

## Accepted Manuscript

*Design and implementation of a smart card based healthcare information system*

Geylani Kardas, E. Turhan Tunali

DOI: [10.1016/j.cmpb.2005.10.006](https://doi.org/10.1016/j.cmpb.2005.10.006)

To appear in: *Computer Methods and Programs in Biomedicine*

Accepted: 18 October 2005

Please cite this article as: Geylani Kardas, E. Turhan Tunali, Design and implementation of a smart card based healthcare information system, *Computer Methods and Programs in Biomedicine*, doi: [10.1016/j.cmpb.2005.10.006](https://doi.org/10.1016/j.cmpb.2005.10.006).

This is a PDF file of an unedited manuscript that has been accepted for publication. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.



# Design and Implementation of a Smart Card Based Healthcare Information System

Geylani KARDAS and E. Turhan TUNALI\*

International Computer Institute,

Ege University, Bornova, 35100, Izmir, Turkey

**Abstract:** Smart cards are used in information technologies as portable integrated devices with data storage and data processing capabilities. As in other fields, smart card use in health systems became popular due to their increased capacity and performance. Their efficient use with easy and fast data access facilities leads to implementation particularly widespread in security systems.

In this paper, a smart card based healthcare information system is developed. The system uses smart card for personal identification and transfer of health data and provides data communication via a distributed protocol which is particularly developed for this study. Two smart card software modules are implemented that run on patient and healthcare professional smart cards respectively. In addition to personal information, general health information about the patient is also loaded to patient smart card. Health care providers use their own smart cards to be authenticated on the system and to access data on patient cards. Encryption keys and digital signature keys stored on smart cards of the system are used for secure and authenticated data communication between clients and database servers over distributed object protocol. System is developed on Java platform by using object oriented architecture and design patterns.

**Keywords:** Smart card, healthcare information system, distributed object protocol

---

\* corresponding author: Ph: +90-232-388-7228 E-Mail: tunali@ube.ege.edu.tr

## 1. INTRODUCTION

Automation systems in hospitals and medical centers serve the purpose of providing an efficient working environment for healthcare professionals. Access to accurate health data quickly is one of the main functions of these systems. To fulfill these requirements, these systems should contain an information network that acquires processes and stores patient information.

There can be many sources that the information related to the patients can be obtained from: the patient himself, results of tests applied to patients, online patient monitoring systems, doctors' diagnoses for patient illnesses and previously stored patient information [1]. In addition to the symptoms of a suspected illness, it's important for the doctor to be able to retrieve patient's previous information during his examination.

The usual way of obtaining relevant patient data is to connect to the hospital database. In some cases, simultaneous database accesses from different terminals located in examination rooms can cause performance problems due to high data rate. In other cases patient data can be needed in an environment without network connection facility (in an ambulance for example) or another hospital where patient is not registered. An information network can be established so that patient data can be shared by different hospitals but mostly, factors like different IT and network solutions of hospitals, data security and network infrastructure make realization of such network difficult.

These problems can be solved by increasing the capabilities of hospital automation systems by using intelligent storage and retrieval mechanisms. Portable media can play a key role in sharing limited amount of patient specific information, which in turn, may provide important data to a hospital automation system [2]. The patient can carry the media with him/her anywhere and any time and present it to the doctor at the time of consultation.

The media selected for the above purposes should be cheap, easy to use, carry and update with new information and shouldn't get damaged easily. "Smart card" appears as the most suitable medium to be used in healthcare information systems when such requirements are considered.

Smart cards can be described as portable integrated devices that store and process data. These tiny computers with their own memories and processors have a widespread usage especially in telecommunication and mass-transit systems [3].

Speed, security and portability properties make smart cards a potential tool in healthcare systems. Many countries implement or continue to develop such systems including smart card components. An intensive study on these systems is seen in working projects such as Sesam Vitale in France and DENTcard in Germany [2]. Also European Commission supported France – Belgium project Transcards [4], EU IV. Framework R&D project Netlink [5] and Netcards project which aims to set up a wide trans-European network of health services for mobile citizens [6, 7] can be considered as important studies in this area. Furthermore the Health Insurance Card project of

Slovenia can be accepted as the most recent and complete operation in health card systems during development of the system introduced in this study. The system is the first healthcare system which provides nation-wide use of smart cards in health sector [8 and 9]. The United States has been slow to adopt smart card technology since businesses have already invested heavily in the magnetic stripe technology used for credit cards [10]. However like other sectors, smart card usage in healthcare is beginning to become popular in this country and some state-based solutions are already implemented [11].

In this study, a novel smart card based healthcare system is developed. The system includes smart card usage in data flow and provides data communication via a distributed protocol that is also developed within the scope of this study. Smart cards are used both for mobile data transmission and security and authentication purposes in the system. Unlike most of the other studies, in addition to system management data, smart cards of our system contain personal identification and health data. This feature provides availability of limited but important health information at presence of the card as well as database access if possible.

The paper is organized as follows. Section two gives general specifications of the system. It is followed by the system description and software design in the third section and implementation in the fourth section. Section five includes a status report of the system. Lessons learned and future plans are presented in the sixth section.

## 2. GENERAL SPECIFICATIONS

The system developed is called SCHS (Smart Card Healthcare System). Both patients and healthcare professionals have smart cards in SCHS. Doctors use their cards to be authenticated in the system while patient cards include owner's general health information which can be accessed without any database connection.

There exists a central hospital database (which can be in a distributed structure) to store users' on-card and off-card health data. Department / clinic specific patient data are stored on local databases of related departments and the system also interacts with them over a distributed object protocol.

### 2.1 User Requirements

During domain analysis phase of the SCHS development, user requirements are determined by following an extensive investigation of clinical requirements in Ege University Medical School & Hospital. Ongoing healthcare process and health information system in the hospital are examined in detail and subsequent incorporation of those requirements into the SCHS's functional specification has been realized.

We had to take into consideration only the local requirements of the hospital due to current non-standardized healthcare system in Turkey. Health information systems in use vary especially in structure of health document and clinical process of those documents. On the other hand we also surveyed on European CEN Technical

Committee 251 (CEN/TC 251) [12] active work items in health informatics. We mostly concentrated on the first working group studying on information models because work item, called 102 (PATCARDS) of this group, involves smartcard integration on healthcare systems. Unfortunately, we had a restricted access to standards documents about those related items and hence we couldn't gain details of the data structure proposed for patient health cards. However we took into account PATCARDS's data organization (grouping of data about identification, administrative, electronic prescription, etc.) and reappraised it according to our system needs during design of our smartcards.

## **2.2 Security and Authentication Properties**

In each system component design, security is considered as an indispensable feature. A secure channel is established between terminals in examination rooms and card acceptance devices (CAD) which are connected to those terminals. When a doctor or a patient smart card is inserted in CAD, authentication is assured between card and host computer software by key exchange. Then card access PIN is requested from card owner and entered PIN is checked by smart card itself. Following proper PIN entrance, smart card session is opened.

Database accesses from client terminals are also realized via a developed secure distributed system protocol. Data is transmitted in an encrypted form over protocol's channel.

To provide doctors' authentication for system and access to system servers, private digital signatures stored on doctors' smart cards are used. Detailed discussion about security issues is given in subsection 3.6.

### **2.3 Data Stored on System Smart Cards**

During design of health data model, we had to determine the amount and specification of the health data that would be stored on system smart cards. In smart card integrated healthcare systems, we should take into account of the following alternatives: Should we put health data on smart card or should we use smart cards just only as a media to access health data already stored on a network in a secure way? We evaluated both alternatives and introduced a hybrid approach in which smart cards are used in the system both as a mobile health data carrier and a security key to access private hospital network. Keeping all health records in a smart card is currently impossible due to space limitation on smart cards. Even if it were possible, it would causes lack of system management in case of lost or damaged user cards.

On the other hand, considering advanced capabilities of the state of the art smart card technology, use of smart card only as a security key to access health data on a network would waist such capabilities. Maybe the most important disadvantage of such an approach would be seen in medical environments with no network access. Examination rooms may have no Intranet connection but essential health records of a patient may be urgently needed. According to the data model design we have developed on the smart card, those urgent records can be stored on a patient's smart card and the healthcare



professional can quickly access related records stored on the card. Let us consider an emergency situation in which a patient is carried on an ambulance. Currently most of those vehicles have no Internet / Intranet access. Hence, paramedics can quickly access any needed urgent health data of the patient by using the patient's smart card. We believe that our proposed approach about smart card data model will meet all above needs and smart cards on a healthcare system should have both authentication & data security capabilities and secure mobile data storage roles.

As it is mentioned before, two types of smart cards exist in system: patient card and doctor card. In each patient card, owner's personal information is stored. Besides unique patient ID and card access PIN, patient's name, surname, birth date, blood type, gender, address, home, work and mobile telephone numbers are stored in patient card as personal information. Emergency contact information (name, surname, home, work and mobile phone numbers of the person to be contacted and his/her relationship with patient) and insurance information (patient's insurance company's name and relevant SSNs) are also stored in the card.

Both patient personal information and emergency contact information are not PIN protected. Especially, in an emergency condition, it may not be possible to obtain PIN from patient. In such conditions, card provides personal data and contact information without any PIN entry. However all the other data on card is PIN protected and card can block itself against repetitive wrong PIN entries.

Patient health information stored in the card can be grouped as follows: chronic and/or important former diseases with diagnosis dates, permanently used medications with doses, allergies with diagnosis dates, immunizations with their dates, surgical operations including operation date, clinic name and summary information. Additional data is stored as a memo on card.

Patient's last examination and prescription information are also stored on card. Last examination information includes last examination date, clinic and doctor data (doctor ID, name and surname) and summary of examination. Prescription information includes prescription's date, clinic, medicine list, state approval information and again related doctor's data (doctor ID, name and surname).

Card issue and last update dates, network address of hospital database server on which related patient's records are stored, and patient DES (Digital Encryption Standard) key are located in card as system information. Storing database network address on patient smart cards supports distributed hospital database system such that card terminals can dynamically determine patient database for communication purpose. Patient DES keys are used for encryption/decryption purposes during data transmission over network.

All data on patient card is read-only to doctors except last examination and prescription data. After every examination, proper inspection and prescription (if needed) data are written to card by doctors.

On the other hand, doctor smart cards only store doctor personal information and system data. Doctor's unique ID, card PIN, name, surname, hospital department, address, home, work and mobile phone numbers are stored as personal information. Card issue and last update dates, again network address of relevant hospital database server and doctor digital signature are stored as system data. Doctor digital signature is a DSA (Digital Signature Algorithm) private key which provides doctor authentication on system.

#### **2.4 Smart Card Sessions**

Computers located in examination rooms are defined as system client terminals. Each terminal has a connected CAD and they can connect to specific system servers to access databases. The software running on terminals can open doctor and patient sessions. There must be an open doctor session to accept patients and carry out examinations. Doctor session can only be opened by a doctor smart card. When a doctor card is inserted in CAD, a secure channel is established between host application and smart card. After mutual authentication, card PIN is requested. User's PIN entry is sent to smart card and card processes the PIN. If PIN is valid, card session will be successfully opened and terminal application communicates with the remote server over system protocol to get messages for related doctor. Remote database server address is obtained from smart card. Doctor personal data including DSA private key is temporarily transmitted to application from smart card. This data will be available during doctor session. Now doctor session is open and host application waits for patient smart cards.

One approach is to keep doctor session open only when doctor card is inserted on CAD, which increases security [13]. So, a patient card can only be accepted when a doctor card is also present in reader. However this can only be possible in CADs with two slots and advanced mutual card authentication capabilities. We have used a single-slotted CAD with a simpler structure. In our system, opening a doctor session before patient session is sufficient for operation.

Doctor can close his/her session using the application interface or session will be automatically closed when application ends. Figure 1 represents the doctor session.

Figure 1 approximately here

When an open doctor session condition is satisfied, application can accept patient smart cards and open patient sessions. Like doctor session, a secure channel is established and mutual authentication is realized when a patient card is inserted on CAD. Entered PIN is validated and remote messages for patient are received in same manner.

To obtain patient's clinic specific health records, an extra communication channel with clinic server exists. Using system private distributed protocol, terminal receives requested data from clinic server in an encrypted form. Data is encrypted with patient DES key and to decrypt it on client side, terminal needs same DES key stored on patient smart card. In addition to general health information, decrypted clinic specific patient health data are displayed. After examination, doctor updates inspection and prescription information on patient smart card with new data and he/she also updates clinic specific

patient data if he/she has authority. Updated clinic specific data are again encrypted using patient encryption key, signed with doctor's private digital signature and sent to remote clinic server. Reverse procedure will be carried out on the server side: Signature is verified with doctor's public key and patient data are decrypted again with patient DES key. Notice that the required DES and DSA keys are obtained from hospital central database on server side using database network addresses sent from client terminal. Data update on clinic database is completed if everything is in order. Result of remote process is returned to terminal and patient session is closed. In every step of patient session, patient's smart card should be located on CAD; otherwise session and all session related interfaces are automatically closed for security considerations. Figure 2 demonstrates patient session.

Figure 2 approximately here

After examination, patient will apply to system administration unit to record new inspection and prescription data stored on smart card to hospital database and to realize prescription approval.

## **2.5 System Administration**

A hospital unit is proposed to carry out system administration functions. Basically that unit is responsible to manage hospital database. Other responsibilities of this unit are to record new patients and doctors to system and to perform smart card related operations

like smart card preparation for new users, smart card cancellation, and data update from smart cards to database and database to smart cards.

### **3. SYSTEM DESCRIPTION**

In this section, overall system architecture and major software / hardware components of SCHS are discussed.

#### **3.1 System Architecture**

Based on 3-tier software architecture, the system consists of client, server and database layers. In an ordinary client/server application, clients connect directly to the database server and perform queries. However in such a system, client software components will completely depend on database server and related database software. To overcome the dependency problem, instead of the clients directly being involved in database related processes, an extra control structure is added to manage queries. This lets each component to be designed independent of the others and provides software reusability.

The system has an object oriented software in which MVC (Model, View & Controller) software architecture design pattern is applied. In MVC pattern, view components communicate with system data model by means of a controller mechanism. So, client applications only include view components (user interfaces such as forms, dialogs, etc.) and those applications are completely independent of relational database which stores system data model [14, 15].

By use of MVC pattern in the system, client applications on examination computers are database independent and they only communicate with controller remote objects over special system protocol. The special protocol in question is an implementation of Java RMI (Remote Method Invocation) technology over TCP/IP. Java RMI technology lets distributed remote objects to communicate with each other without depending on network infrastructure. System clients use remote objects' interfaces over RMI channel and realize database operations via those objects. For example, when a data update on local clinic database is needed, client application obtains data from user interface and calls remote object's related method with those data as parameters. Authentication of client for that operation and data update are performed by that remote object.

Implementation of MVC pattern and RMI also provides digital signature and data encryption key usage in system authentication and data security respectively. One of the important aims of the system is to access clinic databases in a secure and authenticated way in which smart cards play an active role. Data is transmitted as encrypted and signed on protocol. RMI has already facilities like object serialization and parameter marshalling on its channel so encapsulated data are not purely transferred. In addition to that mechanism, parameters of remote object's method are encrypted and signed. Figure 3 shows whole system architecture with components.

Figure 3 approximately here

## 3.2 Smart Card Software

### 3.2.1 Patient smart card

The software package running inside a patient's smart card is called `tr.edu.ege.ube.schs.patientCard`. This package contains both the Java card applet and model classes to hold patient data discussed in section 2.3. This software manages 4058 bytes of patient data. The UML diagram of card software's object model is shown in Figure 4.

Figure 4 approximately here

"PatientApplet" extends `javacard.framework.Applet` class and it communicates with off-card smart card applications to serve patient data in a secure way. It aggregates patient health data, system management data (like DES key) and card owner's PIN in instance of objects called "Patient", "SystemData" and "OwnerPIN" respectively. The "Patient" object encapsulates patient's health data both in collections of objects (like Immunization and Allergy) and sole instances of objects (like InsuranceInfo and EmergencyContactInfo).

On the other hand, system dependent management information is encapsulated by an instance of the object called "SystemData". It has attributes to store system specific data like patient's DES key, card last update and address (IP) of the hospital database which stores all data about the related patient.



### **3.2.2 Doctor smart card**

Following the design of patient smart card software, design and development of the software for doctor smart cards have been straightforward. It is called `tr.edu.ege.ube.schs.doctorCard` and it manages 1333 bytes of doctor data. The object called “DoctorApplet” is an extended JavaCard applet that serves doctor data to host applications. It aggregates the objects called “Doctor”, “OwnerPIN” and “SystemData” to store above mentioned doctor smart card data. “OwnerPIN” object stores card PIN and its validity properties in the same manner with the patient card. “Doctor” object encapsulates information like doctor’s name, surname, address and department while “SystemData” stores doctors DSA private key in a serialized form in addition to the other system management data.

### **3.3 Card Terminals**

Computers located in doctor’s examination rooms, with a CAD connected are defined as card terminals and they occupy the client layer of the system. It is sufficient to have a Java Virtual Machine (JVM) and Open Card Framework (OCF) library in addition to healthcare system software on each terminal.

OCF is a smart card middleware developed by Open Card Consortium including IBM and other leader card manufacturers. The framework is implemented in Java and provides a portable, card manufacturer independent smart card application development

environment. In architecture, OCF is located between CAD and host application running on PC [16]. OCF is expected to be integrated in most of the smart card based healthcare solutions [8].

Smart card client package running on client computer carries out both doctor and patient smart card communication. It prepares card environment on PC, selects related card application and communicates with cards via APDU (Application Protocol Data Unit) packages of ISO 7816 standard smart card communication protocol [17]. This package is designed as an API which aims at abstracting card clients including user interfaces from APDU communication. Thus, user interface components can use prepared objects provided by this package especially in APDU byte vector formation without struggling with limited smart card data structure conversions. Hence this client software bridges the gap between graphical user interface components and on-card applications as it could be seen in data flows (1 and 2) given in Figure 3. Object model of this client software is shown in Figure 5.

Figure 5 approximately here

Object called “CardManager” is responsible for smart card environment initialization and shut down on PC, secure communication channel establishment between smart cards and card session management. Data about card sessions are handled by the objects called “DoctorSession” and “PatientSession”. User interface components access card data over those objects without striving for APDU communication. It should be noted that domain objects of the package (like “PatientCoreData”, “DoctorCoreData”,

“InsuranceInfo”, etc.) are high level counterparts of the on-card system software. For example “PatientCoreData” stores the patient’s name in a well-known Java object (String) instead of a hex-coded byte array. Therefore it could be processed in interface components in a very simple way.

On the other hand, smart card terminals act as clients in RMI protocol calling remote object’s methods. As it is mentioned before, client computers do not contain any software component which is responsible to access databases and perform queries. Remote objects fulfill those operations instead of client software. Client software only contains user interface components (instances of Java Foundation Classes) and forms view layer of MVC architecture.

### **3.4 RMI Servers**

Each hospital clinic or department has its own RMI server which provides local clinic database and hospital central database access to connected clients. Card terminals in a department take service from RMI servers located in the same department. Remote objects, whose methods are called by clients, exist on those servers.

RMI server application creates objects with remote interfaces and registers them to RMI registry on server. When a client object wants to use a remote method, it first connects to the RMI registry on server and obtains interfaces of relevant remote objects. Then they can call remote methods by using those interfaces according to RMI protocol [18].

Remote objects in the system are responsible from encrypted data transmission, authentication control and remote database connection by using JDBC (Java Database Connectivity) technology. For example, when a client wants to update a patient's information in the clinic database, it calls proper remote object's method with data in an encrypted and signed form. Data is sent to stub on RMI server over RMI channel on TCP/IP with parameter marshalling. Stub prepares data and sends it to remote object. Object controls authentication of client, decrypts data, prepares proper query and updates data on remote database. Operation result or exception (if occurred) is returned to client over the same channel. So, all control and database connection operations are abstract to clients. Remote objects on RMI servers fulfill those operations on behalf of clients and form controller layer of MVC architecture of the system.

### 3.5 System Databases

The system supports a hierarchical data structure in which patient general health information (including data on smart cards) is stored in a hospital central database while clinic specific information in clinic databases. Relational model of the central database is given in Figure 6.

Figure 6 approximately here

As their names depict, the tables called "PATIENTS" and "DOCTORS" store each system user's personal data with his/her unique system id. Each patient record stores the patient's core data (name, surname, birth date, blood type, etc.), smart card data (card

access PIN and last update), insurance, emergency contact, system data and DES key. On the other hand, a doctor record stores the doctor's core data (name, surname, department, etc.), DSA public and private keys, system data and smart card data.

Allergies, surgical operations, immunizations, medications and former diseases are stored in corresponding database tables with their unique IDs. IDs have been generated automatically during system implementation except in medications. Barcode number on each medication is given to the related medication as its system ID.

Each patient's health records are stored in appropriate enrollment tables. Those tables have names started with the prefix "PATIENT\_". For example a record in the table - called "PATIENT\_DISEASE"- stores a patient's former disease with its first diagnosis date. This record is associated with the related patient and the disease via their unique IDs. It should be noted that the "PATIENT" table has a "one-to-N" relation with each of those tables.

As it is mentioned before, clinic based patient data are stored on separate clinic databases to provide both flexibility and modularity of the system. Due to non-standard form of the information systems and database structures used in clinics of many hospitals as in the Ege University Hospital, our design seems most appropriate. For example, the database structure and type of the data stored on neurosurgery clinic's database are different from the ones used in pediatrics. Our proposed layered system approach brings a solution to integrate databases used by those clinics into the SCHS such that it is enough to deploy an RMI server in front of each clinic's local database

and prepare the corresponding wrapper software (including its model objects and server remote object). So, smart card terminal applications communicate with those remote objects and doctors can access the local patient data during examination as described in above sections.

Databases store attributes of system data model's objects. Model is completely independent from use and view of data. So, databases form the model layer of the MVC architecture. Controller components map proper bean objects to tables of those databases.

### **3.6 System Security Architecture**

Like every distributed healthcare system, security of health data is vital in SCHS. Hence, transmission of data is realized over secure protocols both in smart card and remote database communications. These secure communications are discussed here in detail.

#### ***3.6.1 Communication between smart card and host PC***

The first step in any communication between an off-card entity (typically clinic applications residing on doctor workstations) and a smart card is to provide *mutual authentication*. This involves comparing keys, which are stored in a key file. The key file contains the value of a “mother key”. By default, the value of this key in the key file

is filled with zeros. The real value of the mother key is provided by the card manufacturer with the card.

Every smart card session begins with authentication stage in which both entities perform the check: the card checks that the keys in the key file match its own and the clinic software checks that the keys on the card match its own keys stored in the key file. After successful authentication, a secure communication channel is established between the two entities. At this time, a smart card owner should enter his/her card access PIN. Given PIN is checked by smart card itself and successful entry opens the card session. In case of three consecutive wrong entries, smart card blocks itself for any communication. In such a case, card user should apply to system management unit of the hospital to unblock his/her card.

### ***3.6.2 Communication between smart card terminals and database servers***

As mentioned before, doctors may want to access a patient's health data stored on remote hospital databases. This requires a network communication over a secure protocol. In SCHS, remote communication is realized via the Java RMI protocol which works on top of the TCP/IP. In RMI, incoming and outgoing data are transferred in a serialized form meaning that data encapsulated by objects are not purely transferred over RMI channel. Receiver of the data should know the type of the object to deserialize content for if otherwise, it only gets a meaningless data stream. From this point of view, SCHS specific communication protocol provides a degree of security. However, it couldn't be said that RMI meets all security requirements during network

communication and there should exist an inner crypto protocol to provide security indeed.

In Figure 7, sequence diagram of the communication between card terminals (doctors' PCs) and database servers is given. SCHS presents a communication protocol in which both data security and digital identity verification are duly provided.

Figure 7 approximately here

Let us consider that after an examination, a doctor needs to update patient's health record on the system database. Precondition of such an update procedure is that both doctor and patient smart card sessions are opened on doctor's computer that can access remote database object over RMI protocol.

Data to be updated is first encrypted, then signed and finally serialized before network transmission. Encryption process uses related patient's DES key which is retrieved from patient's smart card. Encrypted data is signed by the doctor's DSA private key that is obtained from doctor's smart card. Both signed and encrypted data is encapsulated by an object and this object is serialized and sent over the RMI channel to remote side. Figure 8 gives flowchart of this protocol.

Figure 8 approximately here



At the remote side, received message is first deserialized and data which is both encrypted and signed is obtained. Using related doctor's DSA public key, remote component authenticates the doctor. Encrypted data is then decrypted with related patient's DES key and updated in the database (Figure 9). All database communications are logged in system servers.

Figure 9 approximately here

On the other hand, key management should be taken into consideration. Generation, cancellation and replacement of security and authentication keys are all managed by system administration unit within SCHS. During card preparation for a newly recorded system user that may be a patient or doctor, related cryptographic and/or signature keys are generated. In the case of registering a new patient, the key generator of the system produces new patient's DES key that will be used in appropriate system security operations. Generated key is both stored in database and patient's smart card. The key is stored on card in a serialized form. Essentially this is a byte stream and this stream is again transformed into the real DES key on authorized host applications during their processes.

On the other hand, system administration unit prepares DSA key pairs for healthcare professionals so that they can work within the SCHS. A Secure Random key generator is used and a DSA key pair is produced during registration of doctors. Private key of this pair is stored in both system database and doctor's smart card. Public key is stored

only in the database. To obtain key generators and random seeds, Java API for Crypto Extension is used.

Of course DES is potentially vulnerable to a brute-force attack and it may need to be replaced with an algorithm containing multiple encryptions with multiple keys such as 3DES. In fact, due to modular architecture of the host applications and proper API calls, any encryption/decryption algorithm with symmetric cipher (such as 2DES, 3DES or Blowfish) may easily be applied into the security protocol of the SCHS. Considering 3DES (with 3 keys), we need to generate an effective key with 168 bits length for each patient and store related key in patient's smart card. Above mentioned crypto protocol of the SCHS will remain same except encryption/decryption processes will need to be modified for 3DES.

#### **4. IMPLEMENTATION**

Starting from smart card software, all the way up to server components, the whole system is developed in Java platform.

We have used GemXpresso 211/PK model Java cards with multiapplet support, manufactured by Gemplus. They have 32 K ROM, 32 K EEPROM and 2 K RAM. However smart card programmers can only use 16.5 K of EEPROM and 0.5 K of RAM for their applications because card operation system and JCRE (Java Card Runtime Environment) reserves remaining sources for their own usage. Approximately 8 K EEPROM is used in doctor cards for the developed doctor smart card software and

permanent data objects. Patient card software needs more space (approximately 14K) both for software and permanent data objects. Note that any smartcard that complies with JavarCard specifications and provide sufficient data space can be used in implementation of our design. That is, our design is platform independent and by using JavaCard framework [3], our system can be ported to different platforms without modifying any software.

Gemplus GCR410 serial card read/write unit is used as CAD in system. It is connected to a terminal PC with 9600 baud data transmission rate. Card client, RMI client and card terminal / user interface packages are deployed to a PC with Intel PIII 650 MHz CPU running MS Windows 2000 operating system. It has Java 1.3.1 runtime environment and OCF 1.2 library for smart card communication. That PC represents a card terminal located in a doctor's room. System administration software is also deployed and tested on this computer.

Server components are deployed to a PC with Intel P4 1.4 GHz CPU running MS Windows XP. It does not contain any smart card software. Only a Java runtime environment is needed to run RMI server software and JRE 1.4 is used for this purpose. MS SQL Server 2000 is used both for hospital database and clinic database implementations.

Over 30000 lines of code are written to develop the whole system that includes seven software modules.

Some performance measurements are also obtained during system tests, containing elapsed time measurement during data transmission between smart card client applications and smart cards. Considering data bus with 9600 baud, to send a command APDU and receive a response APDU with 255 bytes of data and display content of the APDU in related interface take approximately 1.5 seconds. To write 255 bytes of data to smart card and receive response from card takes approximately 2 seconds. Furthermore, it takes approximately 9 seconds to start a user session and display PIN entry dialog after insertion of smart card into CAD.

To provide some flavor of the developed environment, the clinic program developed for Neurosurgery Department of Ege University Hospital is discussed with its selected screenshots. Note that the persons, their names and all other personal and/or health data given in the following are all fictitious and for demonstration purpose only. Figure 10 is the screenshot taken in runtime of the SCHS clinic application in which a doctor has recently opened a doctor session using her personal smart card. The application has communicated with the smart card and temporarily transferred doctor's data (including its DSA private key) over the secure channel after her authentication. Depending on the access information on the smart card, application has also communicated with the central hospital database to retrieve personal message for the doctor if any exists. Now the doctor is ready to accept patients for examination.

Figure 10 approximately here

Screenshot in Figure 11 is taken just after the doctor has accepted a patient and opened a patient session on the clinic application. All displayed data (except the remote database message) is received from the patient's smart card. Like in doctor session, when a patient smart card is opened on a clinics application, the application immediately communicates with the central database to retrieve any existing message for the patient. Database address is again gained from the patient's smart card. In Figure 11, patient's medical information about his allergies, diseases, etc. are displayed. The doctor can also access other patient information (e.g. surgical operations, last inspection and prescription) stored on patient's smart card by using the other submenus of the "Patient" menu.

Figure 11 approximately here

Last screenshot (Figure 12) portrays secure access of the application to the neurosurgery department's local database. The doctor has received her patient's neurosurgery records according to the distributed object protocol proposed in this paper. Her record request is validated at the RMI server of the neurosurgery department and patient information is transferred to the clinic application in an encrypted form. Encrypted data can only be decrypted by using the key stored in the patient's smart card. When the doctor demands an update of the patient's neurosurgery records, it is enough for her to simply press the "Update" button and approve the operation. In this condition, all patient data is decrypted with patient's key and signed by the doctor's DSA private key and sent over from the protocol's secure channel in a serialized form. Received information is first

validated with the doctor's public DSA key and then decrypted and updated into the department's database.

Figure 12 approximately here

## 5. STATUS REPORT

During implementation, test and evaluation of SCHS, the environment of the Ege University Hospital's Neurosurgery Department is used as the pilot domain. Hence, SCHS is currently ready to be used in Neurosurgery Department and is expected to be fully operational in near future, upon fulfillment of some system deployment issues. However, contribution of other departments into the system will be fast and easy, owing to above mentioned modular software design of the SCHS.

SCHS can be considered as a powerful healthcare automation with integration of smart card use into existing hospital information systems. Its distributed protocol enables mobile and secure access to the patient records and facilitates roles of both healthcare professionals and patients.

Similar studies introduced in Section 1 also aim to provide system enhancements via smart card use. However, contribution of smart cards in those studies is limited even in the systems that are currently in use. For example system in [11] has a restricted design in which smart cards only behave as a portable health report card. Potential security and authorization features are not fully presented. Working system introduced in [8] has an

involved architecture but use of smart cards in clinic computers is so limited due to computer's stand alone design and lack of network connection.

On the other hand, protocol introduced with SCHS allows use of cards in both data storage and security in addition to mobile data carriage. Designed card environment of SCHS is enriched with the state of the art card technologies (JavaCard, OCF, etc.) and this helps to design and develop system smart cards with higher capacity and security features.

It should also be noted that architecture of SCHS takes care of the easy integration of the currently working health information systems in a hospital with the help of its layered approach. Hence, users of the old information system (both health professionals and patients) can adapt to the new system easily and quickly. For example clinic module shown in Figure 12 has exactly the same GUI and usage procedure with the previous system used in Neurosurgery Department of Ege University Hospital.

A collaborative study has been performed with healthcare professionals employed in Neurosurgery Department of Ege University Hospital during requirement determination, domain analysis and evaluation phases of the SCHS. It is essential that doctors have knowledge of evaluation issues in order that they can assess the strengths and weaknesses of evaluation studies and thus interpret their results meaningfully [19]. Contribution to the design and implementation of such studies provide system developers with useful information. Our colleagues preferred a practical assessment based on their experience instead of a methodological one in this study. As it is

discussed in [20], we also experienced difficulty in deploying a new healthcare system for an environment in which some users are addicted to legacy system. However, we didn't meet any strong resistance against the new system because doctors denoted that they found a new application on their desktop with almost same GUI and working mechanism with the legacy one. Hence SCHS seems to be widely used in the department after a relatively short learning period when it becomes fully deployed.

Weaknesses of the SCHS should also be considered. SCHS is currently not fully operational and therefore it has met a limited number of user requirements. Controller layer of its software architecture should be strengthened with up to date technologies those are mentioned in Section 6. So, concurrent database accesses (especially for central databases) will be optimized with already implemented connection pool mechanism to enhance performance. Finally, it needs to be studied on the current electronic prescription data structure to make it fully compatible with working pharmacy softwares in Turkey.

## **6. LESSONS LEARNED AND FUTURE PLANS**

A healthcare automation system based on smart cards is designed and developed. In addition to carrying mobile information, system smart cards are used in security and authentication processes.

The most important problem encountered during system development is the lack of medical data store and retrieval standardization in healthcare sector. Existence of a



worldwide-generic coding standard for healthcare data will surely ease the design and development of smart card based healthcare systems. Our system has its own specific medical data coding in databases. However as standardization occurs, the database can be redesigned to meet those standards.

The capacity increase and cheaper costs will improve quality of smart card services. For example, the present patient cards of the SCHS can only store last inspection and prescription data as indicated before. One of our future plans is to integrate smart cards with higher capacity into the system to provide storage of more than one inspection and prescription data on card and simplify the doctor's examination process. With use of such high capacity smart cards, we also intend to store extra medical information like x-ray films and test documents on smart cards.

Another further system development that we take into account is the integration of pharmacies to system and processing of electronic prescriptions currently stored in smart cards. Such integration provides a paperless environment for prescription protocols between hospitals and pharmacies.

We also intend to improve the distributed object protocol of the system by the use of advanced RMI facilities and Enterprise Java Bean architecture in J2EE platform especially for an easier and more efficient process management. We believe that those modifications can be easily done by means of layered system architecture of SCHS.

**REFERENCES**

- [1] T. Tunali, S. Yildirim and T. Dalbasti, The use of smart cards in health care, Hermes Project Workshop (2002), pp. 1-6.
- [2] C. Pagetti, C. Mazini, M. Pierantoni, G. Gualandi and H. Schepel, A European Health Card Final Report, European Parliament, Directorate General for Research, Document for STOA Panel (2001) pp. 16-29.
- [3] Z. Chen, Java Card™ Technology for Smart Cards Architecture and Programmer's Guide (Addison-Wesley, Massachusetts USA, 2000).
- [4] Transcards, GIE Sesam Vitale, Transcards Project, URL: [http://www.sesamvitale.fr/html/projets/transcards/tcd\\_accueil\\_eng.htm](http://www.sesamvitale.fr/html/projets/transcards/tcd_accueil_eng.htm), last accessed: 2002
- [5] Netlink, GIE Sesam Vitale, Netlink Project, URL: <http://www.sesamvitale.fr/html/projets/netlink/index.htm>, last accessed: 2002
- [6] Netcards, GIE Sesam Vitale, Trans-European Healthcare Facility Service for Mobile Citizens, URL: [http://www.sesam-vitale.fr/html/projets/netcards/index\\_eng.htm](http://www.sesam-vitale.fr/html/projets/netcards/index_eng.htm), last accessed: 2002

- [7] Netcards Project, Netcards Consortium, Trans-European Healthcare Facility Service for Mobile Citizens, URL: <http://www.netcards-project.com/>, last accessed: 2003
- [8] R. Novak , G. Kandus and D. Trcek, Further development of a smart-card based health care information system in Slovenia, presented at the Fifth International Congress on Conference and Exhibition on Cards Applications in Health Care: Health Cards'99, (Milan Italy, 1999).
- [9] R. Novak, G. Kandus and D. Trcek D., Slovene smart-card and IP based health-care information system infrastructure, International Journal of Medical Informatics, vol. 61, Elsevier (2001) pp. 33-43.
- [10] K. Anderson, N. Marshall, M. Melnyk and L. Schaefer, Smart Cards in Health Care Industry, Management of Technology I (1997).
- [11] Health Smart Card, Health Smart Card, URL: [www.healthsmartcard.net](http://www.healthsmartcard.net), last accessed: 2003
- [12] CEN/TC 251: European Standardization of Health Informatics, URL: <http://www.cen251.org/>, last accessed: 2005
- [13] U. Hansmann, M. S. Nicklous, T. Schack and F. Seliger, Smart Card Application Development Using Java (Springer, Berlin Germany, 2000).

- [14] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad and M. Stal, Pattern-Oriented Software Architecture, Volume 1: A System of Patterns (John Wiley & Son Ltd, New York USA, 1996).
- [15] E. Gamma, R. Helm, R. Johnson and J. Vlissides, Design Patterns - Elements Of Reusable Object-Oriented Software (Addison-Wesley, Massachusetts USA, 1994).
- [16] OpenCard Consortium, OpenCard Framework 1.2 Programmer's Guide (IBM Deutschland Entwicklung GmbH, Boeblingen Germany, 1999).
- [17] W. Rankl and W. Effing, Smart Card Handbook Second Edition (John Wiley & Sons, West Sussex England, 2000).
- [18] A. Danny, S. Li, P. Houle, M. Wilcox, R. Phillips, P. Mohseni, S. Zeiger, H. Bergsten, M. Ferris, J. Diamond, M. Bogovich, M. Fleury, K. Vedati, A. Halberstadt and A. Patzer, Professional Java Server Programming: with Servlets, Java Server Pages (JSP), XML, Enterprise Java Beans (EJB), JNDI, CORBA, Jini and Java spaces (Wrox Press Inc., USA, 1999).
- [19] H. Heathfield, D. Pitty, R. Hanka, Evaluating information technology in health care: barriers and challenges, Vol. 316, British Medical Journal (1998) pp. 1959-1961.

- [20] H. A. Heathfield, V. Peel, P. Hudson, S. Kay, L. MacKay, T. Marley, L. Nicholson, R. Roberts, J. Williams, Evaluating Large Scale Health Information Systems: from Practice Towards Theory, Proceedings AMIA 1997 Nashville, Hanley and Belfus, Inc., Philadelphia, PA, (1997) pp. 116-121.

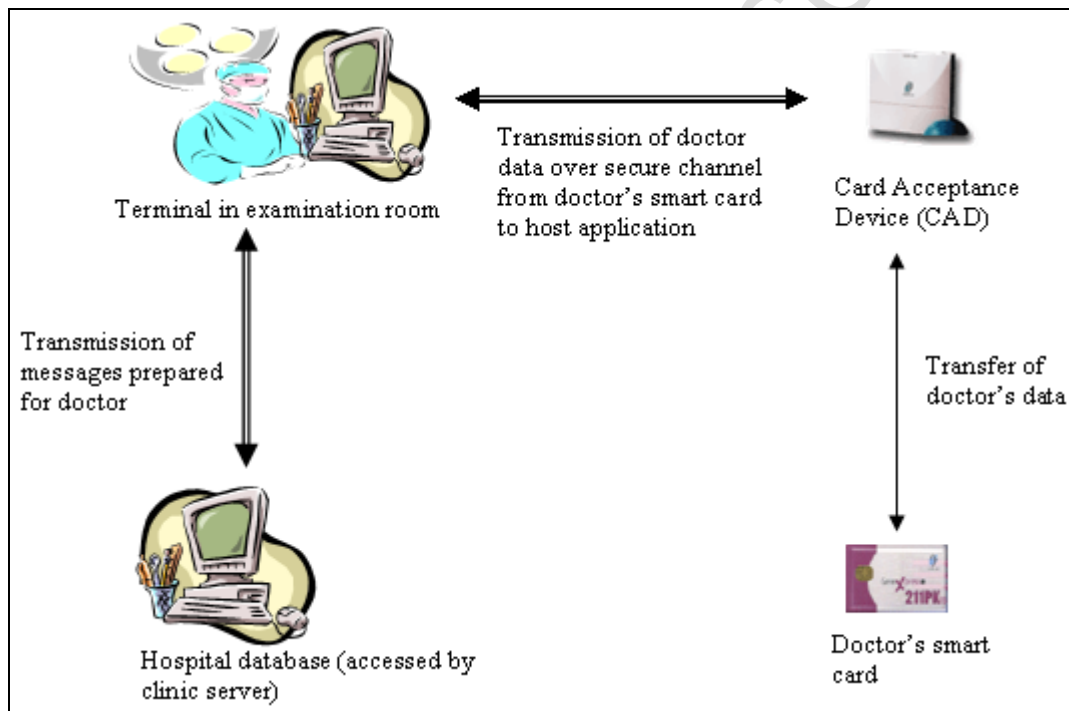


Figure 1: Doctor smart card session. When doctor inserts his smart card into the card acceptance device, his data is transferred in a secure way from his smart card to the application running on his PC after the mutual authentication and successful PIN entry. During this session initialization PC application also communicates with hospital database for notification service according to the communication data stored on card. After all, secure doctor session is opened and the doctor is ready to accept patients for examination.

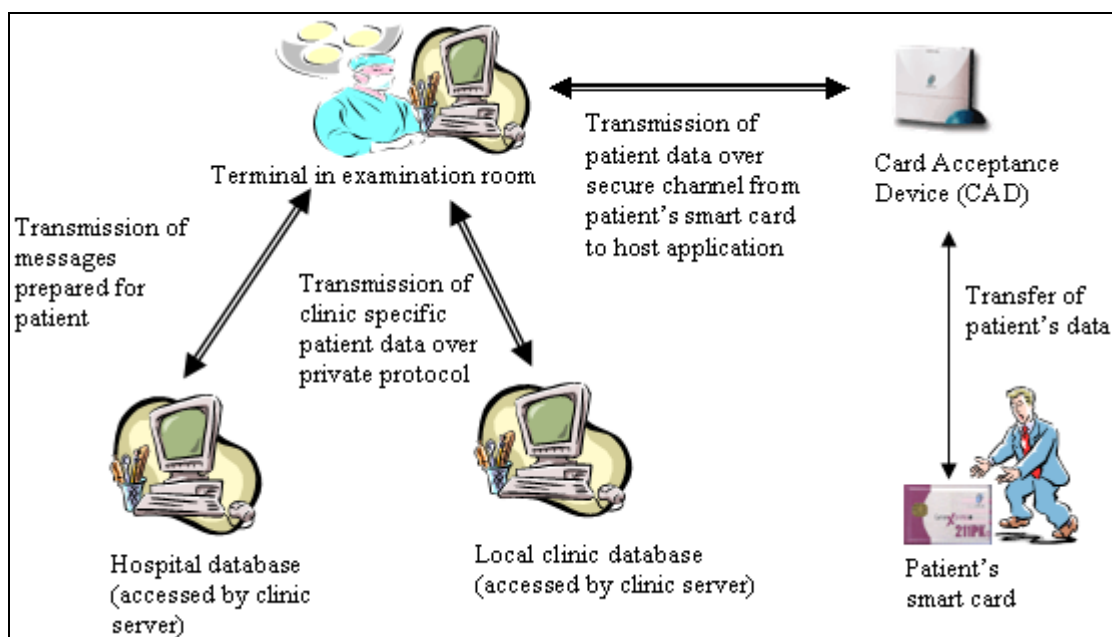


Figure 2: Patient smart card session. When patient's card is inserted into the card acceptance device, his data is transferred in a secure way from his smart card to the application running on doctor's PC after the mutual authentication and successful PIN entry. During this session initialization PC application also communicates both with hospital database and local clinic database to retrieve general health messages about the patient and clinic based patient information respectively.

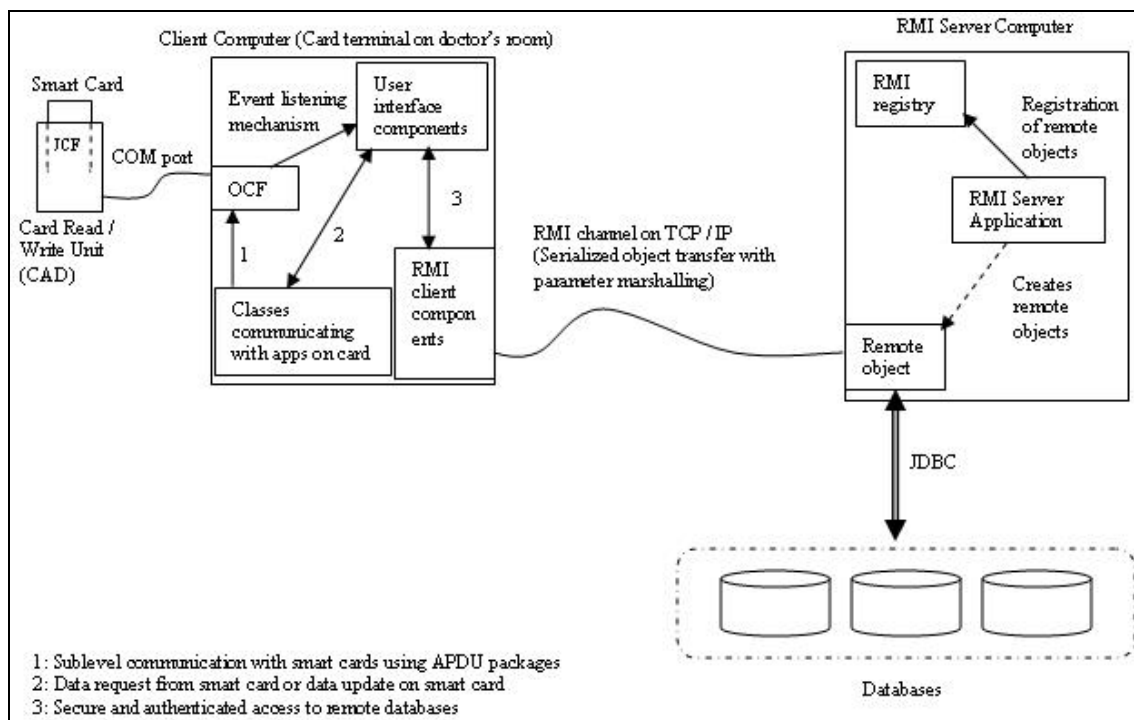


Figure 3: System architecture of SCHS. Four main system components are smart cards, client computers, RMI server computers and system relational databases. Each client computer residing on examination rooms has a CAD to communicate with system user's smart cards. Client computers can communicate with RMI servers to access distributed database of the hospital. The communication in case, is realized via a system specific secure protocol. Remote objects residing in RMI servers access databases on behalf of the requester clients (doctors).

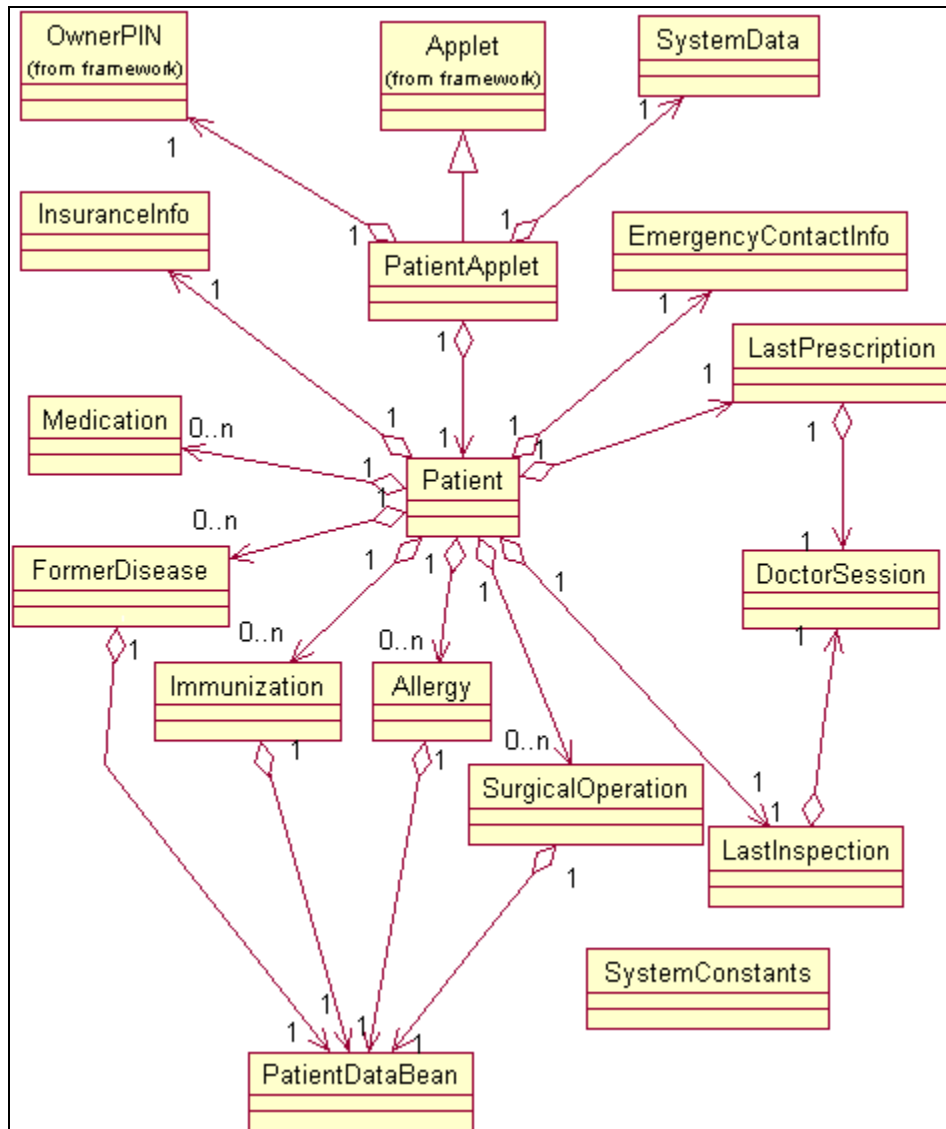


Figure 4: Object model of the software running inside a patient's smart card. Off-card client applications communicate with the PatientApplet object to retrieve patient data. PatientApplet is a JavaCard applet and aggregates objects those are encapsulating owner's PIN, system and patient data. The object called Patient stores a patient's data in related object collections (immunizations, allergies, etc.) and object instances (insurance info, last prescription, etc.).



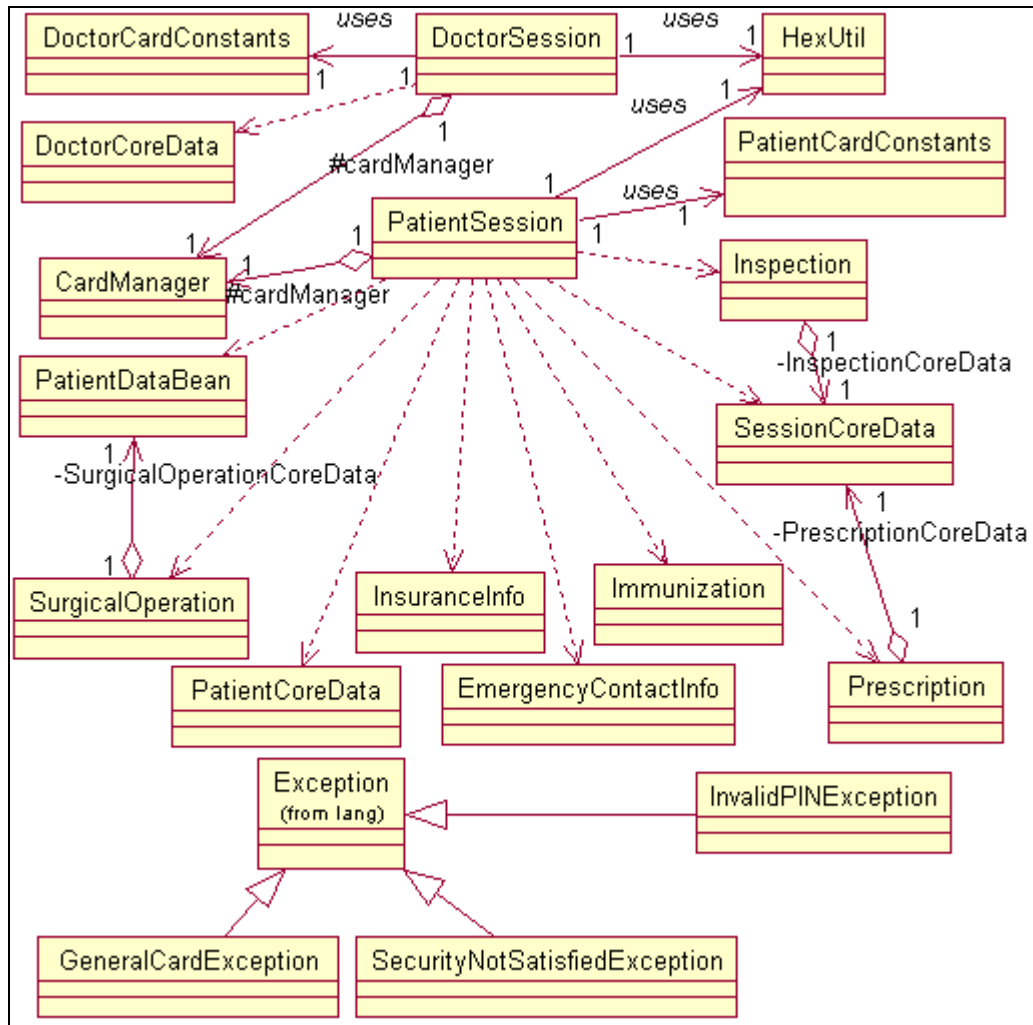


Figure 5: Object model of the software that handles smart card communication and session management on behalf of client interface components.

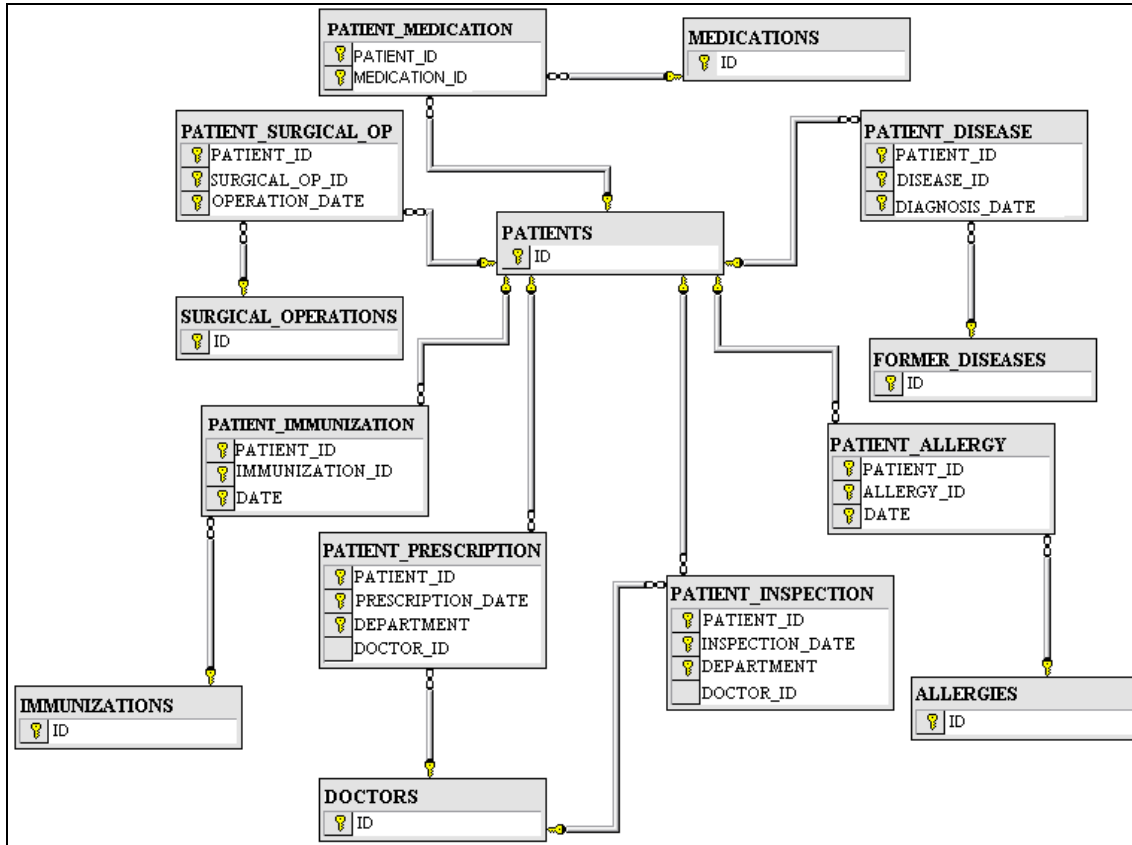


Figure 6: Model diagram of the designed hospital database. The central database is made up of 14 tables in which data about system users (both patients and healthcare professionals) are stored. Table relations – due to space limitations, only regarding primary and foreign key are shown.

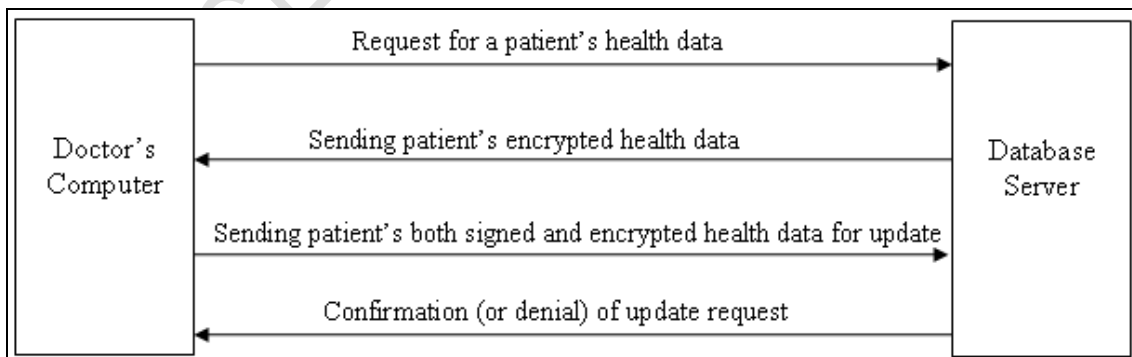


Figure 7: Secure and authenticated communication between a doctor's computer and a database server. All communication is realized inside the established RMI channel which is on top of TCP/IP.

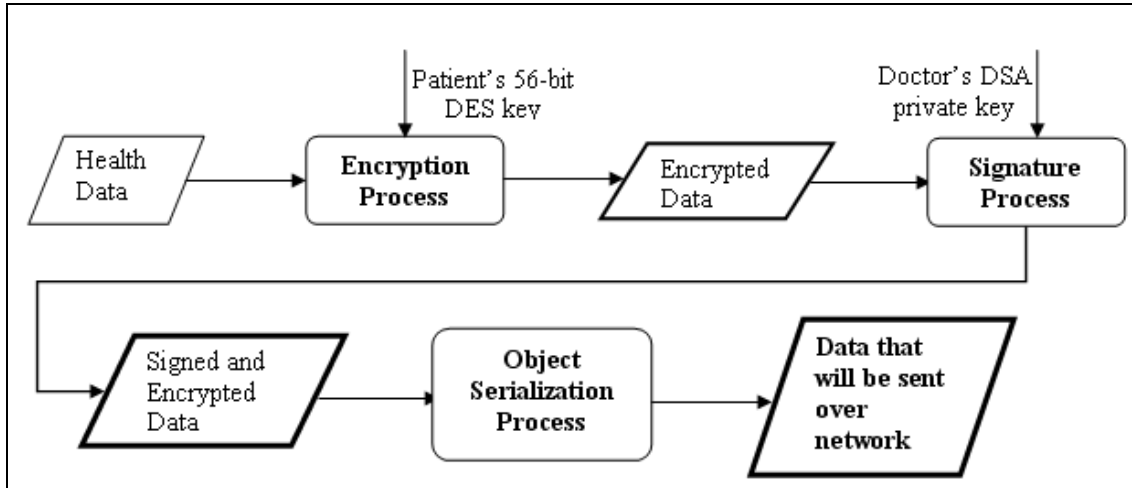


Figure 8: Flowchart that depicts security and authentication operations performed on a patient's health data. Outcome will be sent from a doctor's computer to a remote database server for an update.

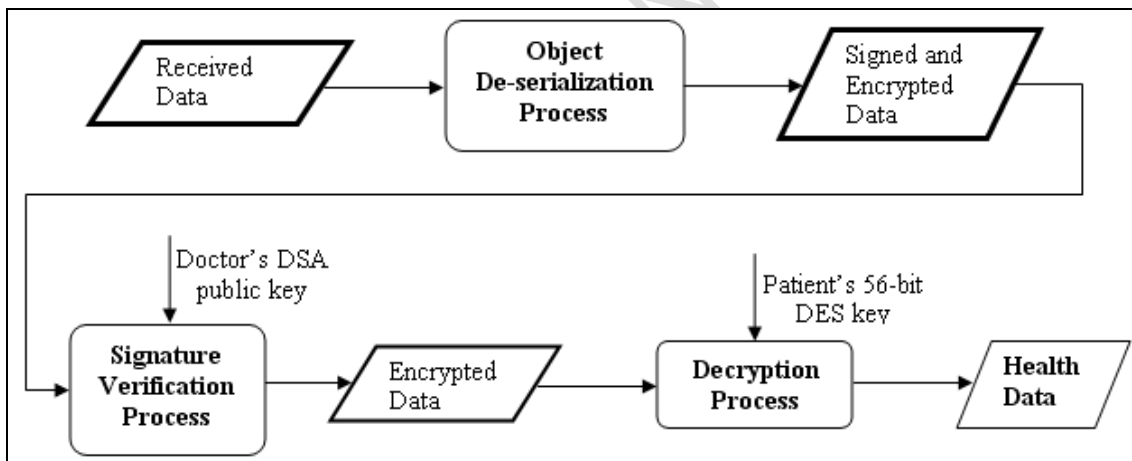


Figure 9: Flowchart that depicts server side operations performed on the received data. Operations are in reverse order as it is expected: Received data is first deserialized, then verified and finally decrypted with appropriate key.

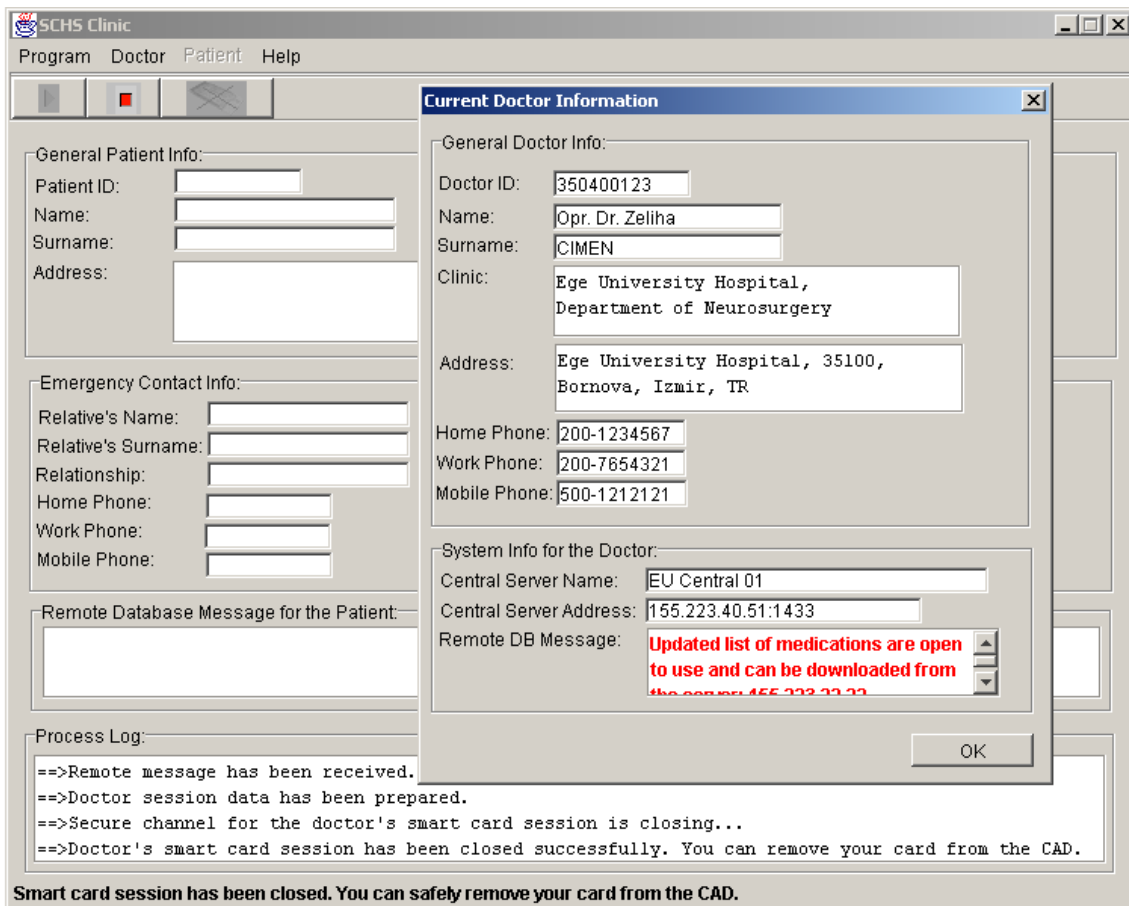


Figure 10: Screenshot of the SCHS client application running on a neurosurgery clinic in which a doctor has recently opened a doctor session by using her smart card. Notice that patient features of the program are currently disabled.

The screenshot displays the SCHS Clinic application interface. The main window is titled "SCHS Clinic" and has a menu bar with "Program", "Doctor", "Patient", and "Help". Below the menu bar are several icons. The main area is divided into several sections:

- General Patient Info:** Patient ID: 123456789, Name: Zafer, Surname: CELIK, Address: Flag avenue, 123/54, Karsiyaka - Izmir, TR. Home phone: 232-223, Work Phone: 232-323, Mobile Phone: 500-211, Birth Date: 1965-12, Gender: M, Blood Type: A Rh(+).
- Emergency Contact Info:** Relative's Name: Melisa, Relative's Surname: CELIK, Relationship: Spouse, Home Phone: 232-2233445, Work Phone: , Mobile Phone: 500-2442234.
- Insurance Info:** Official Foundation: Pension Fund for Civil Servants, Official SSN: 65-0325698, Private Foundation: , Private SSN: .
- Remote Database Message for the Patient:** Dear Zafer CELIK, You are kindly expected to participate into your routine checkup that will be hold on in 05/05/2004 at cardiology clinic.
- Process Log:**
  - ==>Immunization records have been transferred from the smart card.
  - ==>Former disease records have been transferred from the smart card.
  - ==>Permanent drug records have been transferred from the smart card.
  - ==>Patient's medical information has been successfully listed.

A secondary window titled "Current Patient's Medical Information" is open, showing:

- Patient's Allergies:**

Allergy	Diagnosis Date
Allergic Catarrh	2003-03-07
Dust	2003-03-07
Pollen	2001-03-02
Food (Milk)	2000-05-04
- Patient's Immunizations:**

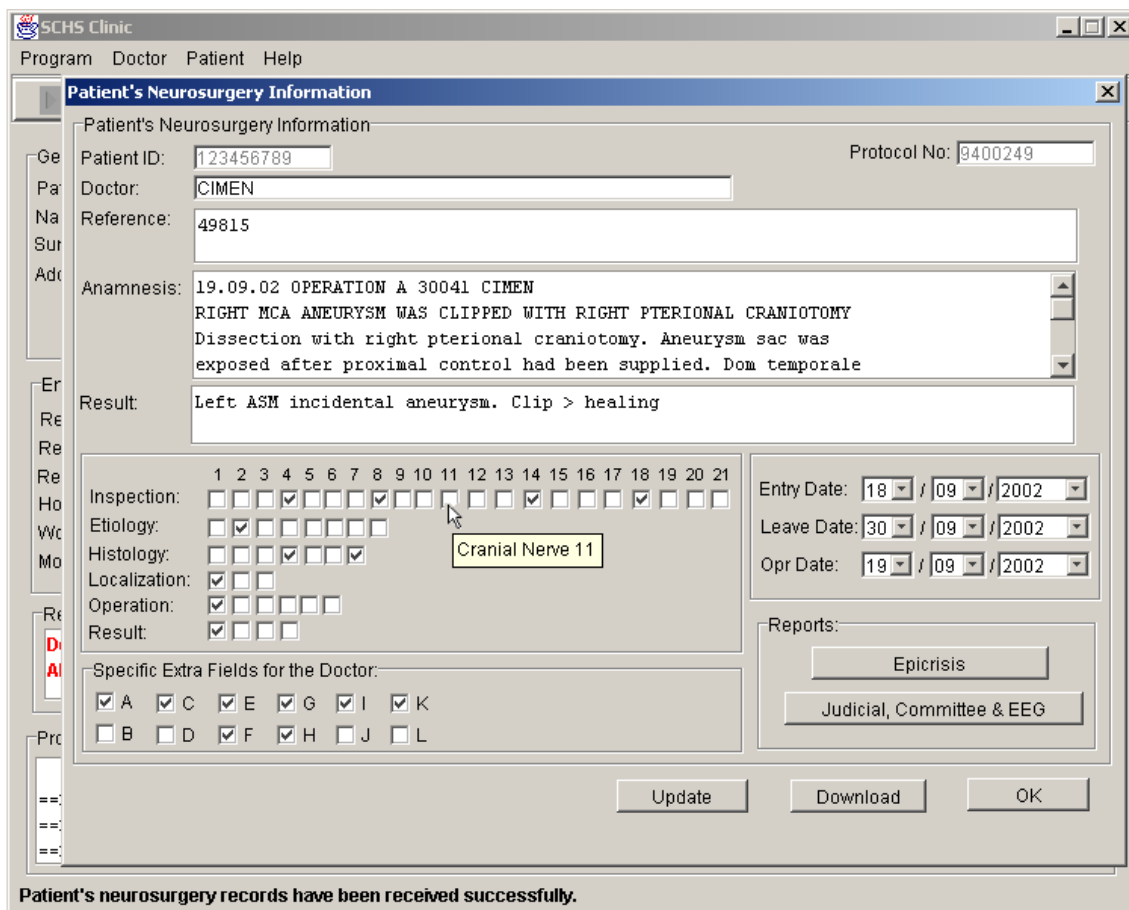
Immunization	Date
Influenza	2003-03-07
Hepatitis A	2002-04-04
Measles	1985-01-01
- Patient's Former and/or Chronic Diseases:**

Disease	Diagnosis Date
Diabetes	1998-04-02
Migraine	1997-09-03
Thalassemia	1996-07-04
Ulcer	1990-06-06
- Drugs Permanently Used by the Patient:**

Immunization	Dose
Apranax Fort 10 Tb	2 t/d
Levotiron Tablet	5 t/d
Ecopirine 150 Mg.	5 gr/h
Ferrum Hau. Amp.	1 t/h

At the bottom of the secondary window are "Download" and "OK" buttons. A status bar at the bottom of the main window reads "Patient's medical information has been successfully listed."

Figure 11: Screenshot of the SCHS client application running on a neurosurgery clinic in which a patient smart card session has been recently opened. All data on display has been transferred from the patient's smart card except the remote health message for the patient. Application has communicated with the remote central database by using the access information stored on patient's smart card.



**SCHS Clinic**  
 Program Doctor Patient Help

**Patient's Neurosurgery Information**

Patient's Neurosurgery Information

Patient ID: 123456789 Protocol No: 9400249

Doctor: CIMEN

Reference: 49815

Anamnesis: 19.09.02 OPERATION A 30041 CIMEN  
 RIGHT MCA ANEURYSM WAS CLIPPED WITH RIGHT PTERIONAL CRANIOTOMY  
 Dissection with right pterional craniotomy. Aneurysm sac was exposed after proximal control had been supplied. Dom temporale

Result: Left ASM incidental aneurysm. Clip > healing

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Inspection:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Etiology:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Histology:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Localization:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Operation:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Result:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Entry Date: 18 / 09 / 2002  
 Leave Date: 30 / 09 / 2002  
 Opr Date: 19 / 09 / 2002

Reports:  
 Epicrisis  
 Judicial, Committee & EEG

Update Download OK

Patient's neurosurgery records have been received successfully.

Figure 12: Screenshot of the SCHS client application running on a neurosurgery clinic in which a patient's neurosurgery information has been retrieved from the department's database over the secure communication protocol. Both keys stored in the doctor's and patient's smartcards have been used during transmission of the data.