DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SECURE MOBILE IMPATIENT IDENTIFICATION IN HOSPITAL

by
Sezer BAYTAR

July, 2015
İZMİR

SECURE MOBILE IMPATIENT IDENTIFICATION IN HOSPITAL

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Computer Engineering Program

by
Sezer BAYTAR

July, 2015
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"SECURE MOBILE IMPATIENT IDENTIFICATION IN HOSPITAL"** completed by **SEZER BAYTAR** under supervision of **ASST. PROF. DR. SEMİH UTKU** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Semih UTKU

Supervisor

Assoc. Prof. Dr. Mehmet S. Ünlütürk
(Jury Member)

Asst. Prof. Dr. M. Hikmet Özcanhan
(Jury Member)

Prof.Dr. Ayşe OKUR

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

I would like to thank my advisor Asst. Prof. Dr. Semih Utku for his help, suggestions and guidance to this study.

I would like to thank my co-workers for their support, especially to bro.

Finally, I would like to dedicate this work to my mom for her supports, prayers and sacrifices since I was born.

This thesis was supported by TUBITAK 1002 program; Project No: 113S419, Project Name: “Yatışlı Hastaların Mobil Cihazlar ile İlaç Takip Sistemi”.

Sezer BAYTAR

SECURE MOBILE IMPATIENT IDENTIFICATION IN HOSPITAL

ABSTRACT

Wireless networking and communication technologies have created different types of wireless systems and solutions for hospitals. These systems are envisioned to coordinate with each other to provide ubiquitous services to hospital staff. However, wireless network solutions leads to the formation of vulnerability for data transmission. The ubiquitous systems used in hospitals have been developed to eliminate or reduce these problems. The main focus of all these solutions is to increase patient safety and quality of patient care. In this thesis, a software solution called IMS-Mobile is developed to solve inpatient drug administration process (IDAP). The proposed IDAP architecture is a complete ubiquitous system and it supports stronger security primitives. The system was developed with Android based NFC which supported mobile devices. In this study, a secured server-mobile device data exchange with asymmetric encryption was constructed. The standard Mifare DesFire EV1 authentication protocol was rewritten, where the authentication steps are carried out by the server. In this way, data sharing during transmission has become more secure. With a mobile application, the process of inpatient hospital stay has been under control until the patient is discharged from the start. Time controls were added to eliminate the problems of correct patient's medication. The system was analyzed for performance, cost and security. Evaluation shows that the proposed system has stronger security, equal efficiency and at little extra cost, than previous works.

Keywords: NFC, secure inpatient tracking, ubiquitous systems

HASTANEDE GÜVENLİ MOBİL YATIŞLI HASTA TAKİP SİSTEMİ

ÖZ

Kablosun ağ ve iletişim teknolojileri hastaneler için birçok kablosuz sistem ve çözümlerin geliştirilmesini sağlamıştır. Bu sistemler hastanelerde bütünleşik sistemlerin geliştirilmesi için birlikte çalışacak şekilde planlanmıştır. Buna rağmen kablosuz ağ çözümleri veri transferinde güvenlik açıklarına sebep olmaktadır. Hastanelerde kullanılan bütünleşik sistemler bu sorunu ortadan kaldırmak için geliştirilmiştir. Bu çözümlerin ortak odak noktası hasta güvenliğini ve hastaya uygulanan tedavinin kalitesini artırmaktır. Bu tezde yatışlı hasta ilaç kullanımı yönetimi(İKY) işlemini yönetmek için IMS-Mobile isimli bir yazılım geliştirilmiştir. Geliştirilen İKY mimarisi tamamen bütünleşik olup ekstra güvenlik temelleri sağlamaktadır. Sistem Android tabanlı NFC desteği sağlayan mobil cihazlar ile geliştirilmiştir. NFC noktadan-noktaya, kısa mesafeli, dokumadan çalışan bir kablosuz iletişim teknolojisidir. NFC iletişim protokolleri ve veri değiş-tokuş format standartları ile ISO IEC 14443 standartını içerir ve NFC desteği sunan cihazlar arasında veri iletişimi sağlar. Bu çalışmada asimetrik şifreleme kullanılarak cihaz-sunucu arasında güvenli iletişim altyapısı oluşturulmuştur. Standart MiFare DesFire EV1 yetkilendirme protokolü yeniden yazılmıştır. Yetkilendirme adımları cihazdan sunucuya taşınmıştır. Böylece veri transferi daha güvenli hale getirilmiştir. Mobil uygulama ile birlikte bir hastanın hastaneye girişinden çıkışına kadarki tüm süreçler kontrol altında tutulur. Zaman kontrolleri uygun hastaya doğru tedavinin uygulanması sırasında oluşabilecek problemlerin giderilmesi için eklenmiştir. Sistem performans, maliyet ve güvenlik açısından analiz edilmiştir. Değerlendirmeler geliştirilen sistemin güçlü güvenlik altyapısı sağlarken aynı zamanda etkili ve az maliyetle çalışabildiğini göstermektedir.

Anahtar kelimeler: NFC, güvenli yatışlı hasta takibi, bütünleşik sistemler

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

Medical professionals want to ensure that patients are checked, treated, medicated properly. They want to track their patients because present patient status is important in making next treatment right. This helps to treat patient in right way and cure them fast, with less effort. Also they can learn patients status if other department has threated the patient. This prevents to treat in wrong way which can cause side effects to patient. Also tracking patient's results in higher patient satisfaction.

With this information in hand, it is obvious there must be an application to help doctors to track their patients. But only one application cannot be effective by itself. Thus tracking applications must work with hospital information systems and other 3rd party assistant applications, such systems are named as Ubiquitous Systems.

With fast growing of Wireless Technologies, Ubiquitous Systems (US) are getting more important duties on health sector and this is helping to gain more benefits. With improvement of US, lack of control on tracking patients will be decreased and patients will get better service from hospital. This will help to satisfy patients more and give them right treatment.

Mobile systems are growing with wireless technologies' rapid deployment. This helps to using wireless technologies easily and adapting ubiquitous systems to HIS with low costs and work. With combination of mobile systems and wireless technologies, wireless sensors such as Radio Frequency Identification (RFID), Near Field Communication (NFC) or small sensor are getting more important role.

US contains sensors, wireless communication devices, patient tracking systems and moreover. These systems helps to track patients in better way and to provide better services with chance to detect possible disease at early stage. US are also helps to provide staff tracking, helping staff to serve easy and better service, giving patients true and right dosage of medicines and more.

US with Wireless Communication Technology have some problems. These problems are transferring data in correct form in right time, losing data on transfer step and most important one is transferring data in a secure way. Patient information is top secret and must be secured from 3rd person who want to steal these data. This is the most common problem with wireless communication.

There are different types to create a secure mechanism. Barcode technology is one them. But there are problems with barcode technology. Low Image quality, being easy to copy barcodes, hard to identify barcodes easily touching or reading are some of the reasons of barcode technologies failing (Wilson & Sullivan, 2004). To secure data RFID technologies are a key to prevent accessing sensible data. It is easy to identify RFID tags and it is not possible to copy them easily. Using right encryption methods and securing encryption keys on RFID tags may be the key of making a well-designed infrastructure.

In theory RFID technology has a well-designed security mechanism (Wei, Chao, & Quan, 2014). But there are different attacks to RFID and vulnerabilities on RFID technology (Hutter, Schmidt, & Plos, 2008). If RFID tag is not well protected, it is possible to sniff, listen traffic or denial of service with third party unauthorized readers. To prevent this attacks NFC technology is presented (Pateriya, & Sharma, 2011).

NFC is a short-range data-transmission system which covers a wireless communication protocol and data exchange formats that allows secure exchange of small amounts of data by proximity or touch. The standard is based on existing radio-frequency identification (RFID) standards including ISO/IEC 14443 (Agrawal & Bhuraria, 2012).

NFC is a point-to-point short-range wireless contactless technology. NFC standards cover communications protocols and data exchange formats including ISO IEC 14443 and allows sending and receiving messages between two NFC-enabled devices.

In the process of increasing patient safety and healthcare quality five-right method, namely; right method, right medication, right dose and right time method need to be implemented properly (Fernando, McKinstry, & Sheikh, 2006). After evaluating the patient, Care needs of patients should be identified and planned. For a reliable care process in accordance with the patient's needs;

- Treatment which will be applied should be determined
- Process which will be applied to patient should be planned (anesthesia, surgery, medicine, nutrition)
- All information about the patient's treatment should be given to patients and relatives
- Nursing care should be planned
- Patient records should be kept regularly

HL7 is a world standard for exchanging medical data between health information systems in world. HL7 keep within different standards such as CEN/TC 251, W3C, DICOM, ISO/TC 215 etc. Context Management Specifications (CCOW) is a standard of HL7 for applications or runtime environments that complements HL7 standards. CCOW is the main part of HL7 that ensures security and consistent access to patient information. Other one of important standard of HL7 is Messaging Standard (Dolin et al., 2001). This is the most implemented standard in healthcare world and it is the key of the data exchange. This standard allows health systems to exchange data between them.

In this study, mobile impatient tracking system has been developed for the drug administrations, treatment planning and care process for impatient. Hospital stay and drug administration process for an impatient can be monitored through this system with a secure structure. System provides to verification of patient information and establish a link between the patient and nurses/doctors.

IMS-Mobile consists of NFC tags and mobile tablets which used by hospital staffs. Tablets are used to check patient’s wristband to verify they are administering the right medicine to the right patient, at the right dose, at the right time, and by the

right route. NFC data communication protocol is cited on the server by changed NFC official protocol. Authentication and data security process is carried out in the server. Tablets are only able to read NFC tags and pass reading data to IMS-Server. The pre-shared encrypted keys are used for patient authentication to avoid vulnerability.

The system is also connected to the hospital information system (HIS). Basic information about the patient is transferred from HIS Server to IMS Server. The IMS-Server use HL7's Version 2.x (V2) messaging standard for communications between hospital information systems to transfer patient's data. IMS-Mobile application was developed based on the CCOW standard. Confidentiality, maintain data integrity and ensure data security should be planned according to HL7 for prevention of medical errors (Benson, 2012).

CHAPTER TWO
RELATED WORKS

Shojania et al. (2002) brings up six chief goals for institution of medical care for the 21st century: (1) safety: preventing patient to harm by treatment; (2) effectiveness: reducing too much or too little drug use after scientific knowledge; (3) patient-centered: giving right treatment to patient after their preferences, status, previous treatment info, needs; (4) timeliness : eliminating wait and lateness of patient ; (5) efficiency: eliminating unnecessary resource consume ; (6) equity: giving same treatment to patients without considering their gender, age or region (Shojania, Duncan, McDonald, & Wachter, 2002). Safety is one of the most important of these six important goals (Sun, Wang, & Wu, 2008).

Medication errors are the most important problems in hospitals. To prevent medication errors many reports and standards are prepared for the many Institutes and governments. Institute of Medicine puts the number of people losing their lives to 98,000/year, due to medication errors in the U.S.A. (Özcanhan, Dalkılıç, & Utku, 2014). The same institution reports that about 530,000 preventable "adverse drug events" (ADE) happen each year (Cullen et al., 1997). According to studies, it is preventable to 28% to 75% of ADEs per 100 hospital admissions which have between 6.5 and 15 ADEs rate (Bates et al., 1995). 28% of ADEs of all hospital occurred during medication administration and less than 2% of ADEs of all hospital occurred before completion of administration (Voshall, Piscotty, Lawrence, & Targosz, 2013). Activities are done up to 28% by nurses and 15% by doctors in hospital. Accordingly this percentages, it is essential to have a medication administration for hospital-based nurses (Keohane et al., 2008).

In the United States, for reducing the medication errors in hospitals, the Food and Drug Administration (FDA) has introduced a new rule, which requires the labels of drug, to have barcode on it. However, there will be no gain in patient safety. Because of the implementation details of the new rule is not mentioned by FDA to hospitals. There are many studies and proposals are presented to reduce medication

errors in hospital. These are to support the integration of new technologies in hospitals. An electronic medication administration system and using barcode technology with medicine shown that reducing time errors by 41%, wrong drug use by 51% and eliminate treatment errors after transcription (Poon et al., 2010). Another proposed solution is Wisely Aware RFID Dosage (WARD) system, which is an integration of barcodes and RFID tags (Sun, Wang, & Wu, 2008). With the usage of the RFID and barcodes, the WARD system can build an effective and safe patient care environment and provide the guard of reducing the risk of medication error. But in these solution based on barcode technology. Therefore poor image quality and flawed data quality affects the system achievements. On the other hand security solutions are not available in these solutions.

RFID technology provides significant utility in health care (Ngai, Poon, Suk, & Ng, 2009). Different solutions proposed for RFID in hospitals. Authenticating users, tracking patients and staff, safe and secure treatment to patient operations are can be implemented with using RFID solutions. RFID implementations, for example, store patient identifiers when used in blood transfusion medicine (Hohberger, Davis, Briggs, Gutierrez, & Veeramani, 2012) and patient tracking (Rosenbaum, 2014).

Huang et al. (2009) introduced a grouping proof protocol to verify RFID tags on patient bracelets and medication containers (Huang & Ku, 2009). Unfortunately, the proof was later found to suffer from denial of service (DOS) and replay attacks. Yu et al. (2012) subsequently demonstrated in- creased security using a lightweight binding proof protocol for RFID and medication authentication and verification (Yu., Hou, & Chiang, 2012). The protocol focuses on using existing low-cost RFID tags while increasing security measures. Additional protocols focused on privacy, security, and safety has also been proposed (Hoque, 2010). For example, Chen et al. (2012) propose a tamper resistant prescription RFID access control protocol (Chen, Huang, Tsai, & Jan, 2012). The protocol authenticates the RFID reader and tags with one-way hash and encrypted data. In addition, a challenge-response method is employed to avoid replay attacks. A lot of security threats and attacking attempts exist in RFID system. It is not solved yet properly. Lightweight solutions have been

proposed for RFID, but they are still expensive and vulnerable to the security and do not fully resolve the security issues.

Sanchez et al. (2012) proposed PharmaFabula project to recognize medicines and report about the patient information (Sánchez, Mateos, Fraile, & Pizarro, 2012). They proposed the utilization of the NFC technology in the medical field for helping blind users. One early pilot assessed NFC as a tool for general nursing tasks and training, including e-MAR (Landman, 2014). NFC enabled mobile devices are used to track results (Coskun, Ozdenizci, & Ok, 2013) of treatments that are applied to patient which are compliant to medicine or not (Pitler & Bonomi, 2006). Pozzebon proposes other example application for NFC health care (Benelli, Pozzebon, & Parrino, 2010). These previous NFC health care applications have limited evaluations, no security primitives.

Privacy is the essential concern of patients and the biggest obstacle to e-healthcare deployment (Sharma, Ahmed, & Rathinasamy, 2005). Therefore, a novel, next generation near field communication-enabled medication administration solution has been developed (IMS-Mobile). System provides advantage of a mobile device equipped with a reader for NFC, a server-side wireless communication protocol that allows secure exchange of small amounts of data by proximity or touch. IMS-Mobile system was developed using the Google Nexus tablet. The Google Nexus was used because it was available with the Android 4.0 operating system and NFC support. The one of the major benefits of IMS-Mobile includes the increasing security, patient's management and managing documents with its mobility and usability. The main contribution of this paper is proposal of robust secure ubiquitous healthcare application using Android based mobile devices with NFC.

CHAPTER THREE
SYSTEM ARCHITECTURE

3.1 Android

Android is an open source mobile operation system based on Linux kernel. Android is being developed and led by Google. Android is developed primarily by for touch screen enabled devices such as mobile phones and tablets. Android also can work on TV's, cars or smart watches. Android gets touching as input like gestures, swiping, tapping, pinching or virtual keyboard.

3.2 IDE: Android Studio and Eclipse

Project development has been started with Eclipse ADT (Android Development Tool). Eclipse is an open source IDE. It can work on Linux or Windows. Eclipse is a flexible IDE and it is easy to start a project with eclipse. There are too many contributors are working to improve Eclipse IDE. Eclipse supports plugins and can be used for different platforms. But Eclipse has some problems due to the supporting multiple platforms. Developing Android project on Eclipse may be hard and some errors may occurs often.

Google decided to change Android development platform due to the Eclipses problems. And Android Studio's chosen as primary IDE to develop Android Projects. Android Studio is based on IntelliJ Idea IDE which is an open source product of Jetbrains.

There is difference between Eclipse ADT and Android Studio for project structure and building system. Thus ADT project must be converted to Android Studio project properly to be work.

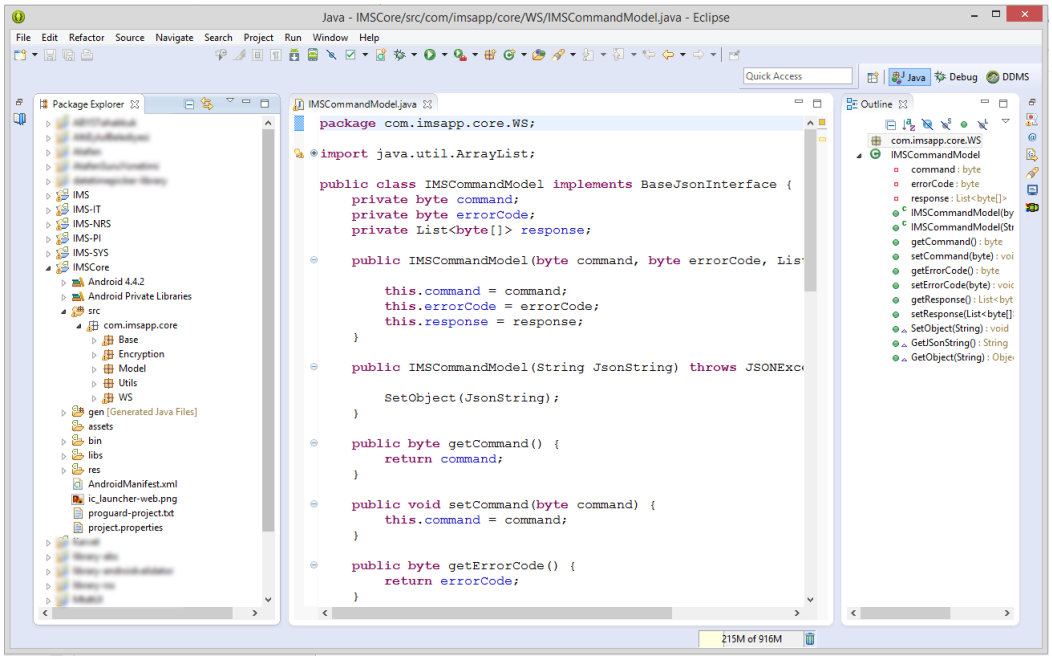

Figure 3.1 Eclipse IDE

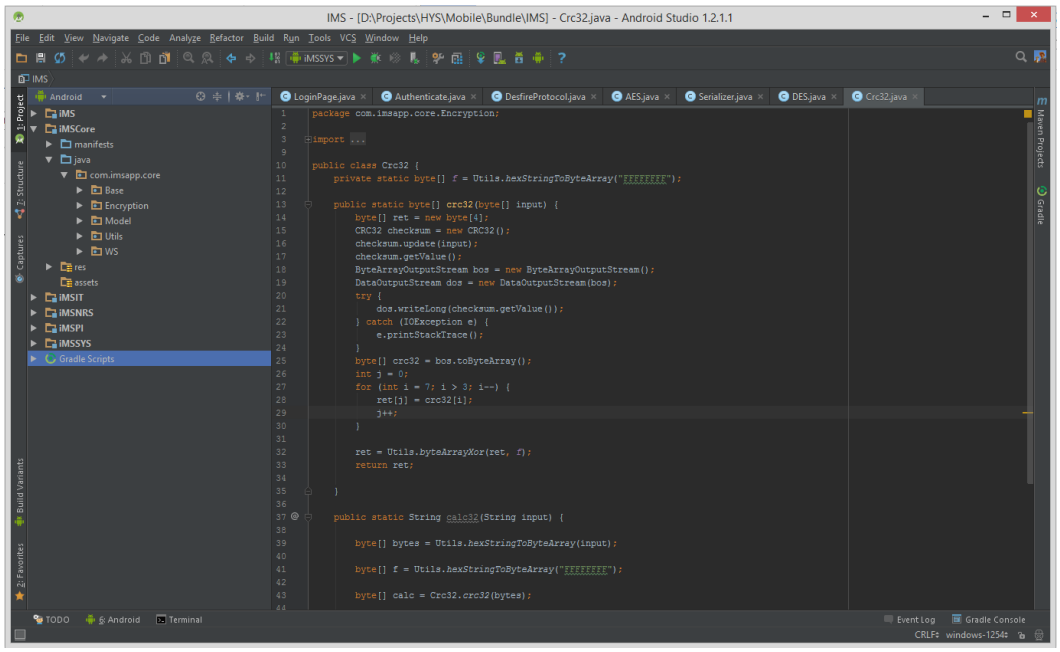

Figure 3.2 Android Studio

3.3 JAVA

JAVA is an Object Oriented Programming language which is class based and designed to have implement dependencies as possible as. It supports write once run everywhere which means compiled JAVA code can run on all platforms that support JAVA such as mobile devices, TV's cars, and computers.

Java was used for developing mobile application and web services in the project as primary language.

3.4 Application Background

There are several apps and they use common functions and objects mostly. A custom library has been created to write once use in all projects. This library contains helpers, common models, web service call mechanisms, security options, NFC operations and other common properties. This helps to maintain apps easily and apply fixes/updates to all projects easily. If I need to change a web service calling mechanism or NFC steps I just change it on custom library and all app automatically gets the new updates.

3.5 Mobile Application Development

Mobile application development started on Eclipse and after Google's Android Studio release, project moved to Android Studio. Application written with JAVA and KSoap2 Library used to make web service calls.

This application based on making secure call to web service with server and android devices idea. Application collects necessary data from user and sends them to server in a secure way. All users have their pre-defined NFC cards on system. A user must be logged id to app with his NFC card and user information to start application.

3.6 Architecture

System is working on a server with web-service that is connected to Hospital Information System database. There is a private database for IMS to store data separately HIS. This data includes NFC tag, card ids, logs, treatments etc. Also there is another service that carries data IMS-to-HIS and HIS-to-IMS. This helps to sync databases.

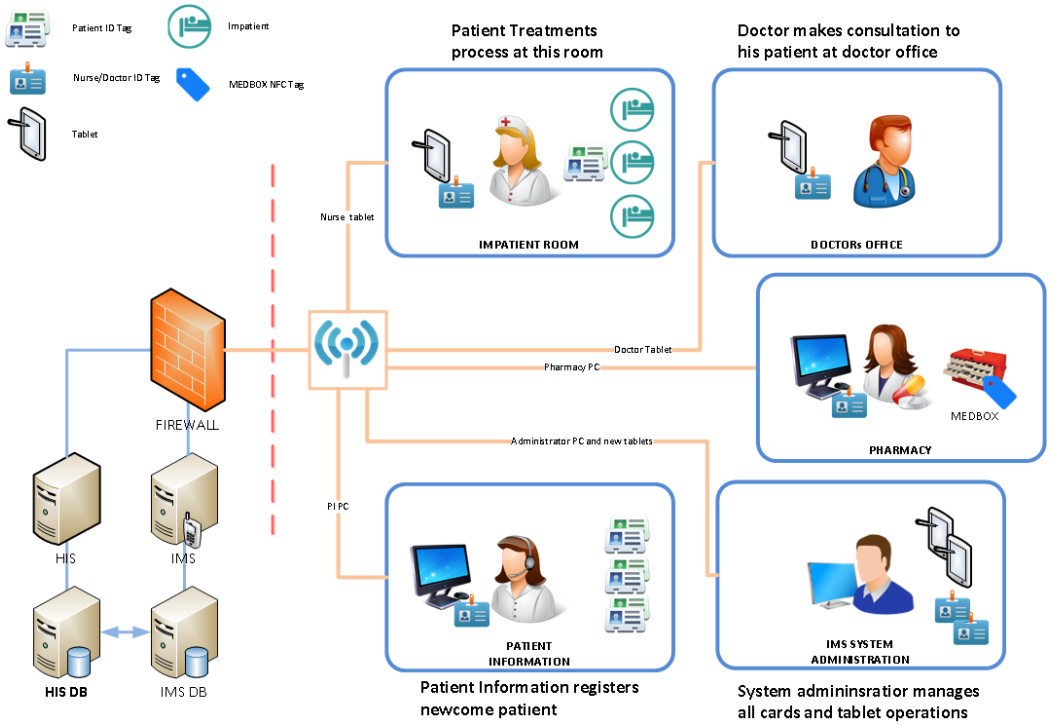

Figure 3.3 System architecture

3.7 Mobile Application

3.7.1 Core Library

This library includes base classes, core elements, logging mechanism

3.7.2 Utils

Includes utils such as helpers, static variables, constants or common functions.

3.7.3 Encryption

Includes encryption classes, files and helpers. AES, RSA, CRC32, DES and Serializer classes are belong to this project.

3.7.4 HTTPS Calls

Handles http web-service call operations. GET and POST HTTP methods are used to send and get data. Data is in JSON format and JSON parser belongs this project too.

3.7.5 Models

Object models are used for all projects. There are also project-specific models for all projects.

3.7.6 UI Elements

This are basic elements such as text edits, combo box, lists which are used for all mobile apps.

3.7.7 Applications

There are 5 applications for each hospital personnel. This applications builded up on core library. All applications are using core functions. Applications also may have their custom functions such as helpers, desfire protocols or different web service calls beside core functions.

3.8 Web Service

Web Service includes a few layer to work properly and securely. This layer handles different operations.

Web services are developed on Eclipse IDE for JAVA EE Developers. This version of eclipse is ready to develop web services with including required tools.

Axis2 technology was used to run web services on a Windows or Linux machine. SOAP protocol is used and data is in SOAP message as JSON format. MySQL is used for database. Finished web service is working on standalone Tomcat Server which can run on a Windows or Linux machine.

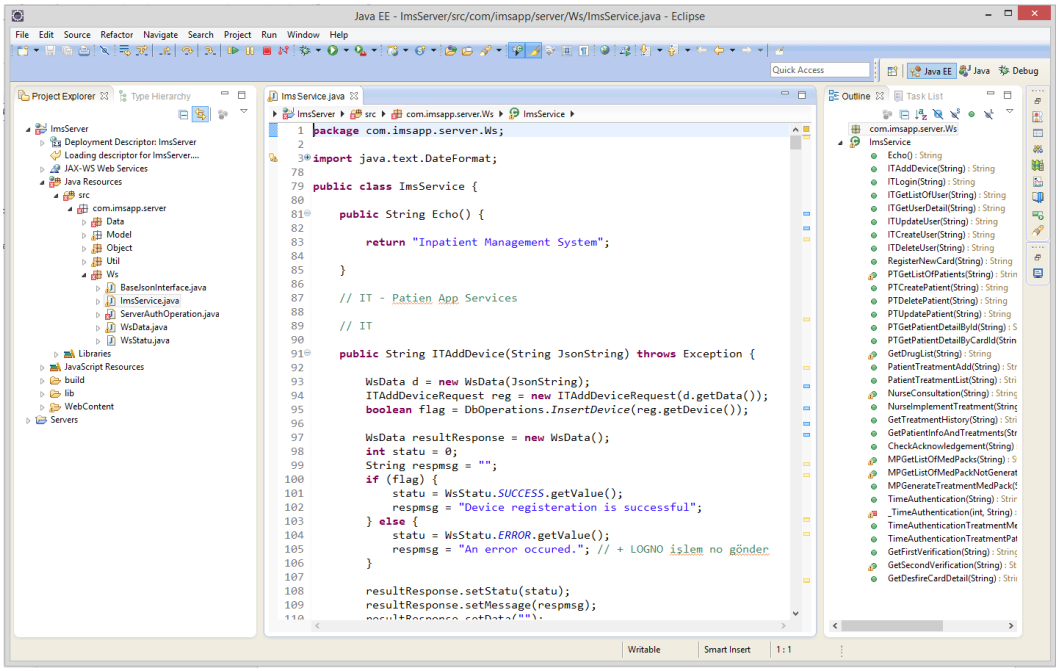

Figure 3.4 Eclipse IDE for Java EE

3.8.1 Infastructure

3.8.1.1 Axis2

Web service is developed with Axis2 web service engine. Axis2 helps to build a SOAP web service with WSDL endpoints. Axis2 is developed by Apache and Apache supports Axis2 continuously.

3.8.1.2 Tomcat

Web service deployed on a Tomcat Standalone application instance. Tomcat is a Java servlet. It is open source and developed by Apache.

3.8.1.3 SOAP/XML

Web Service implements SOAP actions to communicate between clients. Data between client and server is in XML shape. This helps to parse data easily.

3.8.1.4 Database

MySQL database is used for data storage. MySQL is an open source database management system. It is free to use and MySQL has high performance and security.

3.8.2 Security

Securing web service and messaging are provided by enabling https and encrypting body of soap messages. Also some sensitive data is hashing to compare between device and server. With all this securing mechanisms, to ensure a more secured communication on network, HTTPS over SSL/TLS is enabled.

3.8.2.1 HTTPS

HTTPS (HTTP over TLS or HTTP over SSL) is a communication protocol to secure communication over a network. HTTPS is layering HTTP (Hypertext Transfer

Protocol) on SSL or TLS protocol. This enables security mechanism with SSL or TLS capabilities.

HTTPS protocol works between to device over network. Client sends data with encrypting it with using server's public key and server decrypts data with its private key. In reverse way server encrypts data with its private key and client can decrypt it with server's public key. This helps to prevent decrypt data without public or private key of server.

Web browser or other clients checks the SSL certificate of the servers. If SSL certificate is not created by trusted firms (CA : Certificate Authority) client warns or throws error. Android application gives an error for not signed certificates by CA. In this project, a custom certificate was created and signed with using OpenSSL.

The OpenSSL project is a toolkit to develop full featured of SSL and TLS which is developing by community as open source. OpenSSL can create SSL certificates and sign them as CA.

3.8.2.2 ENCRYPTION

Encryption is used to encrypt SOAP messages. SOAP messages are XML files and body of messages contains important data. This body message is in JSON string format. JSON messages are being encrypted by sender and decrypted by receiver. Encryption method is AES and encryption key is pre-shared between sender and receiver. Pre-shared key is 128 Bit and it is not crackable with current solutions.

3.8.2.3 AES

AES is the one of the most known and strongest encryption algorithm which is based on Rijndael cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen. AES is a symmetric-key algorithm which mean is encrypting and decrypting operations use same key.

3.8.2.4 SHA512

SHA512 is a hashing algorithm which includes SHA-2 cryptographic hash functions set which is designed by NSA. This algorithm is being used to check NFC card data and server data equality by hashing them before comparing. This helps to send data between device and server openly. 32 bit words is used to hash data.

3.8.2.5 RSA

RSA algorithm works with public and private key mechanism which means RSA is an asymmetric encryption algorithm. A data which is encrypted with public key can be decrypt only with private key. To achieve this mechanism on the system. Tablets are being registered to system with generating a private key and n public key. After key generation, Private Key is stored on tablet and one public key is stored on the server. This help me to decrypt data with public key of tablet which is encrypted on tablet with its private key.

3.8.2.6 NFC

Nfc layer handles Mifare Desfire protocol. This layer creates commands to be run on NFC card. Layer is responsible for check is card registered, is it a compatible with system, is it changed by other person etc. This layer contains same protocol with NFC card. So service know response of the command before sending it to card. When card gets the command and runs it and returns a response. Web service decrypt the response and checks it. If there is an unexpected response from card, service sends tablet error message and stops responding until authenticating with real card.

3.9 NFC

NFC is a new technology to enable mobile phones and other devices to establish connection via radio frequency by touching them each other or bringing them near 10 cm or lower distance range. Most of new mobile devices comes with NFC support. These devices can establish connection with other NFC enabled devices or NFC tags.

These mobile devices communicates with NFC cards as writer or reader within 2 mm to 10 cm distance. NFC technology has 106 kbps to 1 Mbps data range with lower than 0.1 ms setup time. NFC's pros and cons are shown on Table 3.1

NFC can be used for identification person, making secure process, sending data between devices, password storage, parking, health cards or for more.

Table 3.1 NFC and other technologies comparison

	NFC	**RFID**	**Bluetooth**	**IrDA**
Price	Low	Common	Common	Low
Security	High	Vulnerable	Vulnerable	Vulnerable
Equipment Cost	Medium	High	Medium	Low
Maximum Range	~ 10cm	3-60 m	10-20 m	~ 10 m
Consumer Experience	Touch	Get information	Configuration needed	Easy
Data Range	106 – 1 Mbps	424 kbps	24 Mbps	~4 Mbps
Set-up time	<0.1 MS	<0.1 MS	~6 s	~0.1 MS
Topology	Peer-to-peer	Peer-to-peer	Peer-to-peer	Peer-to-peer
Standardization	ISO/IEC	ISO	Bluetooth SIG	
Network Standard	ISO 13157 etc.		IEEE 802.15.1	

3.9.1 NFC Tags

There are different types of NFC tags. NFC tags can be in different shapes like a card, button, and wristband or paper tag. These tags can contains data between 96 to 4096 bytes and can be read only or rewritable.

3.9.1.1 NFC TAG Manufacturers

There are several manufacturers for NFC tags. This companies creates different shape of tags with different technologies. Some of these tags can be read only and

some of them can be rewritable. Also these tags can come with built-in data secure protocols.

3.9.1.2 NXPs MiFare NFC Tags

There are several types of NFC tags manufacturing by Mifare, shown on Table 3.2. These cards have different specifications. These cards can be in different sizes, with different protocols or have different built-in securing mechanisms.

Table 3.2 NXP MiFare NFC cards

MIFARE	**Detail**
Classic	ISO/IEC 14443-3 Type A compliant with an NXP proprietary security protocol for authentication and ciphering.
Ultralight	Low cost cards with ISO/IEC 14443-3 Type A compliant protocol.
Ultralight C	Includes Triple DES cryptography with low-cost IC
DESFire	Comply to ISO/IEC 14443-4 Type A with a mask-ROM operating system from NXP
DESFire EV1	DESFire with AES encryption
DESFire EV2	Includes MIsmartApp, Transaction MAC, Unlimited Applications
Plus	Replacement of MIFARE Classic with certified security level (AES 128 based)
SAM AV2	Secure access module that provides the secure storage of cryptographic keys and cryptographic functions

3.9.1.3 Mifare DESFire EV 1 4K

MIFARE DESFire EV1 is based on open global standards for both RF interface and cryptographic methods. The highly secure microcontroller-based IC is certified with Common Criteria EAL4+ on both hardware and software implementation. It features an on-chip backup management system and mutual three pass authentication, allowing it to hold up to 28 different applications and 32 files per application with 4k memory.

In this study these cards have been used to be apply a more secured mechanism with these cards' AES encryption support. In application a random app id and a secure key on server are being created and write command is being generated on server with this random data. After creating message, server sends commands to mobile device and mobile device runs the command on NFC card. This creates an application on card with pre-created secure key. With this approach only one application is being used on card by the application. In all further operations server creates commands with finding cards application id from database with using cards unique secure key. With this approach tablet doesn't need to know cards application id neither secure key. Also device doesn't need to know which command should be send to card or what the response of card is. Tablet only make communication between card and server. Also remaining applications of card are free to use for further improvements or other applications.

Mifare Desfire EV1 has a restricted protocol to provide an AES encryption enabled secure system, shown on Figure 3.5(a). This protocol is confidential and only allowed people can access the protocol. In this study authorization granted from NXP. This protocol is re-written on server to ensure NFC cards are being used by personnel are true cards that are registered to system, shown as Figure 3.5(b). Both device and server calculates data with using protocol and server compares them. IF results are same server decides NFC card is registered to system. This operation is being used for specific operations.

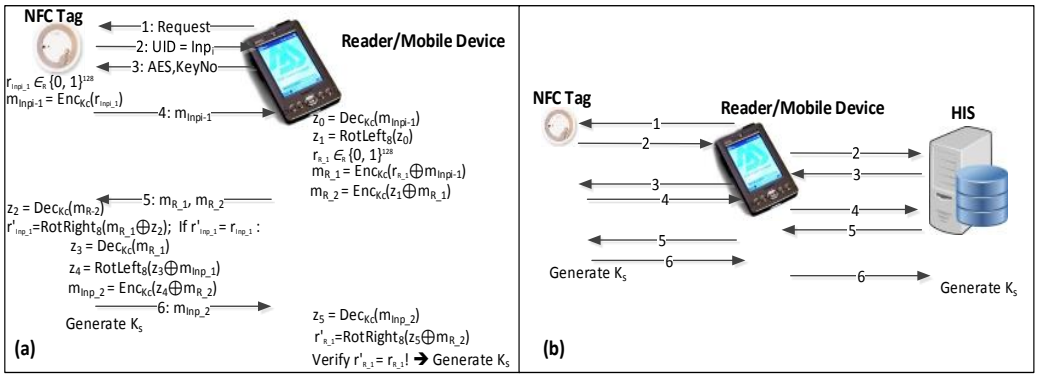

Figure 3.5 (a) NXP Protocol, (b) IMS Protocol

3.9.1.4 Commands

This commands are being used to communicate with card to authenticate, to write data to card and to read data from card which are described as detailed on Table 3.3. All commands are generated by server and tablet carries messages between tablet and server.

Table 3.3 Used commands to communicate with card

Command	**Explanation**
SELECT APPLICATION	This operation selects application with given app index.
AUTHENTICATE	This operation is being used to start authenticate selected application with known app key for DESede_CBC encryption.
READ DATA	This command is being used for reading data from authenticated application
WRITE DATA	This command is being used for writing data to authenticated application
AES AUTHENTICATE	This operation is being used to start authenticate selected application with known app key for AES_CBC encryption.
MANUFACTURING DATA	Gets touched NFC cards manufacturer to check is card convenient for application.
FORMAT PICC	Formats NFC card to SET it for before first usage
GET KEY SETTINGS	Checking for cards encryption method is AES_CBC or DESede_CBC

3.9.1.5 Responses

All command returns a response from card after a request shown on Table 3.3, responses are detailed on Table 3.4. This helps to determine is command worked right or not. These are possible responses of used commands as seen on Table 3.4.

Table 3.4 NFC card responses

Response	Explanation
OPERATION OK	Successful operation result for running command
AUTHENTICATION ERROR	If operation fails when trying to authenticate this result's returned by card
APPLICATION NOT FOUND	Trying to reach a nonsexist application will returns this error
INTEGRITY ERROR	If an unknown or damaged request send to card, this message returns by card.
ADDITIONAL FRAME	This is the second step response of authentication operation on card.

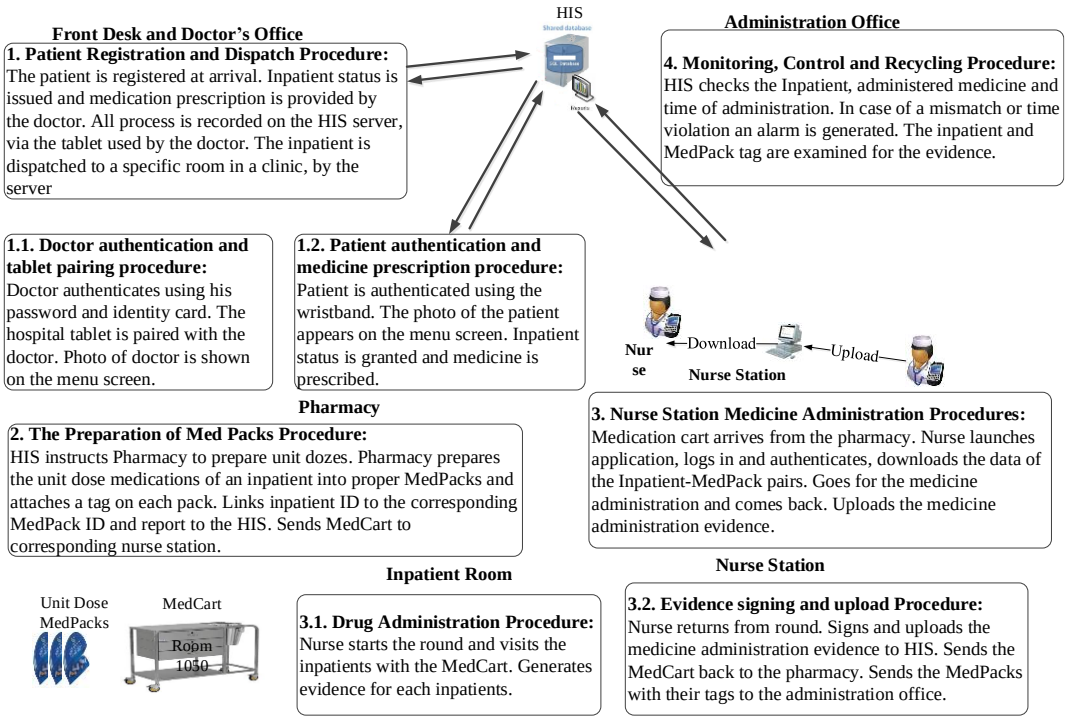

Figure 3.6 Secure IMS system

CHAPTER FOUR
SECURE MOBILE IMPATIENT IDENTIFICATION SYSTEM

The abbreviation standard used in the common notation is preserved, shown on Table 4.1 as detailed. For emphasis X is repeated.

Table 4.1 Notations

X	A doctor (Dr), nurse (Nurse) or inpatient (Inp)
i	The i^{th} X, i.e. the i^{th} doctor is Dr_i
id_X	The unique identity of X (e.g. Social Security Number)
$card\_id_{Xi}$	The extended UID of the EV1 card or wristband that belongs to i^{th} X
K_{Xi}	The n^{th} AES compatible encryption key of the tag of i^{th} X, known by the server
usr_{Xi}	The unique user name of i^{th} X
password	Password known only by its owner (doctor or nurse)
t_{R_1}	Time when the doctor enters username and password in the tablet
cpu_id_x	Tablet device ID of the hospital tablet used by X, starting at t_{R_1}
h()	A hash function based on SHA512 algorithm
E(m,k)	Uses the key k, to encrypt message m, based on the AES algorithm
D(m,k)	Uses the key k, to decrypt message m, based on the AES algorithm
N	The N^{th} AES key to be used in authentication with $card\_id_{Xi}$
t_0, t_3	Time of reading doctor's (nurse's) $card\_id_{Xi}$ by the tablet, initialized with zero
t_1	Time of doctor consultation and drug administration, initialized with zero
t_2	Time of med pack preparation and disposition, initialized with zero
K_{sXi}	Session key generated at the end of X_i's card authentication
eInpi	Generated evidence for the ith inpatient with Inpi.
SR(m)	Message m signed by private key R (SRK / PhaRK / NRK).
VU(m)	Message m verified by public key U (SUK / PhaUK / NUK).

4.1 Card Register Operation

This operation, shown on Figure 4.1, is used to introduce a new card to system. Register operation gets card UID and writes it to system database with a random app id, key settings and register time in epoch time format. Before registering card to system, server creates required commands and runs them on card via tablet. If operation is successful, card data is being write to system database. Card registration can be done only if master key of card is known by server. If master key is not known it is not possible to create an application on card or format card. So after registration master key is changed with unique one to prevent possible stealing. This operation can be performed by IT or Information Desk.

Steps:

1. Tablet gets card UID (**card_uid**)
2. Tablet generates a random file no between 1 and 32 (**rnd_file_no**) and application id between 1 and 28 (**rnd_app_id**)
3. Tablet generates a random key with 16kb size (**key0**)
4. Server sends select application 0 command and tablet redirects it to card
5. Server sends authentication command with using default master key of card
6. Server sends format card command
7. Server sends create application command with using **rnd_app_id**
8. Server sends select application command with using **rnd_app_id**
9. Server sends create data file command with using **rnd_file_no**
10. Server sends authentication command with using default master key of card
11. Server sends write command using current date time data in epoch format (**dt**)
12. Server sends change 0^{th} default key with **key0** command
13. Server sends authentication command with using **key0**
14. Server sends change 1^{st} default key with **key0** command
15. Server sends authentication command with using **key0**
16. Server sends read **rdn_file_no** file on **rnd_app_id** app command (**_dt**)

17. Tablet compares **_dt** with **dt** and sends server **card_uid**, **rnd_file_no**, **rnd_app_id**, **key0** and **dt** to be saved as new card registration data.

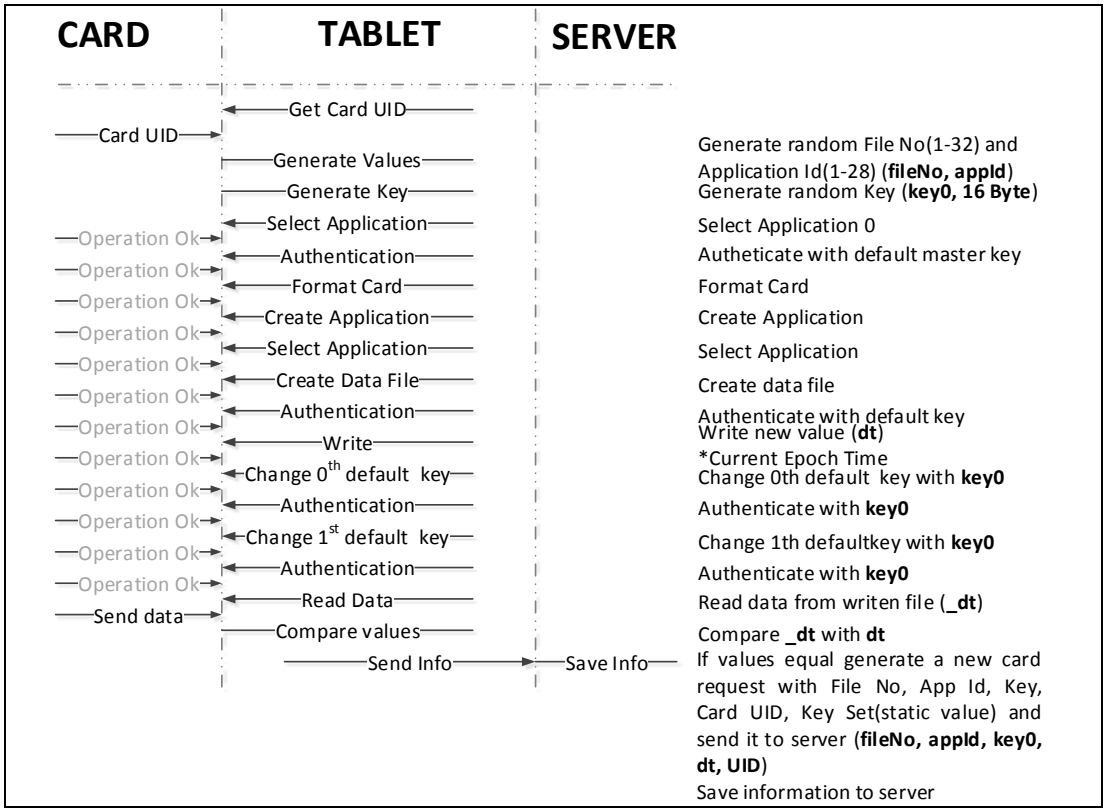

Figure 4.1 Card register operation

4.2 First Verification

Before authenticating system with users NFC card application device waits from user to enter his user name and password. When user logins his credentials and press login button, tablet concatenates user_name(user name), cpu_id(tablets cpu id) and t_{start}(request time) and encrypt this data with user_password(user password). Encryption method is AES in this state. Tablet sends encrypted value ('cpu_id'+ t_{start} +'card_id') to server.

When server gets the encrypted data try to decrypt it with user's password. Server gets device id from decrypted message and checks it. If there is not any mismatch server gets t_{start} parameter and logs it to use for next step of verification.

At next step server gets card_id (users NFC card id) and concatenates it with (‘cpu_id’+ t_{start} +’card_id’). Server hashes this data with SHA512 algorithm and encrypts it with RSA algorithm using unique public key of tablets.

Server sends new message to tablet which is hashed with SHA512 algorithm. Tablet gets response from server and try to create same data with response using information that is already stored on itself with following same way server did. If tablets created data is same with servers response next step is being initiated.

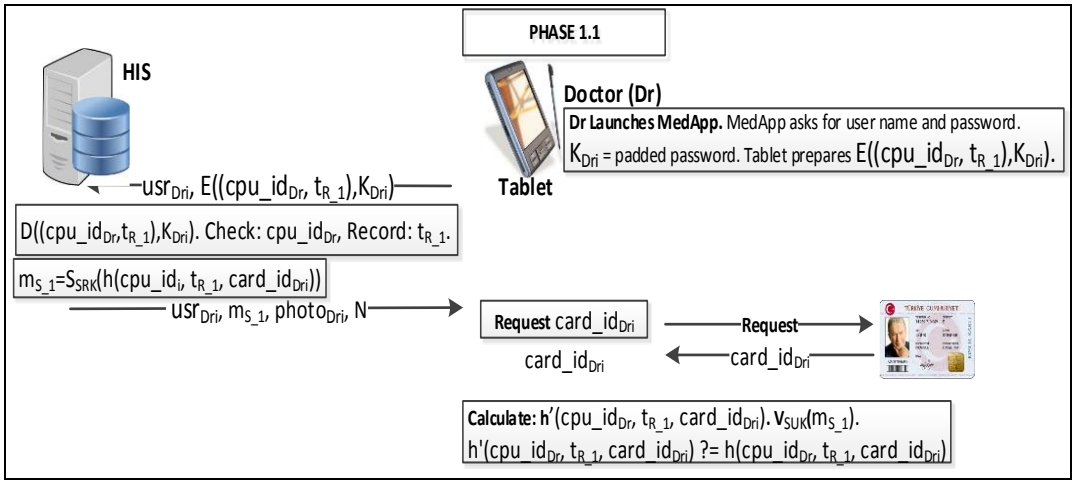

Figure 4.2 First verification operation for doctor

4.3 Card Time Authentication Operation

This is a common operation and being used for all apps to authenticate application with using users NFC card on tablet, shown on Figure 4.3. Before user logon to app, nurse made a treatment, doctor made a consultation or pharmacy load drug to med-box, authentication operation must be done. Authentication operation logs the device id, time and operation type to database and writes time info to NFC card to log all activities before starting. This helps to ensure operations are done correctly and in exact time which is logged to server. Also this helps to check all activities from database log.

There are different operations that uses Authentication Operation

a. User Login
    I. Doctor
    II. Nurse
    III. Information Desk
    IV. Registration Desk
    V. Pharmacist
b. Patient Consultation by Doctor at his Office
c. Treatment Start and Finish by Nurse
d. Medbox Drug Usage by Nurse
e. Medbox Preparation by Pharmacist

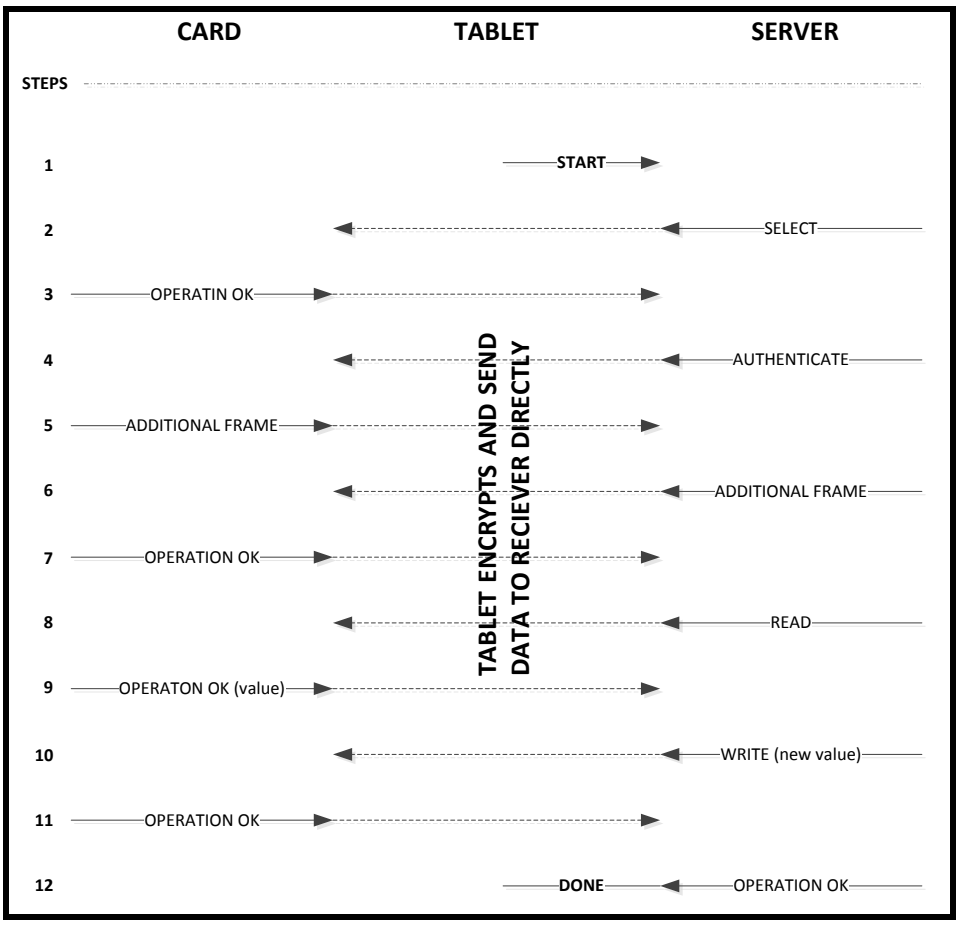

Figure 4.3 Card authentication operation

Step 1: User starts authentication operation from mobile devices. Card touch screen appears and user touches his/her card to mobile devices. After this step mobile device sends Start command to IMS-Server with card UID.

Step 2**:** IMS-Server gets File No and App No from database with given card UID and generates Select command. IMS-Server sends command and waits for Operation Ok response.

Step 3: Card operates Select Command with given AppId and Key. Card generates Operation OK Command and sends it.

Step 4: IMS-Server generates Authenticate request and sends it.

Step 5: Card operates Authenticate Command. After successful operation Card generates Additional Frame request and sends it back.

Step 6: IMS-Server generates Additional Frame request from cards response and sends it. At this operation manufacturers protocol is being used.

Step 7: Card operates Additional Frame command and sends Operation OK response.

Step 8: IMS-Server gets response and try to generate same response with already stored card data. If response and generated data is equal server sends Read command.

Step 9: Card operates Read command on selected app and file. Then sends data to IMS-Server with Operation Ok result.

Step 10: IMS-Server gets value that is stored on card and compares it with value, which stored on database. If values are equal server generates new value and sends Write command with new value.

Step 11: Card operates Write command and writes new value to file. If write operation is successful card generates Operation Ok command and sends it.

Step 12: If write operation is successful IMS-Server updates cards value on database and sends Operation OK command to mobile device.

Step 13: Authentication Operation is done.

4.4 Doctor Consultation

As shown on Figure 4.4, doctor starts his application, shown on Figure 4.5, to make consultation to his patient, at time tR_1. Before authenticating with card, doctor enters his credentials as his user name usrDri and password. Doctor's password being encrypted by device using AES encryption algorithm with a pre-shared key KDri. Device concatenates cpu_idDr with tR_1 and encrypts it, then sends it to HIS with usrDri as a request. Server gets request from tablet with usrDri and uses usrDri to find decryption key KDri from database to decrypt encrypted message, which is pre-loaded at register time. Server decrypts message and access the cpu_idDr, to check is tablet registered to system, tablet is authenticated to use with usrDri, is usrDri using another tablet already and the tablet is available to use. If server check control passes, the server concatenates cpu_idDr, tR_1 and card_idDri and hashes them to mark usrDri is using cpu_idDr. Following operation of hashing is signing hashed data with private key of server SRK. Then server sends signed data tablet to with doctor's photo and nonce N.

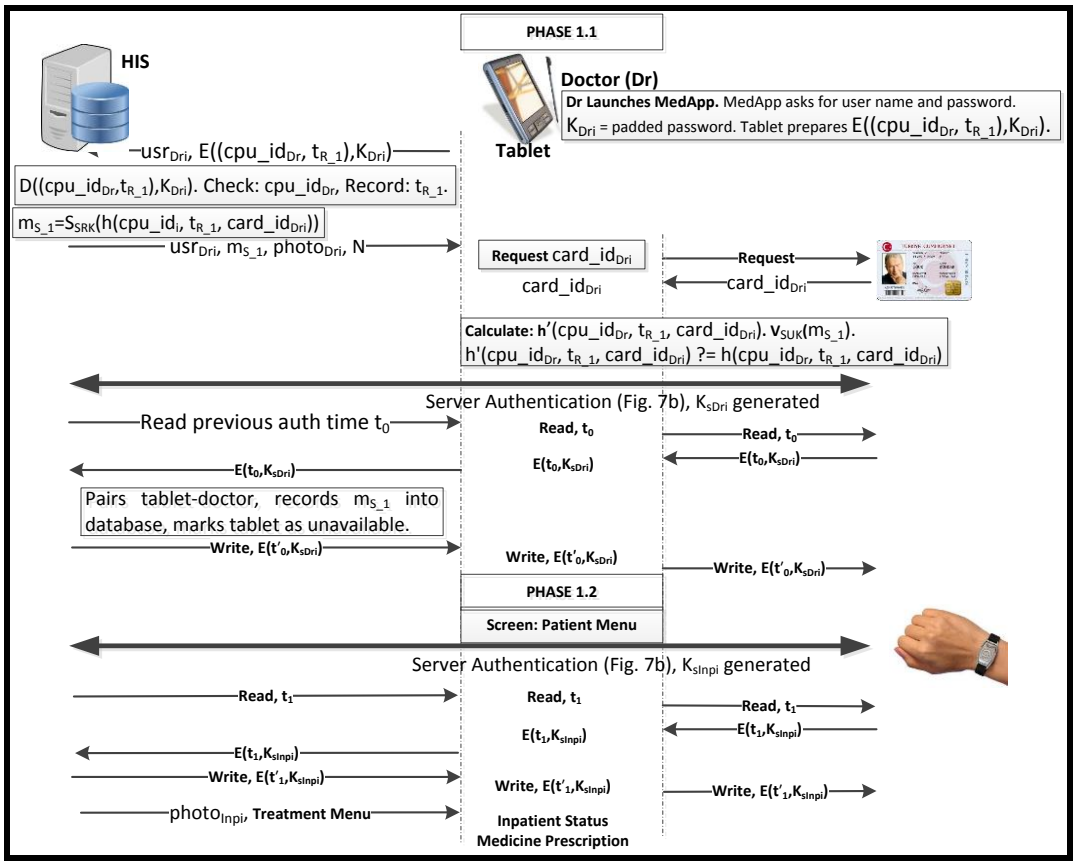

Figure 4.4 Time authentication operation

According to response, tablet knows it is planned to be use by doctor usrDri. Signed response is being decrypted with severs public key (VSUK(mS_1)). After decryption and checking, tablet asks to doctor touch his identification NFC card. Tablet gets card_idDri from doctor's card and compares hashed value with using it to verify correction. As described Figure 4.4, tablet behaves as a messenger between doctor and HIS to complete authentication of doctor, therefore session key K_{sDri} has been generated. Tablet reads previous authentication time data t_0 from card's memory with authenticating on card with using K_{sDri}. The server checks the time value t_0 and updates it with new time value t'_0 to doctor's card, as an evidence to operation is completed. At the end of successful authentication operation with using doctor's card and credentials, doctor has entered right credentials, used right NFC card, has authenticated to server and doctor's tablet paired with server successfully. After Phase 1.1, the doctor application shows doctor's name and photo on tablet and enables "Accept Patients" button on screen to apply new patient acceptances.

On Figure 4.4, Phase 1.2 comprises to giving doctor necessary rights to accessing patient status and medicine prescription. Patient touches wristband to tablet shortly. Tablet gets $card\_id_{Inpi}$ from wristband and starts authentication process, shown at Figure 4.4. Afterward, last doctor inspection time t_1 is checked on card, with using session key K_{sInpi}. If patient has never visited by a doctor, time value can be zero, as initialization value. After inspection, t'_1 time value is written to patient's wristband, to proof of the visit. In this step, doctor can see patient's information and photo on tablets screen, thence doctor can affirm patient-wristband consistence. If patient photo is missing or there is a difference, doctor can take photo of the patient with using tablet's camera.

At next step, doctor make his treatment to patient and transacts patient status and prescribes the medicine. In due time, patient is dispatched to related clinic and gathered patient data is send to HIS. Subsequently, the HIS reserves a room to patient and declares the pharmacy with prescription info to make ready med packs with given information of unit dose, ending phase 1.

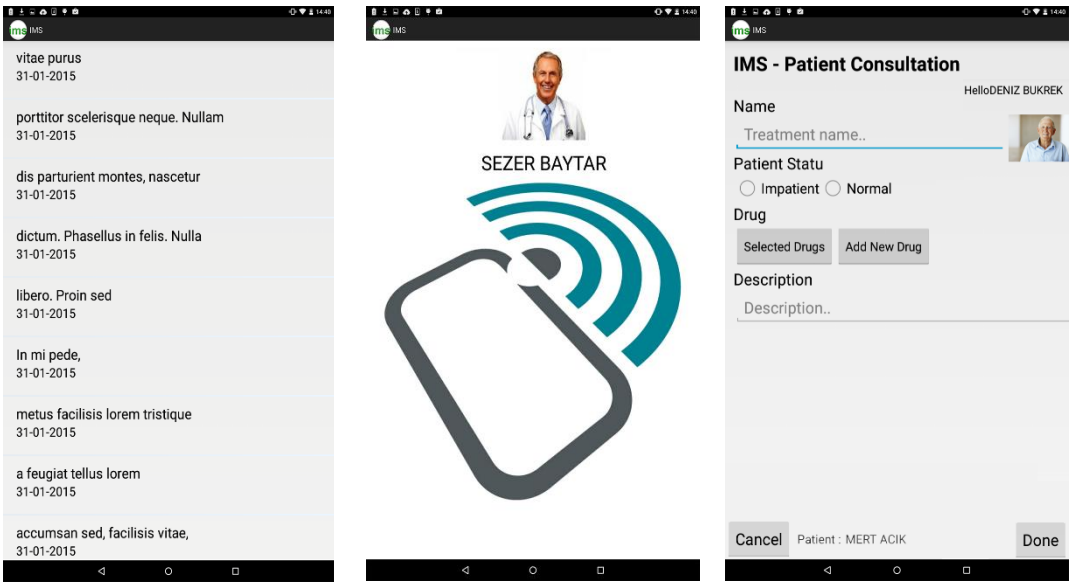

Figure 4.5 Doctor app

4.5 Med Pack Preparation

Pharmacy prepares med box with adding tags to them, after getting orders from the server, using an automatic medicine dispenser. In authentication step, device checks time data t2 on card if it is the initialization value or not. If it is the correct value, time of packaging t'2 value is being written to card's memory. At next step, card_idMPInpi is linked to inpatient card_idInpi. Med pack is put in to suitable drawer of the car, which has an NFC identification tag on it. System organizes the putting med packs to same drawer, which are belong to inpatients are in same room. After filling, HIS sends information to canalize med cart to clinic.

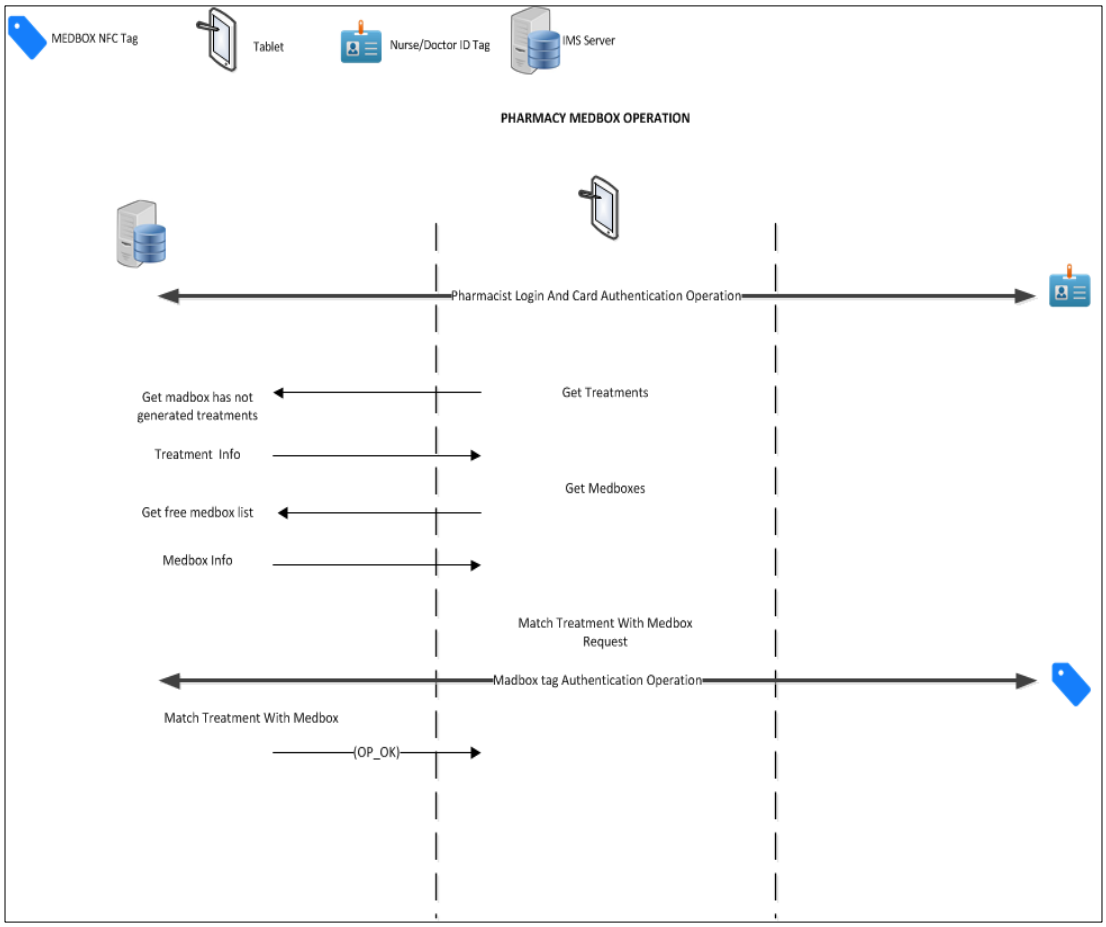

Figure 4.6 Med pack preparation

4.6 Nurse Treatment

Nurse authentication and pairing the nurse's tablet with HIS has exactly the same steps with doctors. Pairing time t3 is being written to nurse's NFC card. End of the pairing and authentication, application shows nurse's photo on the screen and menu. Before to start consultation, nurse touches the card on med cart first, then touches the card on in inpatients door, for each one. Confirmation of on time drug administration is provided by time values of med cart and patient door card read times. The NFC tags on door plate determines the inpatients, who are going to be treated. Hereafter, identification the patient and administrating inpatients drug is take place, in offline or online mode.

4.6.1 Online Procedure

In this mode nurse device is connected to wireless connection of the HIS to stay online, throughout the drug administration. In online mode, all authentication operations are follows steps as shown in Figure 4.7**.** First step of the medication is authenticating the inpatient, as shown in Figure 4.4. Tablet reads time data t'1 from inpatients wristband, which is last doctor visit or nurse treatment time. t'1 is being checked and if it matches, new time t''1 is being write to wristband, using session key KsInpi. Next step, nurse controls patient photo on screen, then opens the inpatients drawer which has been already reserved to inpatient. Nurse touches the med packs tag to locate correct drug to administrate it. After authenticating med pack with server, arranging time t'2 is being read and checked by device. If time matches, medication time t'' is being written to med pack's tag. Meanwhile, t''1 is being written on med pack tag to make additional confirmation of medication. Following, application informs nurse, and nurse administers medication. At last, to add last cross-evidence, nurse touches the inpatient's wristband. After authentication, t''2 time data is written to inpatient's tag. Hereby, multi cross-evidences are exist on both of inpatient and med pack tags. Application concatenates (card_idInpi, card_idMPInpi, t1, t′1, t′′1, t′2, t′′2) into a package. Nurse enters her password, and then private key of nurse's NRK is being formed. Next step, package is signed with

using RSA, to create an evidence eInpi = SNRK (card_idInpi, card_idMPInpi, t1, t′1, t″1, t′2, t″2). Signed data is being send to HIS server as soon as possible.

4.6.2 Offline Procedure

In possibility of losing network connection, because of network or power failure, nurse application has to work properly, to let nurse continue to treat inpatients. In the other hand, some hospitals may be disqualified for setting up a local wireless network. In the offline-mode, pairing and authenticating with registered tablet is very important, because all data is downloaded to tablet, includes sensible patient data and AES encryption keys. Thence, off-line procedures must be built clearly and related data must be downloaded from nurse station. Afterward, all steps proceeds same as the online mode, unless the authentication step shown at Figure 4.4 2(a) that works at offline mode, storing and running commands on tablet until nurse returns to her station. When nurse reaches to station, all stored data on tablet is being transfer to server for each round.

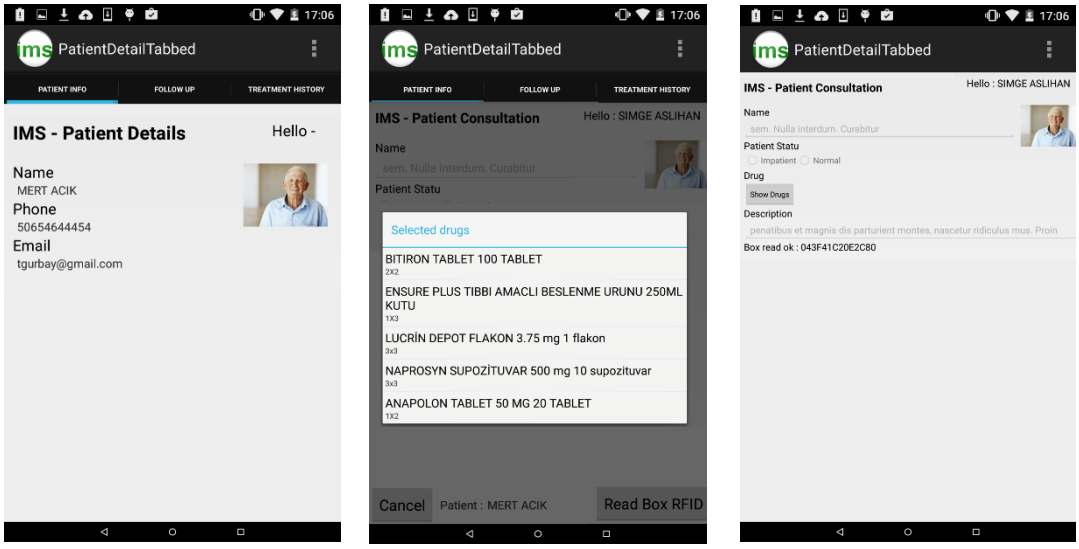

Figure 4.7 Nurse app

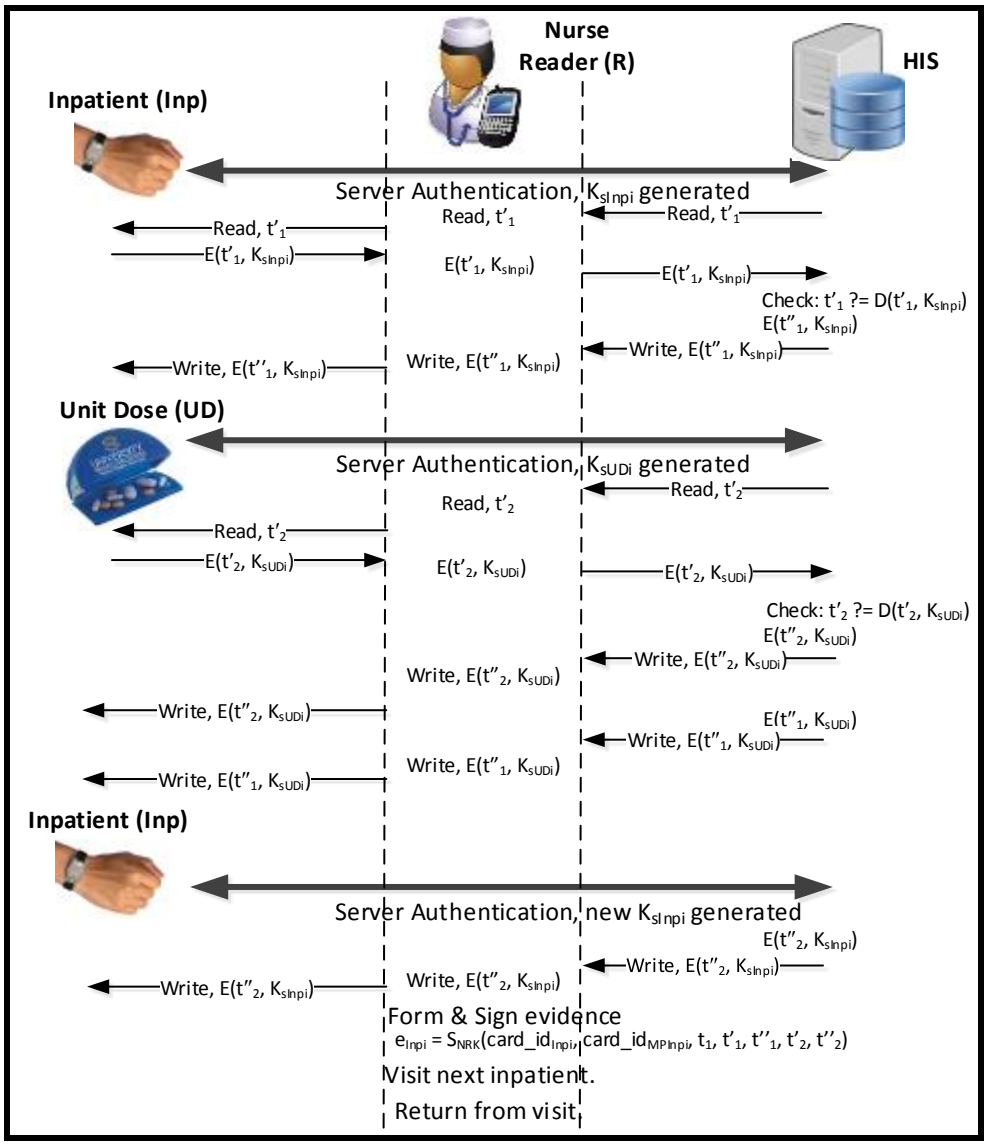

Figure 4.8 Nurse treatment operation

To confirm treatment, nurse is being forced to press a button after completing round and uploading data. After confirmation, all memories are going to be clean by application, including primary and secondary memories. This prevents getting sensible data from device's memory by unauthorized person. Finally, used med packs are being returned to administration office by nurse, to control and recycle. En of the round, med card is being redirect to the pharmacy.

4.7 Patient Register on Patient Information Desk

Patient Information app, shown on Figure 4.9, is using for meeting patient before the admission to hospital. PI User registers patient to system with getting personal info from patient. After getting required information PI user assigns a NFC tag to patient.

Required Patient Information

- Name and Surname
- Address, email and phone
- Patient picture

Registered patient can be identified with his NFC tag by any authorized personnel. Registered new patients are oriented to doctors' office immediately.

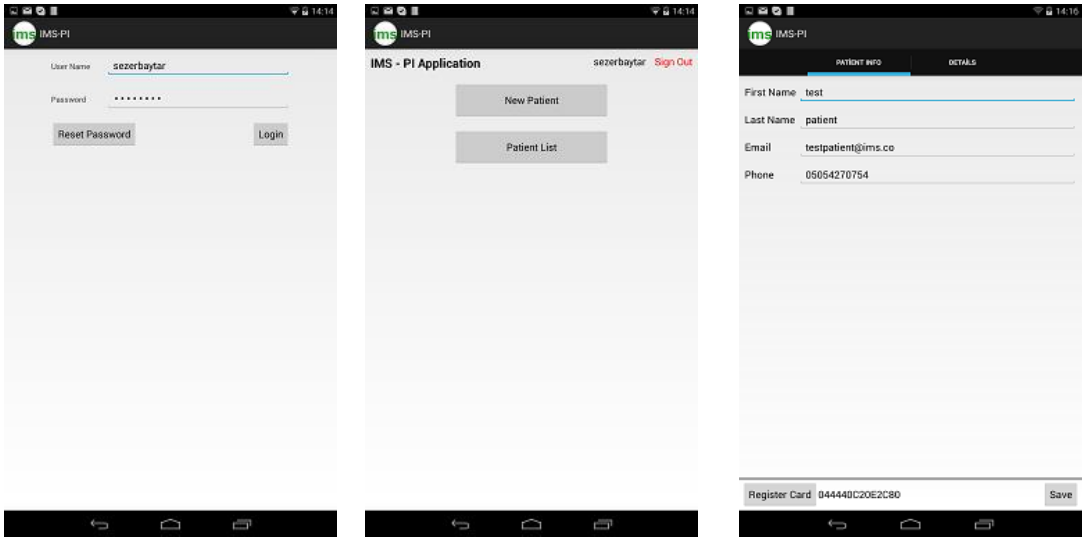

Figure 4.9 PI app

4.8 User Registration by IT

IT app is using to register new hospital personnel to system with assigning a NFC card. Also IT user can update user info or delete users with IT app.

CHAPTER FIVE
SYSTEM EVALUATION AND RESULTS

5.1 Mobile Application Testing

Most research on mobile application testing has focused on mobile usability testing. Advances Ubiquitous mobile health applications provide services to people anywhere (Nkosi & Mekuria, 2010), anytime using broadband and wireless communications, as well as mobile computing devices (Varshney, 2014). Software test methods and tools provides to ensure quality in functions, behaviors, performance, and quality of service, as well as features, such as connectivity, security, and privacy.

5.1.1 Quality-of-Service Testing

The QoS requirements for mobile applications include software performance, reliability, availability, scalability, and loading speed. Rabeb Mizouni and his colleagues evaluated the Web service performance. Their focused QoS parameters included response time, availability, throughput, and scalability. Many researchers (Anand, Naik, Harrold, & Yang, 2012) have proposed lightweight frameworks to evaluation of QoS of Web Services on mobile devices (Gao, Bai, Tsai, & Uehara, 2014). However, very few works have tackled the evaluation of QoS of mobile services. In this study, the following QoS parameters are considered: Throughput, Availability, Response Time, and Scalability.

Scalability: It describes the performance of the Web Service under different load conditions (Alférez, Pelechano, Mazo, Salinesi, & Diaz, 2014). It is usually characterized by the number of responses in total, the number of successful vs. erroneous responses, the average of the execution time, and finally the average of server time (Mizouni, Serhani, Dssouli, Benharref, & Taleb, 2011).

Availability: represents the probability that a mobile service is accessible.

Throughput: represents the number of requests processed per seconds.

Response Time: represents the time needed between issuing a request and getting its response. Response Time is the amount of time the system takes to process a request after it has received one.

5.1.2 Performance Testing Objectives

Testing is used to verify that an application is able to perform under expected and peak load conditions, and that it can scale sufficiently to handle increased capacity. Performance testing objective is to verify specified system performances (e.g. response time, service availability, error rate). Application's performance can be evaluated by comparing it with the performance objectives.

5.1.2.1 Testing Tool – Jmeter

Web services are currently the most promising service-oriented computing (SOC) based technology (Papazoglou, Traverso, Dustdar, & Leymann, 2007). Effective Web service load testing is very important for evaluating the performance of service-oriented applications. The load testing results can significantly help service provider/developer with improving the performance of these applications. Testing tool is a program to do various automated testing tasks. Nowadays testing is done with the help of various testing tools (Bertolino, 2007), such as LoadUI, Apache JMeter, and IBM Rational Performance Tester. Apache JMeter is the most flexible and interactive testing tool. Apache JMeter is also a Web service-testing tool that supplies the capability of Web service load testing. Load testing used for measuring the performance of the web services. JMeter is a Java platform application. It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

5.1.3 Experiments

In this section, experimental setup described as well as the mobile services that have been implemented for testing purpose. In investigating the quality of service delivered has been interested by Mobile Web Services and the impact on the performance of the hosting device. The QoS of these web services are identified and evaluated; such as response time, availability, and throughput.

5.1.4 Test Environment

Web Service test steps and requests created on SoapUI and tested for getting successful response. After being persuaded load test steps have been created on JMeter from SoapUI requests. Test scenarios don't include only one function call. Tests include whole flow of operations like login, consultation or treatment. So each load test includes at least more than 1 web service call.

- Aplication server hosted on a PC which has specifications below,
    - I7 8 Cores 3.4 Ghz processor
    - 16 GB RAM
    - 120 GB SSD
    - Windows server 2012 64 Bit
- Web Services deployed on a SSL Enabled Tomcat Server 7 instance with Axis 2
- MySql used as data provider to web service
    - There were 200 patients, 100 drugs, 1k logging messages on the database
- University routers has been used (Eduroam)
- SOAP Messaging protocol used for carrying json objects.

5.1.5 Test Scenarios

We have assumed that there are no restrictions such as firewall, connection problems or closed ports. Test made on a well working environment. Server had real

IP. So server could be accessible from any computer that has Internet connection. Thus that can assume load testing was not made on local network.

5.1.5.1 Login Test

There are 7-web service calls on this test.

1. GetFirstVerification
2. Card Authenticate Operation (6 calls)

5.1.5.2 Nurse Treatment

There are totally 17 function calls on this test. 2x Card Authenticate Operation and 5 other functions

1. PTGetPatientDetailByCardId
2. GetPatientInfoAndTreatments
3. GetTreatmentHistory
4. CheckAcknowledgement (patient)
5. Medbox card Authenticate Operation (6 calls)
6. CheckAcknowledgement (medbox)
7. Patient card Authenticate Operation (6 calls)

5.1.5.3 Doctor Treatment Operation

There are totally 5 function calls on this test.

1. PTGetPatientDetailByCardId
2. PatientTreatmentList
3. GetDrugList
4. PatientTreatmentAdd
5. PatientTreatmentList

5.1.5.4 Pharmacy Operation

There are totally 5 function calls on this test.

1. Card Authenticate Operation (6 calls)
2. MPGetListOfMedPacks
3. MPGetListOfMedPackNotGeneratedTreatments
4. MPGenerateTreatmentMedPack

5.1.5.5 Card Authenticate Operation

There are 6 steps to authenticate user with card

1. Start Authentication
2. Select Application
3. Authenticate
4. Additional Frame
5. Read Old Value
6. Write New Value

5.2 Service Performance Test Results

We have tested web services sending up to 100 requests in a second. Tests show us server can handle easily handle such a load like this

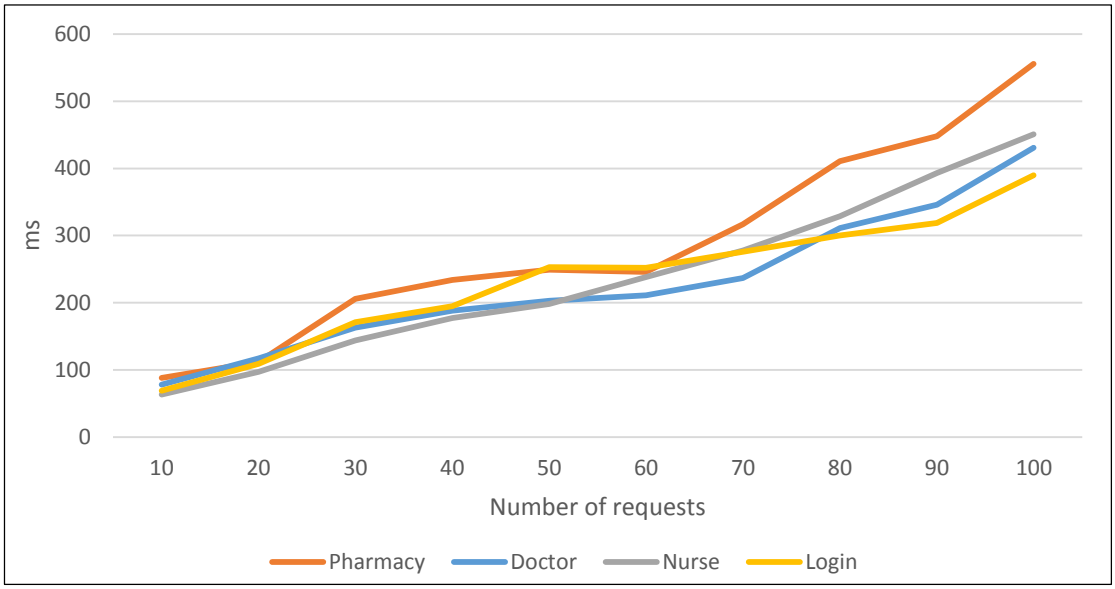

Figure 5.1 Average response time

This operation shows that response times are rising proportionally to request count. But maximum response time is fair enough for 100 requests. Upgrading server capacity or adding options like load balance can decrease this response time to ~100ms.

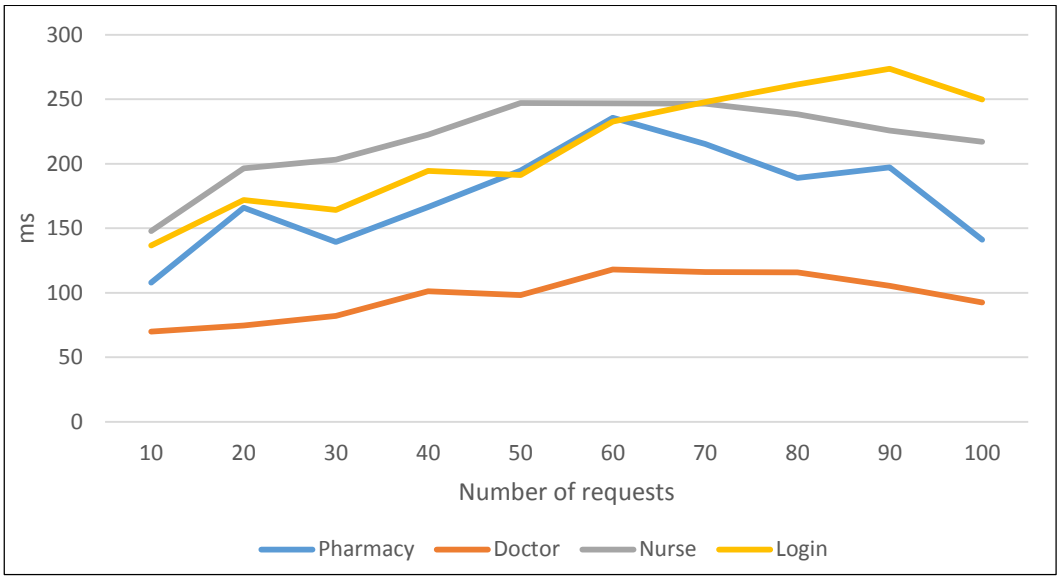

Figure 5.2 Throughput

This operation shows that throughput measured as number of request handled per second. The results show that the throughput of IMS-Services is relatively high and increases when the number of generated request increases. However, it declines when it reaches around 60 simultaneous request sent to IMS-services under test.

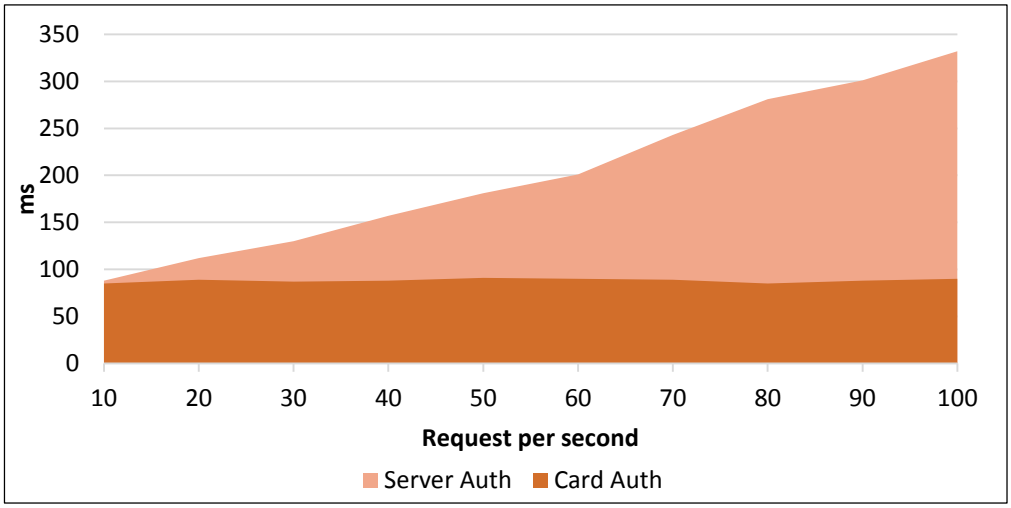

Figure 5.3 Authentication average response time

This test is specially created to show how affected response time after changing authentication protocol *card-tablet* to *card-tablet-server*. As mentioned before authentication operation contains six function calls. It takes approximately 100 ms for 1 request on normal protocol. Same as normal protocol, authenticating from card to server take approximately 100ms for 1 request. But response time increases with request count. So it shows that card-to-server protocol leads to ~200ms time loss for authentication operation. This is an acceptable result for security concern.

5.3 Security Test Results

Security tests are done with enabling/disabling Tomcat SSL support. To enable SSL support, a certificate created with Java key tool and self-signed it with OpenSSL. Port 8080 dedicated for http and 8443 dedicated for https connection.

With this test, web server sniffed with WireShark, shown as Figure 5.5 and Figure 5.6. Results show that, it is possible to get data when SSL is not enabled. But when

SSL is enabled, it is not possible to get raw data. But attacks like Man in The Middle Attack are known. So some extra security option added to our web services. Body of the SOAP messages are being encrypted with AES via pre-shared 128 bit keys.

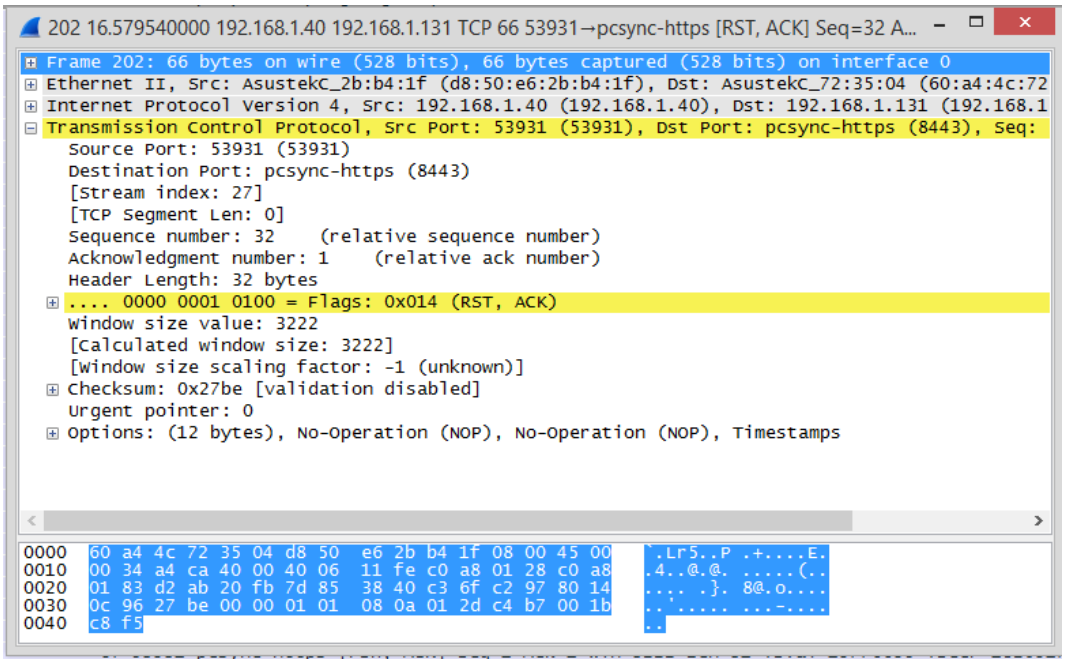

Figure 5.4 Wireshark captured packages with SSL

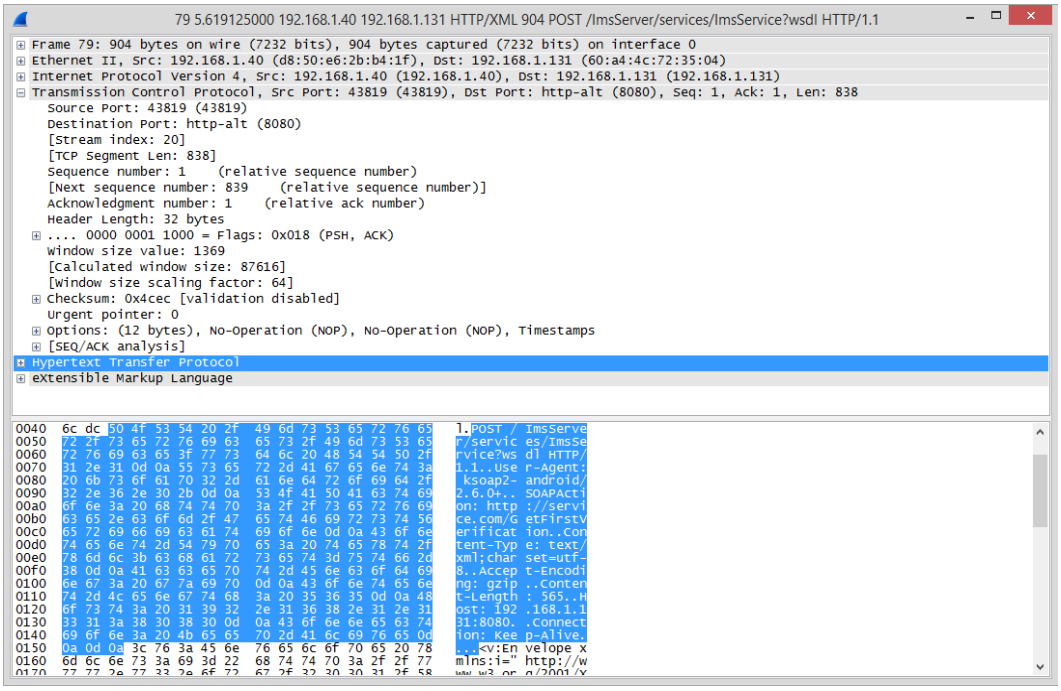

Figure 5.5 Wireshark captured packages without SSL

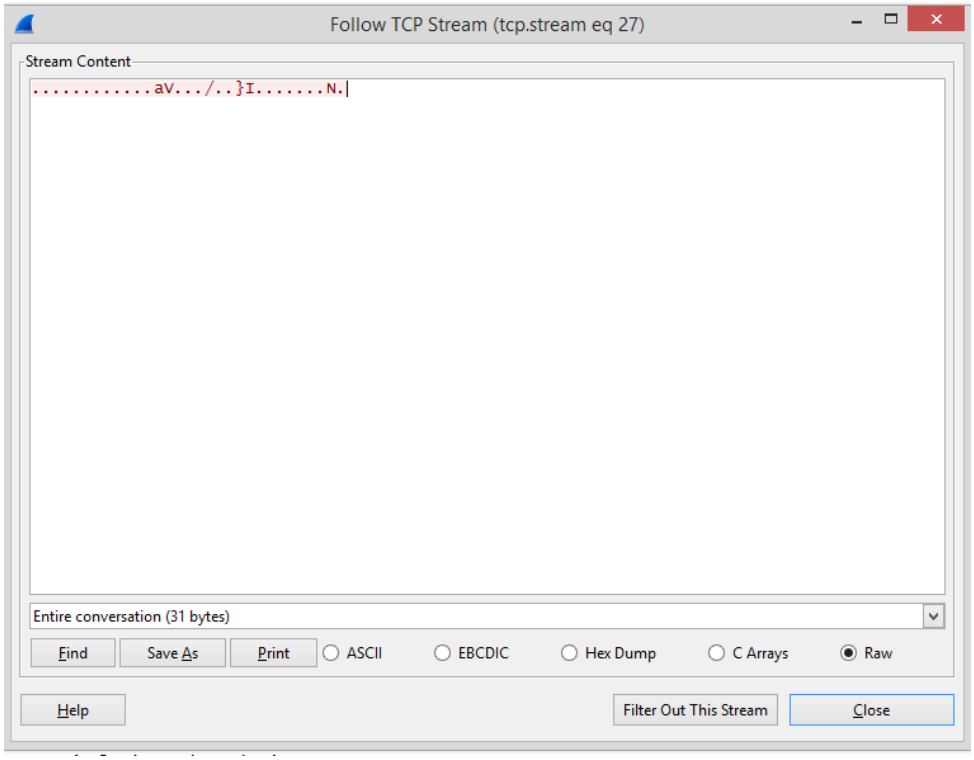

Figure 5.6 Inspection of SSL package

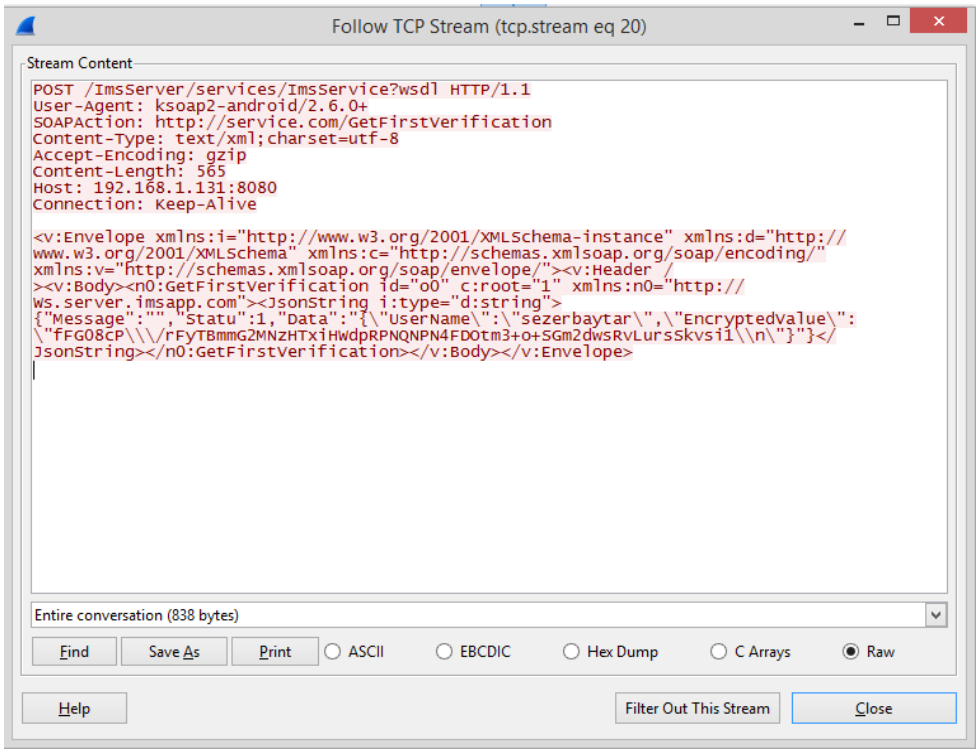

Figure 5.7 Inspection of package without SSL

Figure 5.6 shows that with SSL support-getting data is not possible without specific attacks like man in the middle. Figure 5.7 shows that it is possible to get data when SSL is not enabled. Accessing sensitive data easily from attacker prevented with extra encryption.

Figure 5.7 also shows that, sensitive data is being encrypted with an encryption method (AES). If attacker access to data over SLL/TLS, he will meet an encrypted data. In this case, attacker must find pre-shared encryption key and method do decrypt this data.

CHAPTER SIX
CONCLUSION

This thesis aimed to create a ubiquitous system for tracking an inpatient in hospital with sensors and securing patient data with using wireless and NFC technologies. This will improve the usage of drugs in right dosages and the right patients, patient safety, reducing personnel work, giving more information about patients' health status, making right treatment to patients by nurses. Patient data security is one of the most important concern; which is attained with our improvements, using different encryption methods and approaches. Beside this algorithms and methods also physical security is the other concern. To ensure a more secure system, NFC cards are used to authenticate personnel after logging in to application with their user name and password. NFC cards are not able to be copied from others. This provides accessing data without registered NFC card.

6.1 Gains

In this thesis, one of the most important idea is reducing cost of the wrong drug usages. With this work, aimed to reduce wrong drug usage to zero. Drugs are comprise the most cost of a hospital's expense. To reduce this cost, an inpatient drug administration (IDAP) developed. IDAP helps to reduce wrong drug usage with controlling it from pharmacy to inpatient. IDAP assumes, if personnel follow instructions exactly, it is possible to use exact amount of drug for each inpatient. This helps to reduce wrong drug usage to minimum, beside this, it helps to making medicine with exact drug usage.

6.2 Security and Performance

Security test results show that unauthorized access to sensitive data is not possible in normal circumstances, owing to SSL/TLS, AES encryption and RSA support. Even so, if an attacker cracks SSL and RSA encryption, he will meet encrypted

messages, that are encrypted with AES. Thus attacker cannot access decrypted messages without encryption keys

Wireless communication is able to sniff by applications. With SSL/TLS enabled protocol, sniffed traffic cannot be readable. But there are different types of attacks such as Man in the Middle. In this thesis, an extra security is added with encrypting body of messages. So if an intruder sniff and decrypt data, he will meet an extra encrypted message. This prevents unauthorized data access with known methods.

Android applications are able to decompile by third party applications. Thus it is possible to get encryption keys and algorithms from cracked devices. This is a big security problem if device is stolen and IT is not noticed immediately.

Also there was another problem with cracking android devices. In normal state, NFC commands should be created by device and communication should be between device and NFC cards. But easily cracking of android devices is directed me to implement a more secure way. To achieve a more secure and reliable way, the device was eliminated as a primary contractor with NFC cards. In this solution all commands are being created by server and all response are sending to server directly as encrypted. In this method tablets are acting as a messenger. They don't know the ingredients of the messages. All this messages are encrypted. Secure mechanism is coming from Mifare Desfire EV1s. This cards have built in 3DES AES encryption mechanisms. All mechanism implemented on the server with contacting NXP. So with knowing key of the card, it is possible to create a communication between server and NFC card.

Performance test results show that it is possible to handle all requests, a hospital hosts 100 personnel, with 100 requests in one second. Even if all personnel sends request to server, it is possible to handle all request in real time, with 0,1 second response time. It is possible to handle more request in parallel, still it is possible to handle more request by changing server with a powerful server or adding load balancer option.

6.3 Unit Costs

There are different assets that effect costs. Estimated costs of a hospital with 40 personnel, 3 floor and 200 inpatient room is shown on Table 6.1, total cost is $11.050.

There is no software cost, because the system developed to be able to work on Linux system and open source software, such as Tomcat or MySQL.

Table 6.1 Unit costs

	Units	Unit Price	Total Price
Personnel NFC cards	100	$0,6	$60
Personnel Tablet Devices	40	$200	$8.000
Hospital Wireless Switch	6	$25	$150
IMS Server and Database	1	$2.500	$2.500
NFC Bracelets For Patients	200	$1,50	$300
NFC Tags for Med-Boxes	20	$2	$40
			$11.050

6.4 Mobile Application

In this thesis a mobile application for android tablets is developed. Android Studio is used to develop mobile application. Main language is Java for this application. There are different applications for different personnel. For maintenance, a common library for all applications was created. This library includes core functions such as web services calls, logging, NFC support etc. Android applications communicate server via web services. Messages are encrypted, so the application able to encrypt/decrypt messages via pre-shared keys o algorithms. It is possible to decompile android applications. To prevent accessing secret keys, Android Keystore is used and all keys are being stored as encrypted.

6.5 NFC

NFC tags are used for tagging inpatient with wristband, med-box identifying, and drug package tagging and identifying staff with personnel cards. All activities from tablet devices are need to contact with related NFC cards. This is helping to confirm actions that done in right time by right personnel.

6.6 Web Services

Web services developed with Eclipse EE. Axis2 technology was used to develop a SOAP based web service. Services run on Tomcat 7.0 Standalone server. SOAP body is consist of JSON message. This messages are encrypted with AES. AES keys are shared only with registered devices. Also web services are working on SSL/TLS enabled protocol. All SSL/TLS keys are generated with OpenSSL.

REFERENCES

Agrawal, P., & Bhuraria, S. (2012). Near field communication. *SETLabs Bridfings*, *10*(1), 67-74.

Alférez, G. H., Pelechano, V., Mazo, R., Salinesi, C., & Diaz, D. (2014). Dynamic adaptation of service compositions with variability models. *Journal of Systems and Software*, *91*, 24-47.

Anand, S., Naik, M., Harrold, M. J., & Yang, H. (2012, November). Automated concolic testing of smartphone apps. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering* (59). ACM.

Bates, D. W., Cullen, D. J., Laird, N., Petersen, L. A., Small, S. D., Servi, D., & Edmondson, A. (1995). Incidence of adverse drug events and potential adverse drug events: implications for prevention. *Jama*, *274*(1), 29-34.

Benelli, G., Pozzebon, A., & Parrino, S. (2010). *RFID Applications for Sanitary Environments*. INTECH Open Access Publisher.

Benson, T. (2012). Principles of health interoperability HL7 and SNOMED. *Springer Science & Business Media.*

Bertolino, A. (2007, May). Software testing research: Achievements, challenges, dreams. In *2007 Future of Software Engineering* (85-103). IEEE Computer Society.

Chen, Y. Y., Huang, D. C., Tsai, M. L., & Jan, J. K. (2012). A design of tamper resistant prescription RFID access control system. *Journal of Medical Systems*, *36*(5), 2795-2801

Coskun, V., Ozdenizci, B., & Ok, K. (2013). A survey on near field communication (NFC) technology. *Wireless Personal Communications*, *71*(3), 2259-2294.

Cullen, D. J., Sweitzer, B. J., Bates, D. W., Burdick, E., Edmondson, A., & Leape, L. L. (1997). Preventable adverse drug events in hospitalized patients: a comparative study of intensive care and general care units. *Critical Care Medicine*, *25*(8), 1289-1297.

Dolin, R. H., Alschuler, L., Beebe, C., Biron, P. V., Boyer, S. L., Essin, D., & Mattison, J. E. (2001). The HL7 clinical document architecture. *Journal of The American Medical Informatics Association*, *8*(6), 552-569.

Fernando, B., McKinstry, B., & Sheikh, A. (2006). Reducing medication-related adverse events in elderly patients. *Reviews in Clinical Gerontology*, *16*(01), 79-87.

Frenzel, L. E. (2006). NFC makes great progress in the wireless world. *Electronic Design: For Engineers and Engineering Managers*, (20), 36-37.

Gao, J., Bai, X., Tsai, W. T., & Uehara, T. (2014). Mobile application testing: a tutorial. *Computer*, (2), 46-55.

Hohberger, C., Davis, R., Briggs, L., Gutierrez, A., & Veeramani, D. (2012). Applying radio-frequency identification (RFID) technology in transfusion medicine. *Biologicals*, *40*(3), 209-213.

Hoque, M. (2010). *Protecting privacy and ensuring security of RFID systems using private authentication protocols.* Ms. Thesis, Marquette University, Milwaukee.

Huang, H. H., & Ku, C. Y. (2009). A RFID grouping proof protocol for medication safety of inpatient. *Journal of Medical Systems*, *33*(6), 467-474

Hutter, M., Schmidt, J. M., & Plos, T. (2008). RFID and its vulnerability to faults. *Cryptographic Hardware and Embedded Systems–CHES 2008* (363-379). Springer Berlin Heidelberg.

Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., & Qiu, D. (2014). Security of the internet of things: perspectives and challenges. *Wireless Networks*, *20*(8), 2481-2501.

Keohane, C. A., Bane, A. D., Featherstone, E., Hayes, J., Woolf, S., Hurley, A., & Poon, E. G. (2008). Quantifying nursing workflow in medication administration. *Journal of Nursing Administration*, *38*(1), 19-26.

Landman, A., Neri, P. M., Robertson, A., McEvoy, D., Dinsmore, M., Sweet, M., & Miles, S. (2014). Efficiency and usability of a near field communication-enabled tablet for medication administration. *JMIR mHealth and uHealth*, *2*(2).

Mizouni, R., Serhani, M. A., Dssouli, R., Benharref, A., & Taleb, I. (2011, September). Performance evaluation of mobile web services. In *Web Services (ECOWS), 2011 Ninth IEEE European Conference on* (184-191). IEEE.

Ngai, E. W., Poon, J. K. L., Suk, F. F. C., & Ng, C. C. (2009). Design of an RFID-based healthcare management system using an information system design theory. *Information Systems Frontiers*, *11*(4), 405-417

Nkosi, M. T., & Mekuria, F. (2010, November). Cloud computing for enhanced mobile health applications. *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on* (629-633). IEEE.

Özcanhan, M. H., Dalkılıç, G., & Utku, S. (2014). Cryptographically supported NFC tags in medication for better inpatient safety. *Journal of Medical Systems*, *38*(8), 1-15.

Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F. (2007). Service-oriented computing: State of the art and research challenges. *Computer*, (11), 38-45.

Pateriya, R. K., & Sharma, S. (2011, June). The evolution of RFID security and privacy: a research survey. *Communication Systems and Network Technologies (CSNT), 2011 International Conference on* (115-119). IEEE.

Pitler, L. R., & Bonomi, P. D. (2006). Developing an effective and compliant plan for billing clinical trials. *Journal of Oncology Practice*, *2*(6), 265

Poon, E. G., Keohane, C. A., Yoon, C. S., Ditmore, M., Bane, A., Levtzion-Korach, O., & Gandhi, T. K. (2010). Effect of bar-code technology on the safety of medication administration. *New England Journal of Medicine*, *362*(18), 1698-1707.

Rosenbaum, B. P. (2014). Radio frequency identification (RFID) in health care: privacy and security concerns limiting adoption. *Journal of Medical Systems*, *38*(3), 1-6

Sánchez, M. A., Mateos, M., Fraile, J. A., & Pizarro, D. (2012). Touch Me: a new and easier way for accessibility using Smartphones and NFC. In *Highlights on Practical Applications of Agents and Multi-Agent Systems* (pp. 307-314). Springer Berlin Heidelberg.

Sharma, S. K., Ahmed, N., & Rathinasamy, R. S. (2005). E-healthcare: A model on the offshore healthcare delivery for cost saving. *International Journal of Healthcare Technology and Management*, *6*(3), 331-351.

Shojania, K. G., Duncan, B. W., McDonald, K. M., & Wachter, R. M. (2002). Safe but sound: patient safety meets evidence-based medicine. *JAMA*, *288*(4), 508-513.

Sun, P. R., Wang, B. H., & Wu, F. (2008). A new method to guard inpatient medication safety by the implementation of RFID. *Journal of Medical Systems*, *32*(4), 327-332

Varshney, U. (2014). Mobile health: Four emerging themes of research. *Decision Support Systems*, *66*, 20-35.

Voshall, B., Piscotty, R., Lawrence, J., & Targosz, M. (2013). Barcode medication administration work-arounds: a systematic review and implications for nurse executives. *Journal of Nursing Administration*, *43*(10), 530-535.

Wei, Z., Chao, F., & Quan, Z. (2014). RFID system security overview. *Network Security Technology & Application*, *9*, 077.

Wilson, K., & Sullivan, M. (2004). Preventing medication errors with smart infusion technology. *American Journal of Health System Pharmacy*, *61*(2), 177-183.

Yu, Y. C., Hou, T. W., & Chiang, T. C. (2012). Low cost RFID real lightweight binding proof protocol for medication errors and patient safety. *Journal of Medical Systems*, *36*(2), 823-828