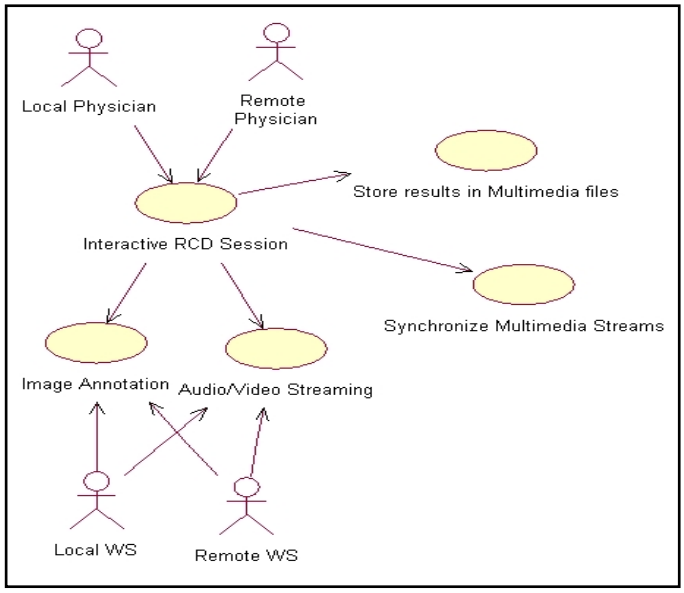


Figure 4: Use case diagram of RC

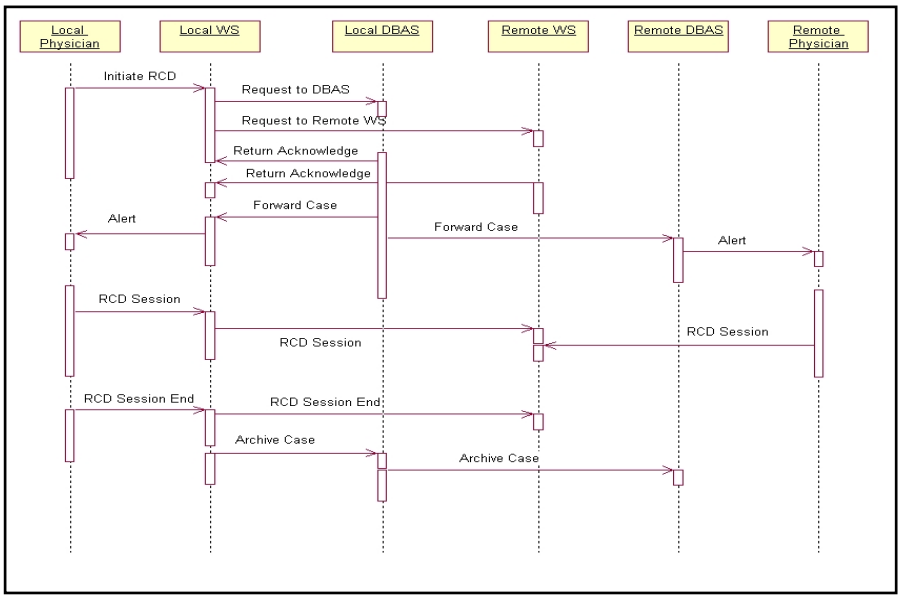


Figure 5: Sequence diagram of RCD

4. SYSTEM PACKAGES

This type of map tries to group the services from an Object Oriented point of view. Package diagram is the level that describes the main blocks in any system and its relations, if found, in brief details. But the packages of

classes should be made independent of each other. The name of the package and its global purposes should be known (Figure 6).

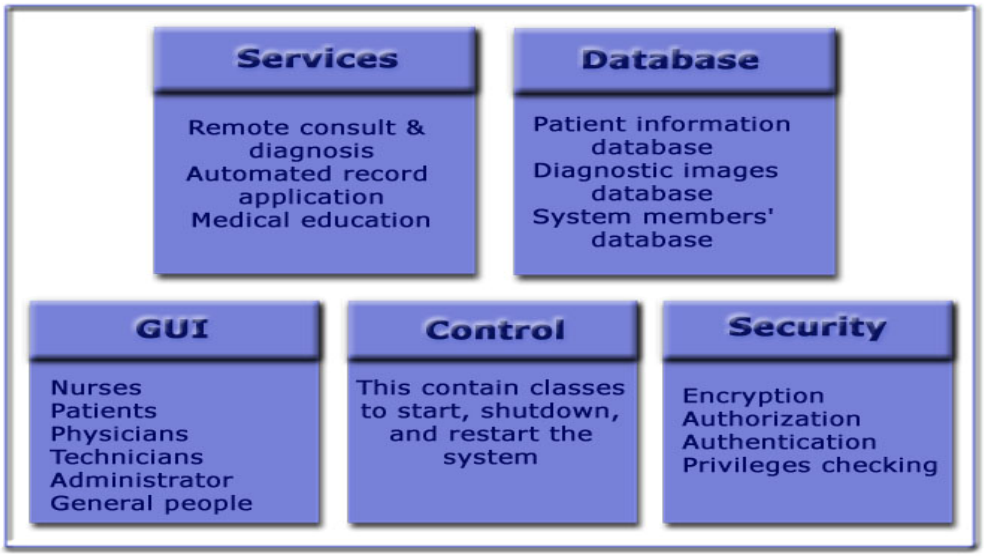


Figure 6: System packages

4.1 Graphical User Interface (GUI) package

We have several types of members each has its own behaviour in the system, so each one should have a different GUI to interact. Each class in this package represents the GUI for an appropriate type of users (Physicians, Patients, Nurses, Technicians, Administrators, Others).

4.2 Security package

As in any other system, the security of networks is very critical. There are four main issues all can be grouped in one package for security management (Authentication, Authorization, Encryption, Privileges checking).

4.3 System services

Services can be strategically grouped into three main categories: Remote consult and diagnosis, Medical education, and Automated record applications.

4.4 Database package

In health care system database can be divided into two main categories. Database related to patient's data whether it is demographic data or medical

information, and the other database is related to patient's diagnostic images and its related reports or annotation.

4.5 Control system package

This package contains classes that will deal with initiating the system, preparing the connection for both sides client and the server, restarting and shutting down the system.

5 SYSTEM STATE DIAGRAM

Figure 7 shows the developed state diagram for the DEHC system. It is important to remember that exit state can be reached from any other state (the arrows are not drawn to simplify the graph). Also note that the action performed to move from any member state to any of its services state is to choose that service (also not written on the arrows for limited space).

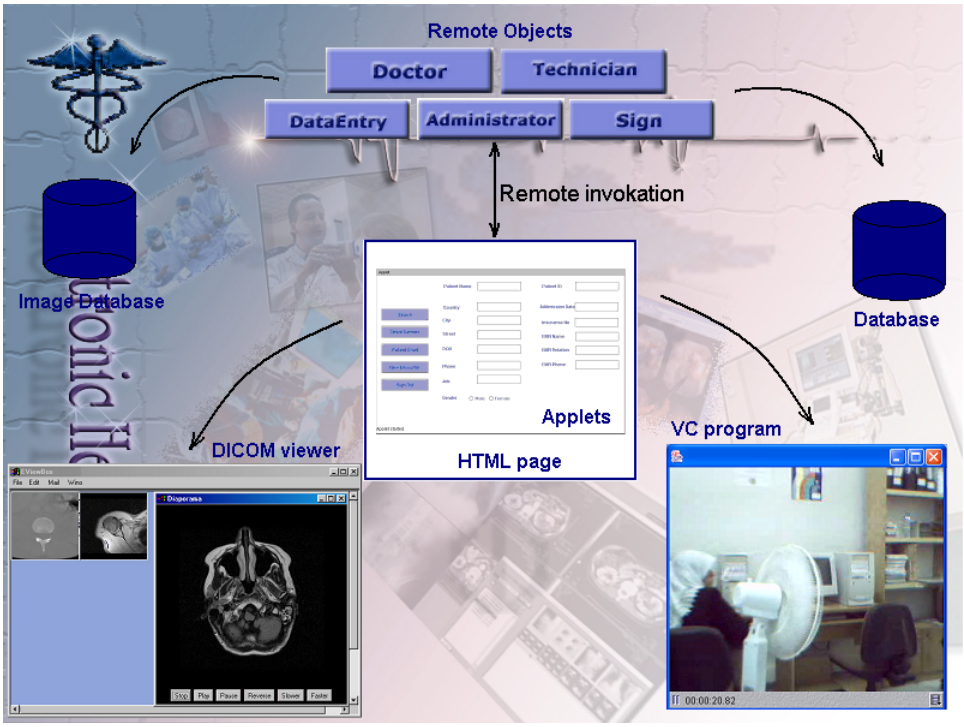


Figure 7: State diagram of the DEHC

6 SYSTEM IMPLEMENTATION

When designing and implementing distributed applications, many mechanisms exist by which such applications can be built. Depending on the nature of the application ranging from its complexity to the platform(s)

it runs on to the language(s) used to implement it there are a number of alternatives for a developer to consider. These alternatives include:

6.1 DistributedObjects Distributed frameworks such as Common Object Request Broker Architecture (CORBA), Remote Method Invocation (RMI), and Distributed Component Object Model (DCOM) allow distributed applications to interoperate (application-to-application communication), regardless where these objects reside (Figure 8).

They use an object-oriented approach for creating software components that can be reused and shared between applications. Each object encapsulates the details of its inner workings and presents a well-defined interface, which reduces application complexity. The cost of developing applications is reduced because once an object is implemented and tested; it can be used over and over again.

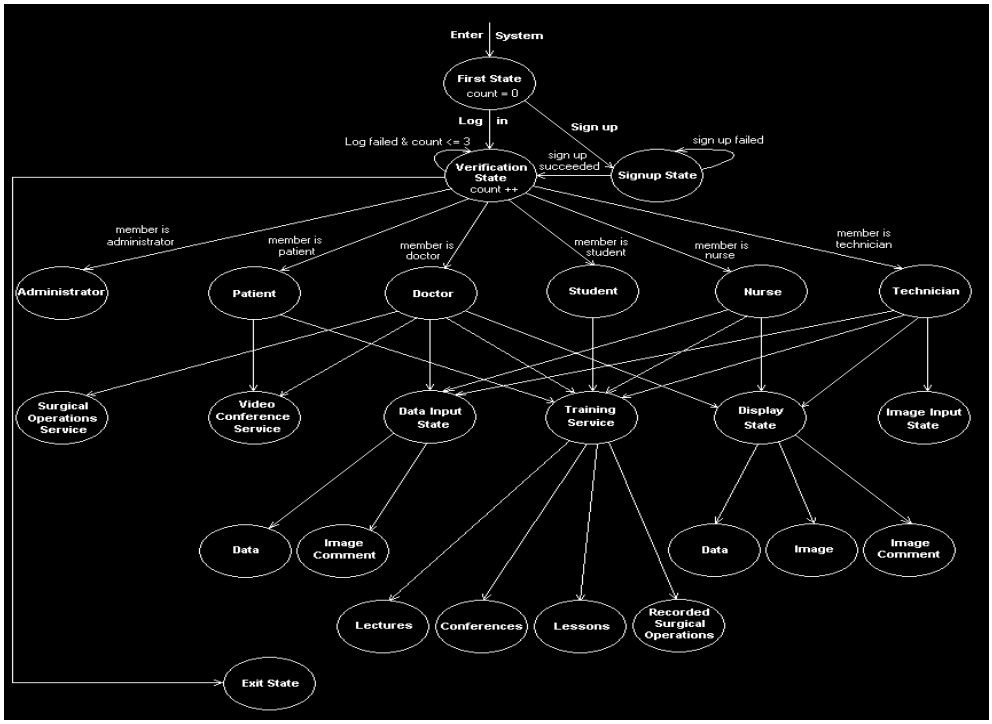
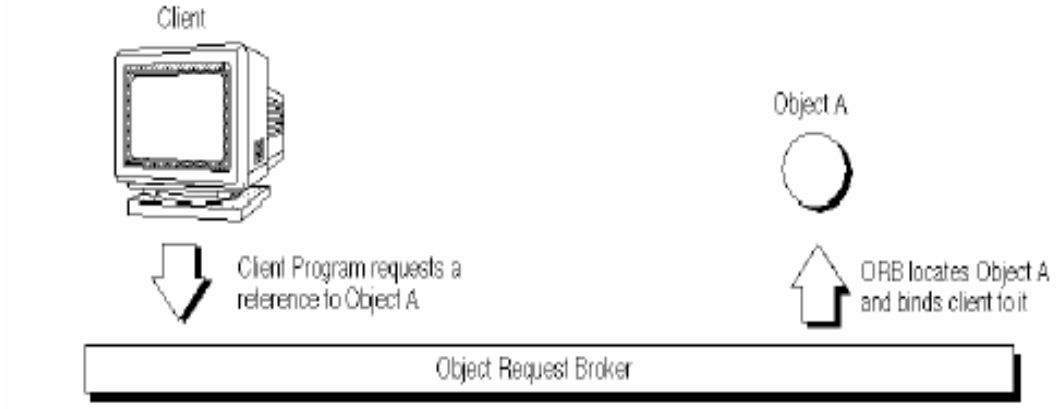


Figure 8: Client program acting on an object

6.2 Common Gateway Interface (CGI) and Servlets

CGI and Servlet provide server-side processing of requests from a client browser. A client program written in any language sends



requests to the server. The client can be as simple as an HTML page. On the server side, the CGI program processes the request, and generates dynamic output, which is sent back to the client. CGI usually don't have a UI, but you can optionally provide one on the client side.

6.3 JavaServer Pages (JSP)

The new JSP API, part of the Java 2 Enterprise Edition (J2EE), gives Web and Java developers a simple yet powerful mechanism for creating these sorts of applications. JSP provides developers with two important abilities. First, it provides the ability to access remote data via mechanisms like Enterprise Java Beans (EJB), Remote Method Invocation (RMI), and Java Database Connectivity (JDBC). Second, it lets developers encapsulate and separate program logic (Java code) from the presentation (HTML tags), to help maximize code reuse and flexibility. This separation of logic and presentation is a major advantage over other Web application architectures, like Java Servlet and CGI scripts.

6.4 Which implementation alternative to be used?

Because healthcare is rapidly taking on a distributed nature, it should be regarded as a collaborative tool that can incorporate all types of data including sound, images, and video in an integrated manner even though the sources of data and processing may be widely distributed. This requires a robust distributed computing infrastructure accessible on a wide range of platforms. The premier open architecture, which can meet this requirement, is CORBA (Common Object Request Broker Architecture) [15].
CORBA is based on the Object Management Group's (OMG).

7. SYSTEM'S SCENARIO

The system objects operate and interact with each other according to the following scenario:

For any member to join the system, a manual or electronic payment is made through the system administrator, who adds the new member to the system giving him a unique secret code. This code is entered with other information by the new member in the sign up operation as a security check process. The sign up operation first checks for a correct and available secret code for a new member in the system. Then, if the member is working in medicine field, the entered job information is checked to be correct as stored in the health ministry database. Also, for a registration to be correct the entered user name and password should be the first in the system. Repeating user names and passwords is not allowed and the member is asked to change it if this occurs. After all of this is done, the member is successfully registered in the system as one of: doctors, nurses, technicians, patients, medical students, or administrators.

Through the data entry methods, a new patient will have a medical record after visiting any system participant medical centre. Doctors, technicians, or any medical staff members can deal with patients' records according to their privileges.

A member doctor can view all of the patient's demographic and medical data. The data may be classified by the patient's encounters or as a clinical summary. Doctors also may select any lab test and view its results. Selected medical images also can be electronically viewed via a special medical image program. Member doctors at any medical centre participating in the system can enter a new patient's encounter ordering any lab test or medical image for that patient.

It is the technician's responsibility to enter lab test results, images and their comment in the system. When a technician enters the patient's ID, all tests and images ordered for that patient in the last encounter will be listed down. Technicians then perform the tests or take the required images and enter the results.

The doctor and patient members have videoconference option that allows doctors to perform remote consultation with other online member doctors or remote diagnosis with member patients.

Finally, all system members can access the medical learning centre for general and specific medical information.

7.1 Class diagram

Classes are an integral feature in object-oriented designs. A class refers to a set of objects that have similar states and behaviour (Figure 9). An instance of a class is called an object.

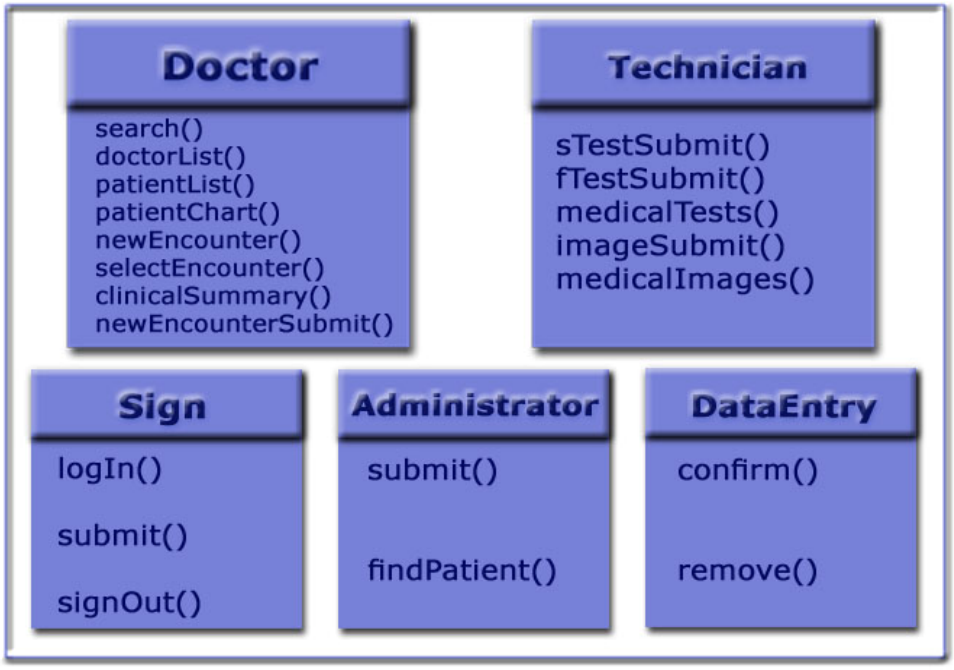


Figure 9: Class diagram

7.2 DICOM viewer

All the physicians, radiologists, and people who work with medical images can look at radiographic films, which are displayed on big and roomy light boxes. But these radiographic films are expensive, they reject chemical polluting agents such as silver nitrates and are heavy and difficult to store. Meanwhile, medical imaging use more and more digital imaging technologies for creating images such as CT, US, MRI, angiographies, digital radiography and fluoroscopy. All these modalities, originally viewed on computer screens are translated into conventional films in order to let other physicians view them on light boxes.

Since **D**igital **I**maging **Co**mmunication in **M**edicine (DICOM) is becoming a standard for digital image storage in medicine, the idea is to replace the old physician light box by a simple DICOM Viewer which uses little room in computer's memory and lets reading the images like the radiologist on a computer screen [14]. DICOM Viewer is a java applet that allows you to view many kinds of images, including medical DICOM images.

When a doctor requests a particular diagnostic image for a patient, it is the role for the radiologist to apply this demand. This is done by scanning the image then storing it in a folder holding the patient ID as a name, this folder is found in a web server. Then doctor can view this image by opening DICOM Viewer program from medical images button. After that he can select any image from the medical images list particulated by that patient. Selecting an image from the list will access this image by its URL to view it, as shown in Figure 10 and Figure 11.

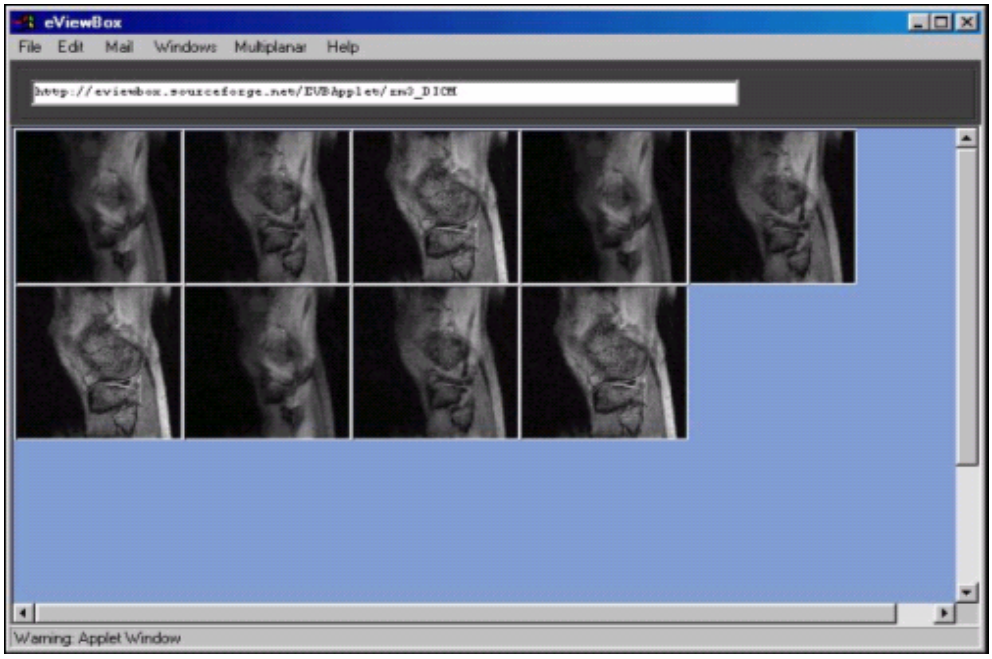


Figure 10: X-ray images opened with DICOM viewer

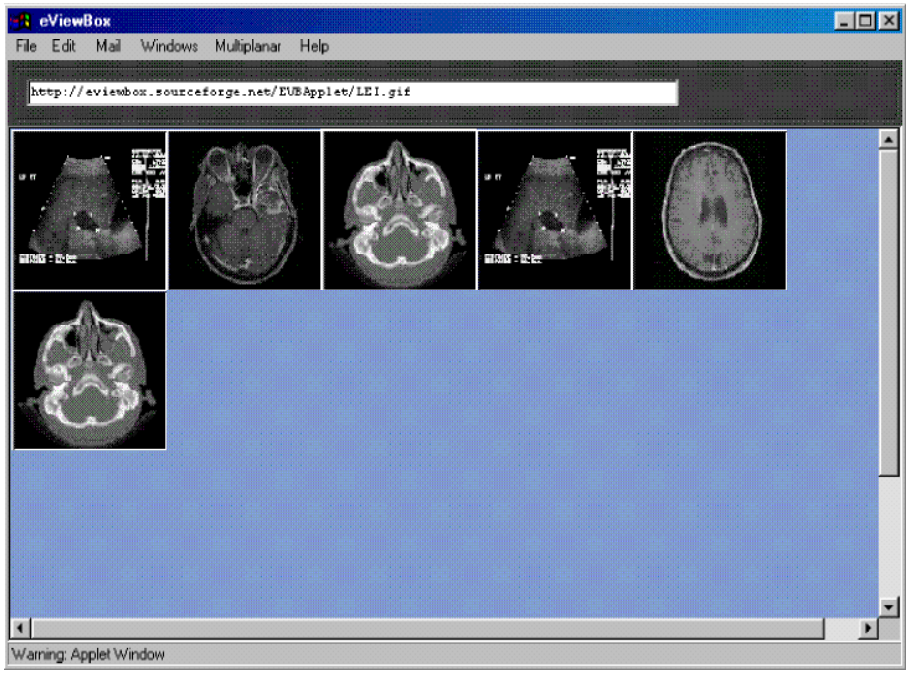


Figure 11: CT images opened with DICOM viewer

7.3 Video conference implementation

The video conferencing program in the system is launched by pressing the Video Conference button on the doctor's and the patient's graphical interface. The program is implemented using Java Media Framework (JMF) package. This package is responsible for audio and video media streaming. The program is a point-to-point application. One party operates as a server, and the other as a client. This means, that for more than one conference with one party, a new connection with new streams is established.

7.4 System screen shots

Some of the Graphical User Interface (GUI) designed for the system is shown in the Figures 12 through Figure 16. Each client has its applet GUI that supports all of the activities that can be accomplished by a client.

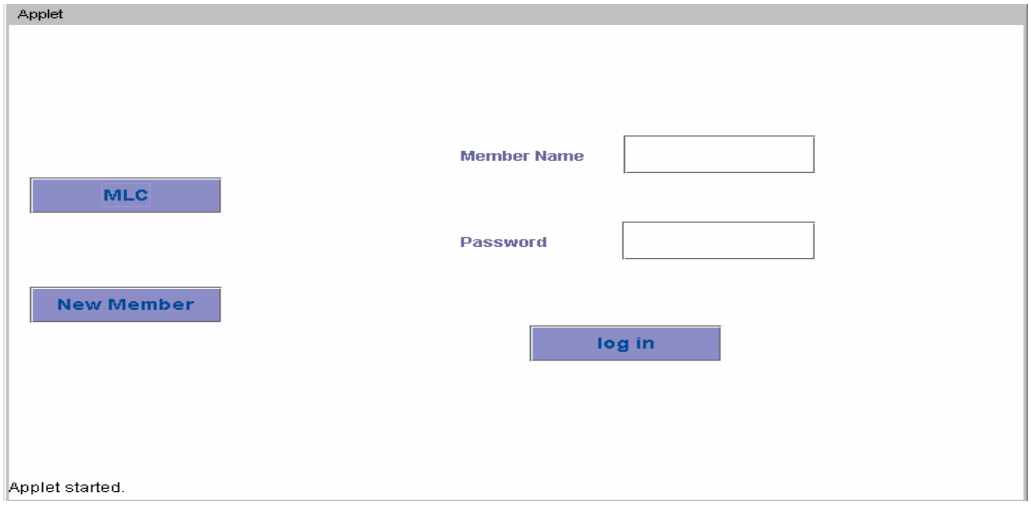


Figure 12: Sign In

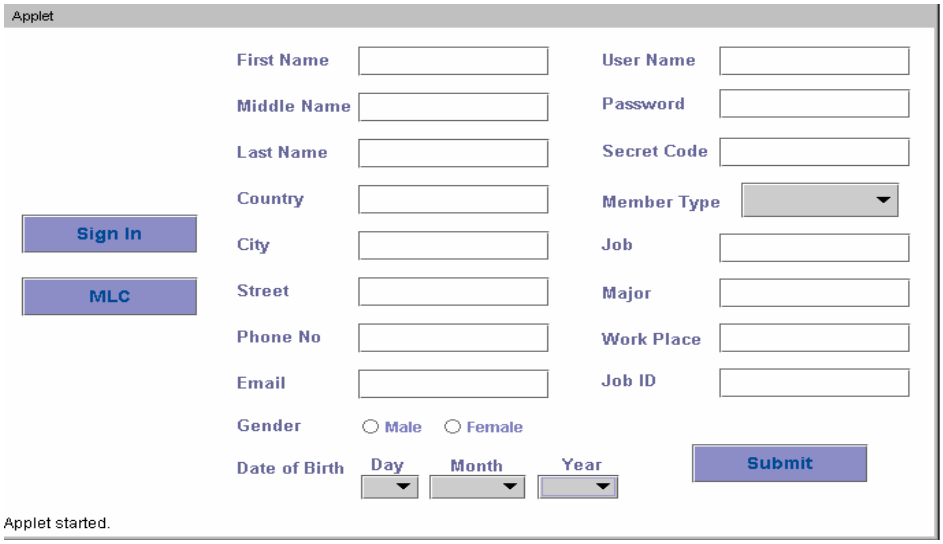


Figure 13: Sign Up

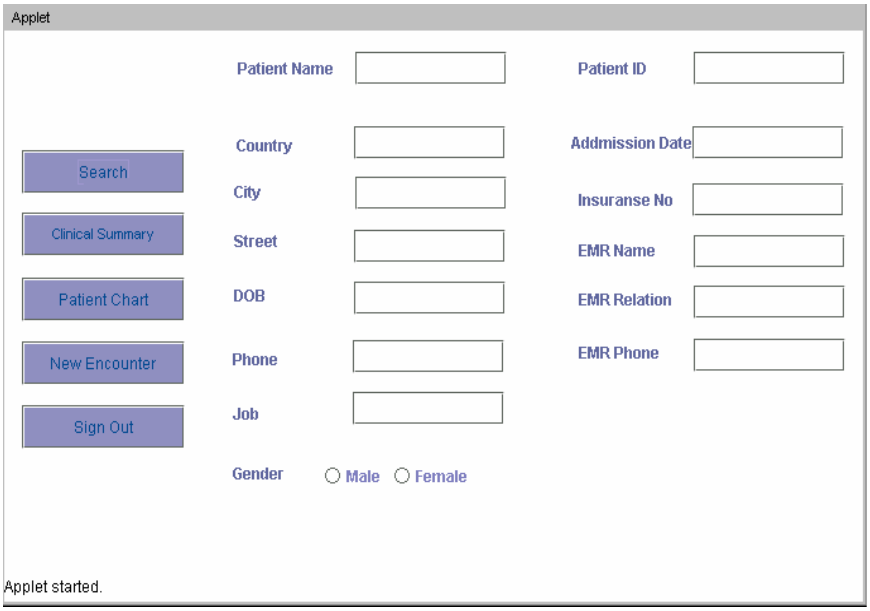


Figure 14: Doctor: search

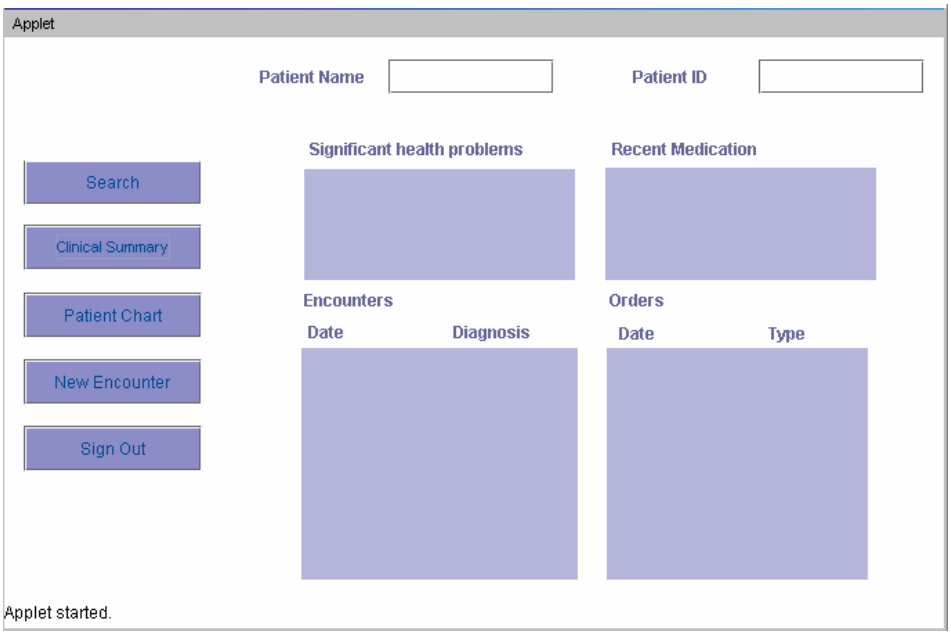


Figure 16: Technician: fluid test

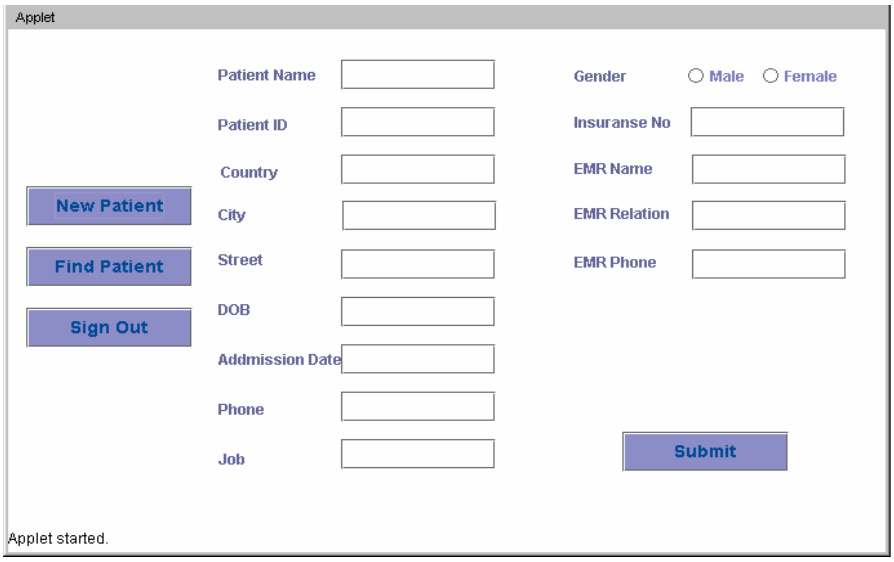


Figure 17: Data entry: new patient

8. SUMMARY

The E-Health care system is designed with the intention of creating an alternate means of (direct) communication between a doctor and a patient. The system will work in harmony with the existing technologies, fitting into the current infrastructure, using existing hospital (medical) databases as well as allowing the implementation of new databases. The system facilitates doctors to receive current vital statistics of patients (e.g. blood pressure, heart rate, body temperature) and descriptions of symptoms/problems from patients, as recorded by (or entered into) the patient's side of the system. Doctor is also able to access the patient's background and medical history. Doctors are able to formulate and respond with appropriate directions, diagnosis and prescriptions as needed. In essence the system strives to reduce the burden on the existing medical system as well as to increase its efficiency. Future work should use our model in developing industrial applications in similar fields such as E-Learning and E-commerce. The system was tested by simulation but not in physically. The actual implementation of this project is out of the scoop of this paper due to the lack of funding and resources. Another future work is to implement this project by the health system personnel in Gaza.

9. REFERENCES

1. Lundin, R., 1999, Telemedicine: Applications and Technologies, A report for the Project on Rural Health Communication and Information Technology (PRHCIT).
2. Grigsby, J., Schlenker, R.E., Kaehny, M.M., Schaughnessy, P.W., & Sandberg, E.J., 1995, Analytical Framework for Evaluation of Telemedicine, Telemedicine Journal, Vol 1, No 1, pp 31-39.
3. National Telehealth Committee, 1996, Australian Telehealth Services Issue Paper, Oct.
4. Bashshur, R.L., 1995, Telemedicine Effects: Cost, Quality, Access, Journal of Medical Systems, Vol 19, No 2, pp 81-91.
5. Friedman, C.P. & Wyatt, J.C., 1997, Evaluation Methods in Medical Informatics, Springer, New York.
6. Telemedicine Glossary, 1998, Telemedicine Today 1998 Buyers Guide & Directory, pp 54-63.
7. Ringwall, E.S., The Journal of the Telecommunications in Telemedicine Healthcare Information Management Systems Society, Volume 13 Issue 1, Spring 1999, pp 49-65.
8. www.lib.uiowa.edu/telemed/
9. www.med.usf.edu/CLASS/telmed
10. www.drgdiaz.com/telesonography.shtml
11. www.tft.tele.no/telemedisin/telektronikk
12. www.radiology.uiowa.edu/MoreRAD/Teleradiology/Tele.html
13. www.medical.nema.org/
14. www.psyc.nott.ac.uk/staff/cr1/dicom.html
15. www.execpc.com/~gopalan/corba/corba.html
16. www.pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/
17. www.smartdraw.com/resources/centers/uml/uml

\

Journal of the Islamic University of Gaza,Vol.11, No.2 P 126 –P145, 2003.

131 132 133 134 135 136 137 138 139 140 141 142 143 144 145