E-Healthpro+ Using Java Swing, AWT (Abstract Window Toolkit) & MySQL

Shivam Kumar, Waseem Khan, Sanjana Kumari, Sumit Singh, Fahad Zafar, Yashvendra Singh
Department of Computer Science and Engineering VCTM Aligarh

Abstract- In the Current Scenario, Healthcare industries are facing many challenges for managing patient records, staff records, appointments, and administrative details. It provides better result for these occurring challenges. There are many challenges facing by different healthcare industries. Challenges like, managing different appointments of patients, high cost, limited access, managing staff records, managing fast billing process, managing doctor details, some receptionist module issues, security issues, integration of emerging technologies. So, this paper represent the Java based healthcare system solves these challenges easily which contains basically four modules like patient module, doctor module, receptionist module and administrative module. Patient module refers to maintaining patients records and stores lot of patient records in a single table. Doctor module solve the challenges related to doctor records. Receptionist module looks for maintaining managing new patients, discharging patients, and other details of staff and patients. Administrative module manages these three modules like doctor records, patient records, receptionist activities.

I. INTRODUCTION

In the current scenario as we all are aware that different healthcare industries facing many health care problems, such managing data of lot of patients, doctors in a single table , high cost, fast billing process, appointments of patients etc.
This java based healthcare system solves these problems efficiently.
So, this java based health care system contains four modules for managing the problems occur in healthcare industries [1]. This system is designed by remembering all the current scenarios and past scenarios of healthcare industries in figure 1.

Four modules are:
1. Patient module
2. Doctor module
3. Administrative module
4. Receptionist module

This healthcare is designed by using these technology Java Swing, java AWT(Abstract Window Toolkit) and MySQL for maintaining huge amount of data. Netbeans platform is used to write code about project [1].

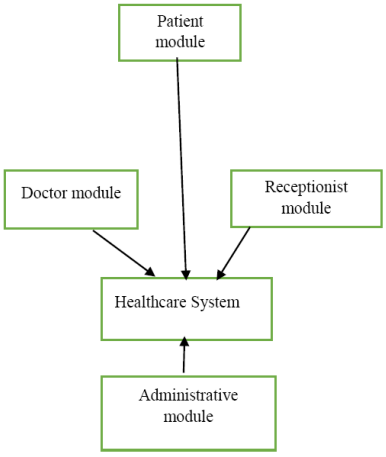

Figure 1: Modules

This java based healthcare system is well designed software solution offers robust and scalable solution that provides wide range of functionality in figure 2. Therefore, this robust health care system is an important enhance productivity, improve patient care, and optimize resource solution[2].

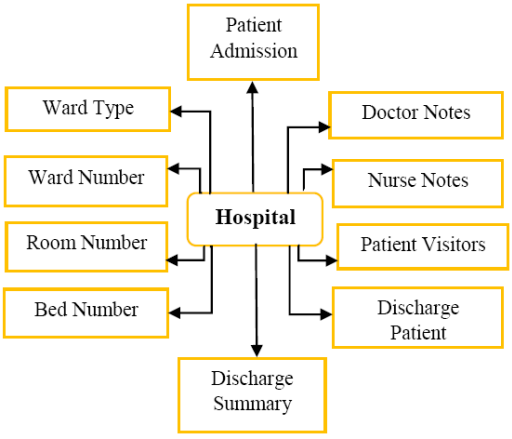

Figure 2: Hospital Structure

The user will go through few registers. Time can be lost because of this. Then, Development team are designing a comprehensive system related to solve this issue. This system will help to enhance the performance of hospital activities[2]. So, the users

would have like to have this system that makes hospitals work efficiently. The main goal of this healthcare system project is to make easier all management processes, comprise patient registration, doctor appointments, and compound writing.

Therefore, there is description of these four modules Administrative module, Patient module, Doctor module, Receptionist module- Administrative module: This module in healthcare system plays an important role in managing the assets, operational and other hospital activities. This module covers other three module included in this system. There are some key benefits like patient registration and control, staff management, billing and assets management, communication and management. Patient module: This module responsible for arranging patient related information, appointments scheduling, reducing wait times. This module reduces errors, assure compliance with fitness, and enhance overall performance and efficiency with in the hospital sustainability. In this module, patient can easily view their medical records, test results, and treatment history. Patient can check availability of doctors in hospitals and book their appointments easily and this system provides timely patient care.
Doctor module: This module is necessary for efficiently managing the staff of hospital. This module maintains complete profiles for doctors in hospitals, like their qualification, experience, and their availability. This module refers to monitoring key points like patient satisfaction, treatment results. This module is essential module to manage all the data related to hospital's doctors. This module refers to patient consulting doctor, specialized doctors and reference doctors. This module may allow to administrative for arranging the details about doctors.
Receptionist module: This module refers to receptionist to manage patient check ins and outs, other demographic information, monitoring registration process. This module having other key benefits like streamlined patient registration, improved appointments scheduling, increase patient experience.

II. LITERATURE REVIEW

Literature review of the healthcare system provide the actual body of research on the access and equity, quality of care, financing, workforce, different technologies. In literature review, organizations faces many problems related to hospital activities. There are different errors occurs across different departments of hospitals for managing huge amount of data manually[3]. A literature study of healthcare systems looks at the amount of research that has been done on how healthcare services are organized, paid for, and provided. Here is a quick summary:

Principal Themes:
1. Access and Equity: Research reveals inequalities in healthcare outcomes and access, especially for minorities and low-income groups.
2. Care Quality: Research aims to enhance the quality of healthcare, encompassing patient-centered care, safety, and efficacy.
3. Healthcare Financing: Research studies look at how various financing models (such as single-payer and multi-payer) affect healthcare expenditures and results.
4. Healthcare Workforce: Studies look at burnout, the need for interdisciplinary cooperation, and workforce shortages.
5. Healthcare Technology: Research looks into how digital health tools, such electronic health records and telemedicine, might enhance the way healthcare is provided.

Approaches:
1. Systematic Reviews: thorough evaluations of current. The current corpus of research on the planning, funding, and provision of healthcare services is examined in a literature review on healthcare systems[3]. Here is a quick synopsis: literature on particular medical subjects.
2. Comparative Studies: Evaluations of various healthcare models or systems.
3. Qualitative Research: Comprehensive analyses of healthcare views and experiences.
4. Quantitative research: statistical evaluations of trends and results in healthcare.

Important Results:
1. Single-payer systems typically result in better healthcare outcomes and cheaper administrative expenses.
2. Patient satisfaction and health outcomes are enhanced by patient-centered care practices.
Future Directions and Gaps:
1. To guide policy, more study on healthcare system comparisons is required.
2. To improve healthcare quality, more research on the wellbeing of healthcare workers is necessary.

3. There has to be a greater emphasis on social determinants of health and healthcare inequities.
4. More research is needed to determine how digital health technologies affect healthcare expenditures and outcomes

Healthcare outcomes are often better and administrative expenses are generally lower in single-payer systems. Methods of patient-centered care enhance health outcomes and patient satisfaction. To address healthcare disparities, a diverse staff is crucial[3]. Digital health technology can enhance healthcare quality and accessibility.

III. METHODOLOGY

The figure 3 shows a classic software development lifecycle (SDLC) paradigm called the Waterfall paradigm. An explanation of each stage in the process of creating a hospital management system (HMS) is provided below[4]:

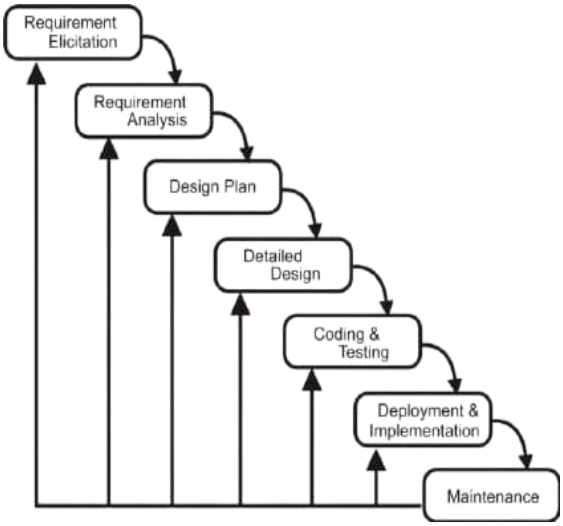

Figure 3: Iterative waterfall model

1. Requirement Elicitation: Ask patients, administrators, nurses, and other stakeholders for their needs. List essential features such as electronic medical records (EMR), billing, appointment scheduling, and patient registration. Specify the requirements for system security, compliance with HIPAA, and GDPR, among other healthcare standards.
2. Requirements Analysis: Evaluate the requirements gathered to make sure they are comprehensive and feasible. Sort features according to their importance and level of complexity. Specify the system's processes (such as doctor-patient interactions and patient admission to discharge).
3. Design Plan: Create the system's high-level architecture, defining the front-end, back-end, and database components. Select suitable technologies (such as cloud storage, SQL, Python, and Java). Make a Plan integration with outside programs, such as insurance companies and lab systems.
4. Detailed Design: To depict system components, create intricate diagrams (ER diagrams, class diagrams, and data flow diagrams). Specify the database schema that will be used to store patient records, physician information, and billing data. UI/UX design should be planned for patient and medical staff ease of use.
5. Testing and Coding: Use programming languages and frameworks to put the system into operation. In order to find bugs, do system, integration, and unit testing. To guarantee user authentication and data protection, conduct security
6. Implementation & Deployment: Install the HMS on a cloud-based server or in an actual medical setting. Teach hospital employees how to use the system. Prior to a full-scale rollout, conduct preliminary user acceptance testing (UAT).
7. Maintenance: Track system performance and fix any problems. Updates and security should be provided. patches. As the hospital grows and new features are requested, scale the system accordingly.

A. Technology used:

Patient records, doctor information, appointments, billing, and other hospital operations are managed using a desktop application called a Hospital Management System (HMS) that uses Java Swing, AWT, and MySQL. A summary of the technologies utilized is provided below: Utilized Technologies: AWT (Abstract Window Toolkit) and Java Swing:[7] A GUI toolkit called Swing offers a wide variety of elements, including tables, text fields, buttons, and more. Basic UI elements like windows, buttons, and layouts are handled by the lightweight GUI toolkit known as AWT. Swing vs. AWT: Swing is favored because to its superior appearance, feel, and richer components.

HMS Swing Components Used: The primary application window is JFrame. UI elements are grouped using JPanel. JButton: Action buttons for users. JTextField: Data entry input fields JLabel: UI element labels. JTable: Shows appointments, doctor information, patient records, etc. To allow scrolling for huge data tables, use JScrollPane. JComboBox: A drop-down menu for choosing a department, doctor, etc[8]. JDateChooser: To choose an appointment or

admittance date. JOptionPane: Alert and confirmation dialog.

2. Database Management, or MySQL Hospital data, such as patient information, physician information, medications, and billing, are stored and managed using MySQL. MySQL and Java Swing are connected using JDBC (Java Database Connectivity). Hospital data, such as patient information, physician information, medications, and billing, are stored and managed using MySQL. MySQL and Java Swing are connected using JDBC (Java Database Connectivity).

B. Proposed System:

Patient records, doctor schedules, billing, and pharmacy administration are all managed by a desktop program called a Hospital administration System (HMS) that uses Java Swing, AWT, and MySQL. Overview of the Proposed System[10]

The Graphical User Interface (GUI) of the HMS will be constructed using Java Swing and AWT, and patient, staff, and inventory data will be stored in a MySQL database.
System Modules: 1. Role management and user authentication Administrator, Physician, Receptionist, Druggist, and Nurse Secure login with a password and username

2. Management of Patients Sign up for new patients. Examine, amend, and remove medical records assign patients to physicians. 3. Management of Doctors Keep track of the doctor's information (specialization, availability) Assign patients to physicians. Observe appointment times 4. Making Appointments Schedule patient appointments. Check the availability of doctors Make an appointment reminders. 5. Management of Billing and Payment create treatment and medication bills[11]. Payroll tracking (paid, pending) 6. Management of Pharmacy Maintain and store the medication supply Keep tabs on medication sales and expiration dates. Create prescriptions 7. Management of Labs and Tests Document patient test results. Assign laboratory tests in accordance with physician advice. 8. Analytics & Reports Create reports (patient history, daily earnings) View hospital performance statistics in a graphical format. The stack of technologies[12].
Frontend: AWT, Java Swing Java is used for the backend (JDBC for database connection). MySQL is the database. Development Tools: NetBeans, Eclipse/IntelliJ IDEA Libraries: JasperReports (for reporting) and MySQL Connector/J.

Flow of the System
1. The user enters their login information.
2. The user is taken to the appropriate dashboard based on their function.
3. Tasks like scheduling appointments, billing, and patient registration are carried out.
4. The MySQL database is used to store and retrieve data utilizing.

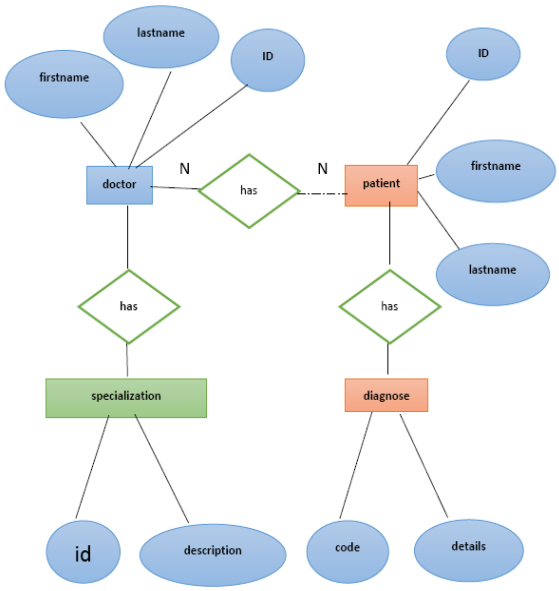

Figure 4: ER-Diagram

A Java Hospital Management System (HMS) Entity-Relationship Diagram (ERD) shows different entities, their characteristics, and their connections in figure 4. The components of the ERD are described as follows: Things and Their Characteristics 1. The patient ID of the patient (primary key) Name, Age, Gender, Address, Number of Contact,History of Medicine 2. Physician Physician_ID (Main Key), Name, Expertise, Number of Contact, Status of Availability. 3. Scheduling Appointment_ID (Key Principal) Foreign Key for Patient_ID, The Foreign Key for Doctor_ID, Date, Duration, Situation, 4. Health Information, Record_ID (Key Principal), Foreign Key for Patient_ID, Making a diagnosis, Prescription drugs, Details of the Treatment, Date[13] 5. Nurse, Nurse_ID (Key Primary), Name, Change Timing, Assigned Ward 6. Ward, Ward_ID (Key Primary), Type of Ward, Capacity, 7. The room, (Primary Key) Room_ID, (Foreign Key) Ward_ID, Type of Room, Status of Availability, 8. Invoicing the primary key, Bill_ID, Foreign Key for Patient_ID Quantity, Status of Payment, Date 9. Drugstore ID of

Medicine (Main Key) Name of Medicine Availability of Stock Cost 10. Lab_Test, Test_ID (Key Primary)A Java Hospital Management System (HMS) Entity-Relationship Diagram (ERD) shows different entities, their characteristics, and their connections. The components of the ERD are described as follows: Things and Their Characteristics 1. The patient, ID of the patient (primary key), Name, Age, Gender, Address, Number of Contact, History of Medicine. 2. Physician, Physician_ID (Main Key), Name, Expertise, Number of Contact, Status of Availability 3. Scheduling Appointment_ID (Key Principal, Foreign Key for Patient_ID, The Foreign Key for Doctor_ID, Date Duration, Situation, 4. Health Information, Record_ID (Key Principal), Foreign Key for Patient_ID, Making a diagnosis, Prescription drugs, Details of the Treatment, Date[15], 5. Nurse, Nurse_ID (Key Primary), Name, Change_Timings, Assigned Ward, 6. Ward, Ward_ID (Key Primary), Type of Ward, Capacity, 7. The room, (Primary Key) Room_ID, (Foreign Key) Ward_ID, Type of Room, Status of Availability 8. Invoicing The primary key, Bill_ID, Foreign Key for Patient_ID, Quantity, Status of Payment, Date 9. Drugstore, ID of Medicine (Main Key), Name of Medicine, Availability of Stock, Cost, 10. Lab_Test, Test_ID (Key Primary).

The index page in figure 5, which offers navigation to various user roles, is the primary entry point in a hospital management system (HMS). Usually, it has buttons or links that lead to the admin, patient, doctor, and receptionist modules. Overview of the Index Page.

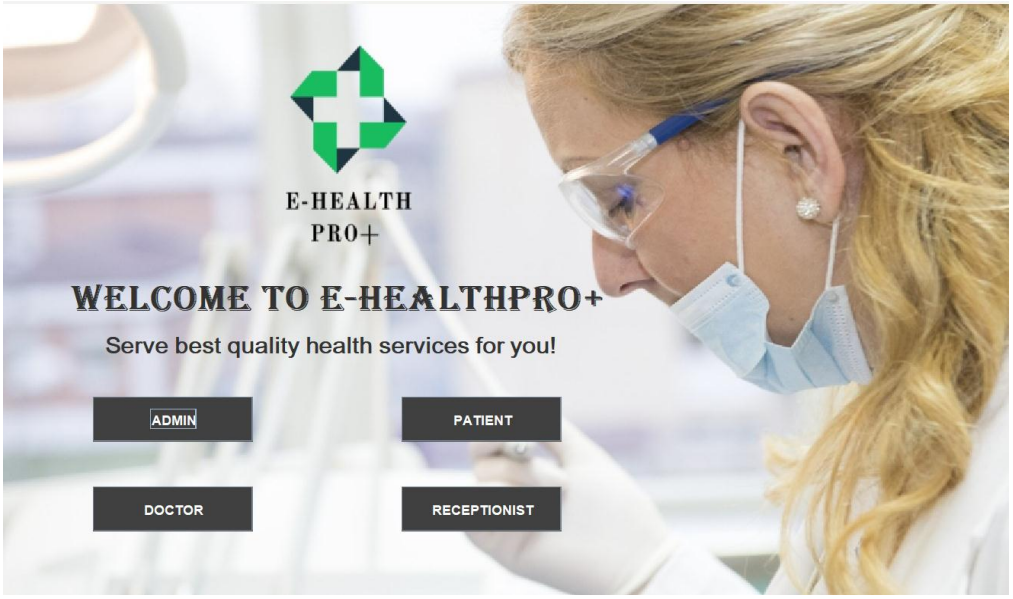

Figure 5: Index Page

The hospital management system's index page contains: a place for users to enter their login information. A navigation panel or dashboard to access various components. Role-based authorization and authentication, which restricts access to particular modules to authorized users only.

Description of Modules

1. The administrative module: The administration module oversees all aspects of the hospital system, including operations, data, and users.

2. The Module for Patients: Patients can manage their appointments, medical records, and bills with the help of the Patient Module.

3. The Module for Doctors: The Doctor Module facilitates the management of physicians' appointments, schedules, and patient information.

4. Module for Receptionists: The Receptionist Module helps manage front desk operations at hospitals.

Include a page with doctor details in the hospital management system. The administrator or receptionist registers new physicians in the hospital administration system using the Add Doctor Details page. It has backend logic to save the data in the database, validation checks, and a form to enter the doctor's details.

1. The Add Doctor Page's features, This form asks for the doctor's name, specialization, contact information, email address, qualifications, and availability. Validation checks: Verify the required fields, phone number length, and email format. Data is stored in the Doctor table via a database

connection. Following doctor registration, a success message and redirection appear.

MySQL Database Architecture for Hospital Management System (HMS) Patient records, physician information, appointments, medical history, and billing are all stored and managed in the Hospital Management System (HMS) database. A comprehensive MySQL database schema for HMS can be found below.

IV. CONCLUSION

An effective and methodical way to digitize hospital operations, the Hospital administration System (HMS) improves patient care, physician administration, appointment scheduling, and billing. It enables rapid access to patient and hospital data, minimizes paperwork, and does away with manual record-keeping. Principal Advantages:

Enhanced Efficiency: Reduces errors and time consumption by automating administrative procedures. Better coordination between physicians, receptionists, and patients is ensured via enhanced patient care.

Secure Data Management: Safely handles and keeps private patient data. Fast Access to Records: Provides real-time access to billing, appointments, and patient history. Among the main advantages of a hospital management system are: 1. Simplified processes and less work for the administrative staff 2. Better safety and care for patients 3. Improved analytics and data management 4. Improved cost effectiveness and financial efficiency 5. Improved cooperation and communication between medical professionals In order to guarantee the effective deployment and use of an HMS, hospitals ought to: 1. Perform comprehensive feasibility studies and needs assessments. 2. Pick an appropriate HMS provider and solution. 3. Offer thorough instruction and assistance to end users. 4. Regularly check and assess the system's performance 5. Update and improve the system to accommodate changing requirements and technological advancements. Healthcare companies may enhance the overall healthcare experience, lower costs, and increase patient care quality, safety, and efficiency by utilizing the potential of a well-designed and executed hospital management system.

Future Scope: The future scope of the healthcare system using Java is promising, as Java is widely used for developing secure, scalable, and high-performance healthcare applications. Here are some key areas where Java will continue to play a crucial role in the healthcare industry:

1. Electronic Health Records (EHR) and Hospital Management Systems: Java-based platforms can efficiently handle patient records, billing, and appointments. Integration with cloud services enhances data storage and accessibility. Compliance with healthcare standards like HL7, HIPAA, and FHIR can be implemented using Java frameworks.
2. Telemedicine and Remote Patient Monitoring: Java is used in developing telehealth platforms for video consultations and remote diagnostics. IoT-enabled Java applications can monitor patient vitals and send real-time data to doctors.
3. AI and Machine Learning in Healthcare: Java frameworks like Deeplearning4j and Weka can be used for predictive analytics, disease detection, and drug discovery. AI-powered Java applications can help in medical image analysis, automated diagnosis, and chatbot-based assistance.
4. Cloud-Based Healthcare Solutions: Java is widely used in developing cloud-based healthcare applications that store, process, and analyze large amounts of patient data securely. Integration with platforms like AWS, Google Cloud, and Azure is seamless with Java.
5. Healthcare Mobile Applications: Java, along with Spring Boot and Android, can be used to build mobile health apps for patients and doctors. Java-based mobile applications help in medication reminders, fitness tracking, and mental health support.
6. Blockchain in Healthcare: Java-based blockchain applications can secure patient data, prevent fraud, and ensure interoperability between different healthcare providers. Decentralized Java applications can improve medical record sharing securely.
7. Medical Research and Bioinformatics Java is used in genomic research, drug trials, and computational biology applications. Frameworks like Apache Hadoop and Spark with Java help in processing big data in medical research.
8. Cybersecurity in Healthcare Java provides secure encryption, authentication, and access control for healthcare applications. Java-based security frameworks like Spring Security ensure data privacy in hospitals and clinics.

V. REFERENCES

[1] Herbert Scheldt, Java Complete Reference, Fifth Edition, Tata McGraw Hill Edition.
[2] Phil Hanna, JSP 2.0: The Complete Reference, Tata McGraw Hill Edition, 2003.
[3] Elmarsi and Navathe, Fundamentals of Database System (Third Edition), Addision Wesley.
[4] Ian Somerville, Software Engineering, Third Edition, Pearson Education.
[5] Ali Bahrami, Object-Oriented System Development, Third Edition, Tata McGraw Hill Edition.
[6] Ivan Bayross, SQL, PL/SQL programming language of Oracle, Second Edition, BPB Publication.
[7] Franklin. Harrison's Principles of Internal Medicine Companion Handbook, 14th Ed. (Electronic Book Card with Player). New York, NY: Franklin Electronic Publishers; Oct 15, 1998.
[8] Beers MH, Berkow R, editors. The Merck Manual of Diagnosis and Therapy: 17th Edition. Whitehouse Station, NJ: John Wiley & Sons; 1999.
[9] Kilsdonk AC, Frandji B, Van Der Werff A. The NUCLEUS integrated electronic patient dossier breakthrough and concepts of an open solution. Int J Biomed Comput 1996 Jul;42(1-2):79-89. [Medline] [CrossRef]
[10] GEHR Consortium. Deliverable 19: GEHR Architecture.
1995. URL: http://www.chime.ucl.ac.uk/HealthI/EUCEN/del19.pdf [accessed 2001 Sep 3]
[11] Grimson W, Berry D, Stephens G, Felton E, Given P, O'moore R. Federated healthcare record server--the Synapses paradigm. Int J Med Inform 1998 Dec;52(1-3):3-27. [Medline] [CrossRef]
[12] KONA Editorial Group. HL7 Document. Patient Record Architecture. Framework Version 1.0, Draft. 2000. URL: http://www.hl7.org/special/committees/sgml/PRA/PRA_ballot_April00.zip [accessed 2001 Sep 3]
[13] Griffith SM, Kalra D, Lloyd DS, Ingram D. A portable communicative architecture for electronic healthcare records: the Good European Healthcare Record project (Aim project A2014). Medinfo 1995;8 Pt 1(8 Pt 1):223-226. [Medline]
[14] Kalra D. Electronic health records: the European scene. BMJ 1994 Nov 19;309(6965):1358-1361. [Medline]
[15] CEN TC251/WG1. Health Informatics: Electronic Healthcare Record Communication - Part 1: Extended Architecture.
[16] Electronic Healthcare Record Support Action Consortium.. Deliverable 3.4: Review of CEN ENV 13606 EHCR Extended Architecture. 2000.
URL: http://www.chime.ucl.ac.uk/HealthI/EHCR-SupA/del3-4v1doc.zip [accessed 2001 Sep 3]
[17] The Good European Healthcare Record.. Frequently Asked
Questions.
URL: http://www.gehr.org/introduction/faq-1.html [accessed 2001 Sep 25] , Third Edition, Pearson Education.