CeMS

Centralized Medical System

By

NC Hafiz Muhammad Hassan

Maj. Humayun Khushal

GC Mohsin Mumtaz Mirza

Submitted to the Faculty of Computer Software Engineering,
National University of Sciences and Technology, Rawalpindi in partial fulfillment for the requirements of a B.E Degree in Computer Software Engineering

June 2012

ABSTRACT

For a common person, it is very difficult to manage all medical records. It is quite possible forpeople to forget to carry all the relevant medical prescriptions or lab reports along while visiting the doctor. Paper-based medical prescriptions may get torn or lost as well. Similarly if a patient is out of country or out of city,it will be more difficult to ensure availability of medical records for analysis, in medical emergency. Moreover for conducting research, statistical analysis or surveys on medical or health related issues (diseases, death occurrences etc.); Pakistan’s current paper-based medical records keeping system is useless.

Towards this end, we built a system (consisting of a centralized web server as web service, an Android tablets application for doctors, an Android mobile application for patients and a desktop application in Java) to manage the medical records as described in Health Level 7(HL7)standard. From HL7 description, patient and prescription entities have been included in the system.

Centralized server,which includes Apache as web server, MySQL for databases and PHP for server side intelligence, stores the data models for medical prescriptions, information of patients, hospitals, clinics, doctors and system operators. Desktop Application enables registration of the system roles (doctors, patients, hospitals etc.) with server and it also performs the submission of medical prescriptions along with patient’s attachments if required. Android tablets application for doctors allows the doctors to view medical prescription of patients along with their X-rays or other images (including Digital Imaging and Communications in Medicine (DICOM) images support) on the tablet and it also encourages the doctor to prescribe from the tablet as well. Android mobile phones application for patients facilitates the patients to communicate their basic information and prescriptions information using GSM technology through sending text messages.

The developed system provides fairly efficient solution for medical records management, hence facilitating common people not to worry about the loss of their medical records. With the help of the system, government organization, private companies, educational institutes and hospitals can use patient’s prescriptions information for their own relevant purposes.

CERTIFICATE

Certified that the contents and form of project report entitled **“CeMSCentralized MedicalSystem”** submitted by 1) Hafiz Muhammad Hassan, 2) Humayun Khushal, and 3) Mohsin Mumtaz Mirza have been found satisfactory for the requirement of the degree.

Supervisor: ____________________

Major Dr Naveed Iqbal Rao

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible

ACKNOWLEDEMENTS

We are eternally grateful to Almighty Allah for bestowing us with the strength and resolve to undertake and complete the project.

We gratefully recognize the continuous supervision and motivation provided to us by our Project Supervisor MajorDr Naveed Rao from Military College of Signals (MCS). We are highly gratified to our committee members Assistant Prof. DrHammadAfzal, DrSeemabLatif, LecTabindaWaheed and Lec Bilal Rauf for their continuous and valuable suggestions, guidance, and commitment.We also thank the Brig(R) Dr AshrafMasood and Col DrFahimAriffor encouraging us to develop a solid understanding of fundamentals and background material that has served us well.We are highly thankful to all of our professors whom had been guiding and supporting us throughout our course and research work.We are grateful for their involvement.Their knowledge, guidance and training enabled us to carry out this research work.

We would like to offer our admiration to all our classmates, and our seniors who had been supporting, helping and encouraging us throughout our thesis project. We are also indebted to the MCS system administration for their help and support.

We deeply treasure the unparalleled support and tolerance that we received from our friends for their useful suggestions that helped us in completion of this project.

In the end we are also deeply obliged to our families for their never ending patience and support for our mental peace and to our parents for the strength that they gave us through their prayers.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

Changing time leaves nothing unchanged. Every dawn is opening new avenues of improvement and exploration. Stagnation and procrastination are thief of time. Gone are the days of documents and paper environment. Mostly used phrase of “Global Village” symbolizes that world is shrinking. Perhaps time is not far off, when all gadgetries will be present in finger tips of an individual. Facilitating lifestyle is no more a luxury, rather it is necessity. If you are unable to keep pace with new technological era, then by merely relying on just few years old technology, you are perhaps living in Stone Age. As average human life is reducing, quest for exploration is multiplying.

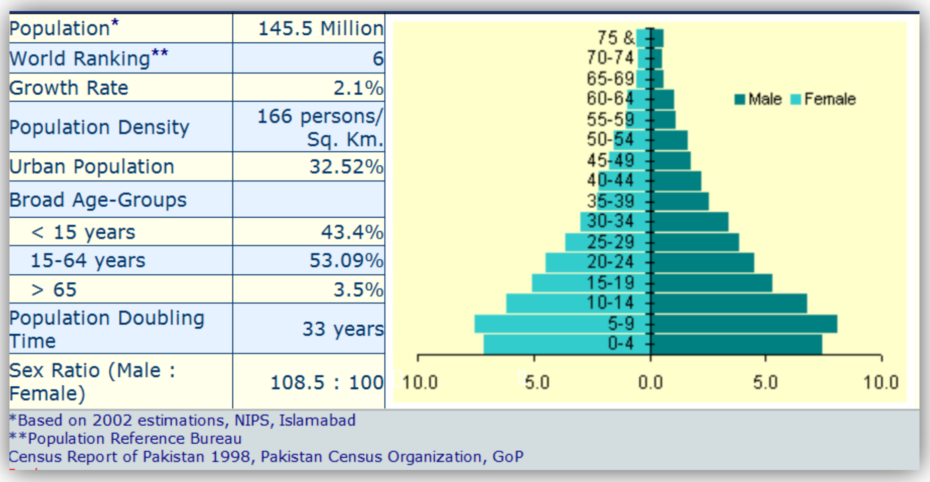
Today’s luxurious, busy and over ambitious life style has led maximum people to be a regular visiting patient of hospital. This can very easily observed from amount being spent on health insurance in developed and developing world. For same very reason, efforts for legislation are under way in USA to double the price of fast food to prevent people from variety of diseases like obesity, cardiac, kidney etc. The most glaring aspect is that an individual is suffering from more than one disease; hence he needs to keep separate medical record of each of medical prescription, doctor and lab visit. This has added a lot of complexity for a regular visiting patient in terms of medical record keeping system, even if he is a meticulous person.

It is quite possible for you to forget some of your relevant medical documents at home while going to doctor. Or they get torn or lost while travelling/house shifting etc. Above all, even if you have got all your documents in perfect shape, how to arrange them properly with respect to date, disease and lab visit. This complexity is further supplemented by the fact that at times in an emergency; you have to see your doctor at very short notice. Imagine a scenario when you are unconscious or injured as result of some road accident, then how to fetch your documents and from where? Or you are out of city for a business trip or picnic and suddenly you have to visit a doctor, you have not been before? Or imagine a scenario when your year old problem aggravates all of sudden and you have lost all relevant documentation?

Analyzing all above problems we have made a sincere effort to develop a system which can make an individual's medical records easily available for himself and his doctor. Patients' complete medical records will be saved in his/her mobile device's external storage (mini/micro SD card, as almost every individual is using it now a day). From the external storage, it will be transferred to doctor's tablet (Android-based) through Bluetooth/Wi-Fi. Similarly the doctor's prescription will be synchronized with patients' mobile's external storage. All the medical records will be maintained in central database which can be made easily available to any doctor or any hospital anywhere in Pakistan. So just take along your mobile while going to doctor and all will be ok. And if you are brought in a hospital unconscious, your record will be found through your name or CNIN no.

1.2 Background

Pakistan's population and health care statistics do show that we are in great problem while providing medical facilities to people, as shown in following figures:



Population*	145.5 Million
World Ranking**	6
Growth Rate	2.1%
Population Density	166 persons/ Sq. Km.
Urban Population	32.52%
Broad Age-Groups	
< 15 years	43.4%
15-64 years	53.09%
> 65	3.5%
Population Doubling Time	33 years
Sex Ratio (Male : Female)	108.5 : 100

*Based on 2002 estimations, NIPS, Islamabad
**Population Reference Bureau
Census Report of Pakistan 1998, Pakistan Census Organization, GoP

Figure 1-1: Population Pyramid of Pakistan, 1998

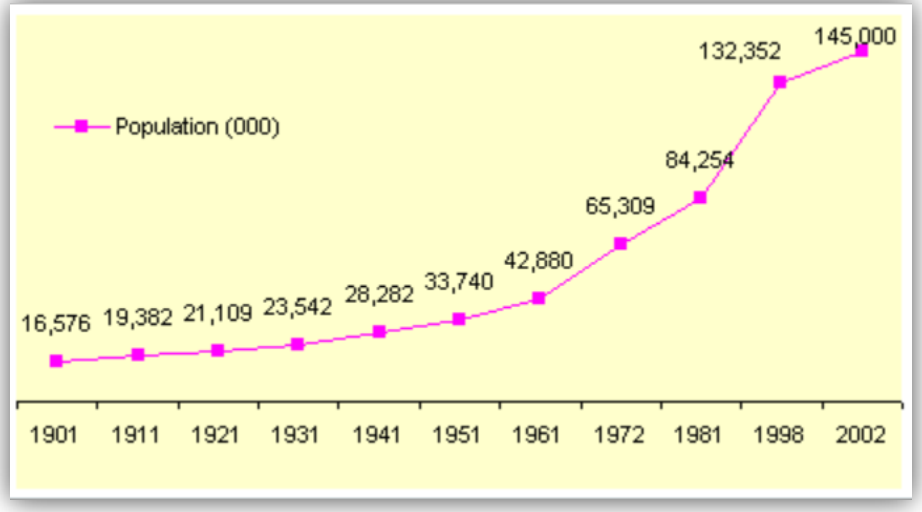


Figure 1-2: Population of Pakistan (1901-2002) from census

The population of Pakistan, since 1901 has been shown in figure 2 which clearly shows that growth rate of population is increasing greater day by day. In following "figure 3" causes of deaths have been shown through charts showing trends and causes of patients death.

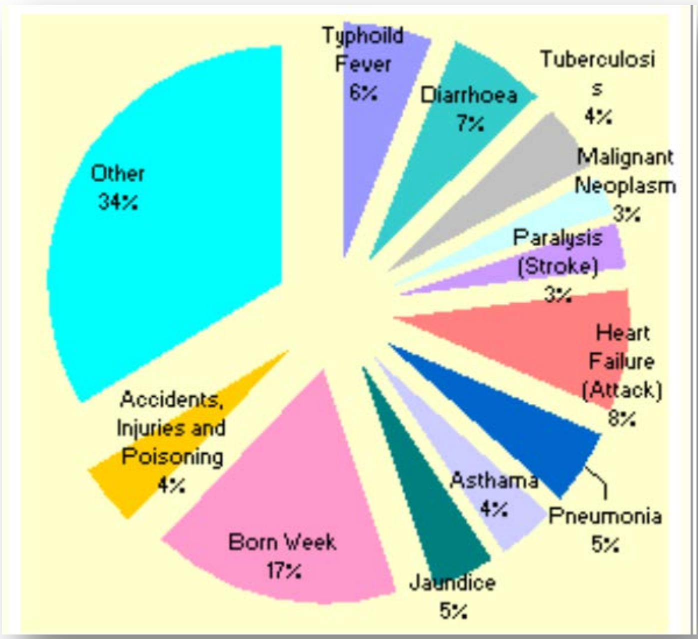
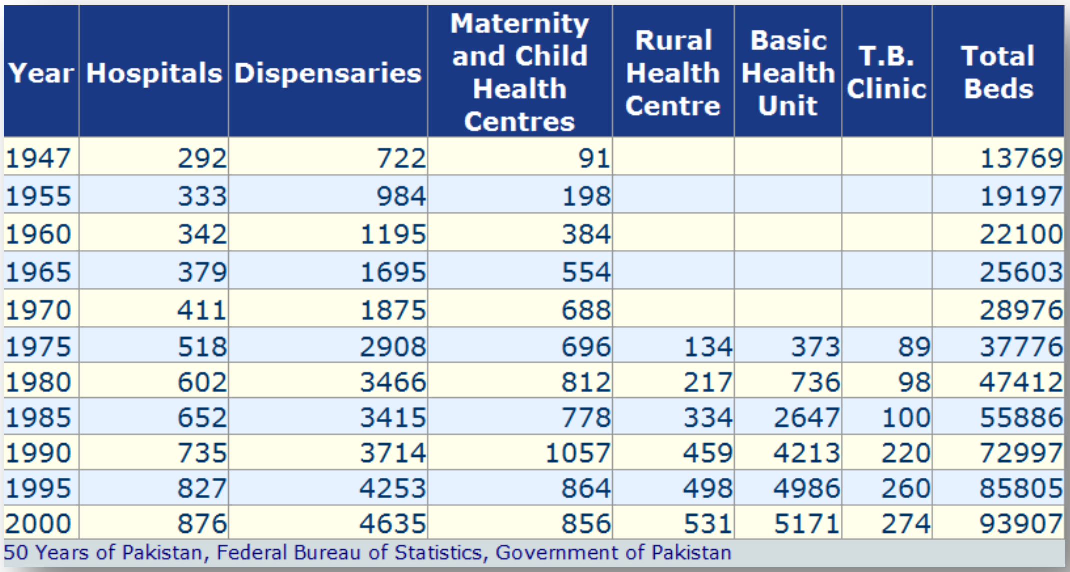


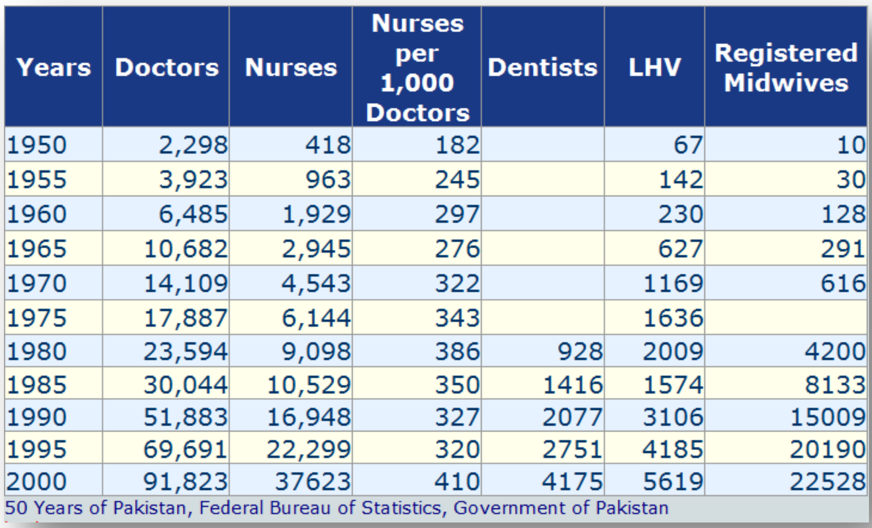
Figure 1-3: Causes of Deaths, Pakistan (2000)
Pakistan Demographic Survey 2000, Federal Bureau of Statistics, Government of Pakistan



Year	Hospitals	Dispensaries	Maternity and Child Health Centres	Rural Health Centre	Basic Health Unit	T.B. Clinic	Total Beds
1947	292	722	91				13769
1955	333	984	198				19197
1960	342	1195	384				22100
1965	379	1695	554				25603
1970	411	1875	688				28976
1975	518	2908	696	134	373	89	37776
1980	602	3466	812	217	736	98	47412
1985	652	3415	778	334	2647	100	55886
1990	735	3714	1057	459	4213	220	72997
1995	827	4253	864	498	4986	260	85805
2000	876	4635	856	531	5171	274	93907

50 Years of Pakistan, Federal Bureau of Statistics, Government of Pakistan

Table 1- 1: Health Establishments in Pakistan 1947 to 2000



Years	Doctors	Nurses	Nurses per 1,000 Doctors	Dentists	LHV	Registered Midwives
1950	2,298	418	182		67	10
1955	3,923	963	245		142	30
1960	6,485	1,929	297		230	128
1965	10,682	2,945	276		627	291
1970	14,109	4,543	322		1169	616
1975	17,887	6,144	343		1636	
1980	23,594	9,098	386	928	2009	4200
1985	30,044	10,529	350	1416	1574	8133
1990	51,883	16,948	327	2077	3106	15009
1995	69,691	22,299	320	2751	4185	20190
2000	91,823	37623	410	4175	5619	22528

50 Years of Pakistan, Federal Bureau of Statistics, Government of Pakistan

Table 1-2: Health Personnel in Pakistan 1950 to 2000, Pakistan

If this, slow, error prone, difficult and unsecure medical system continues in our country, we will be risking people lives and their trust in government capability for providing basic healthcare needs. And we will be wasting so many resources (time, money etc.) with difficult and challenging paper based medical system with poor management.

1.3 Problems Addressed

Changing time leaves nothing unchanged. Every dawn is opening new avenues of improvement and exploration. Stagnation and procrastination is thief of time. Gone are the days of documents and paper environment. Mostly used phrase of “Global Village” symbolizes that world is shrinking.

Considering human nature, it is quite possible for a common person:

a. to forget relevant medical documents at home while going to doctor for checkup

b. to get your medical documents torn

c. to get your documents lost

The complexity is further supplemented by the fact that, being a human being, it is quite tough to arrange or maintain the paper-based medical records properly with respect to date, disease and lab visit. Imagine yourself facing any single situation of the following:

a. When you are unconscious or injured as result of some road accident, then how to fetch your medical records and from where?

b. When you are out of city for a business trip or picnic and suddenly you have to visit a doctor, you have not been before?

c. When your some old medical problem aggravates all of sudden and you have lost all relevant documentation?

d. When all your medical records need to be analyzed since your childhood?

e. When diseases trends need to be statistically analyzed in order to keep the track of specific disease?

f. When medicines usages and effects statistics need to be analyzed?

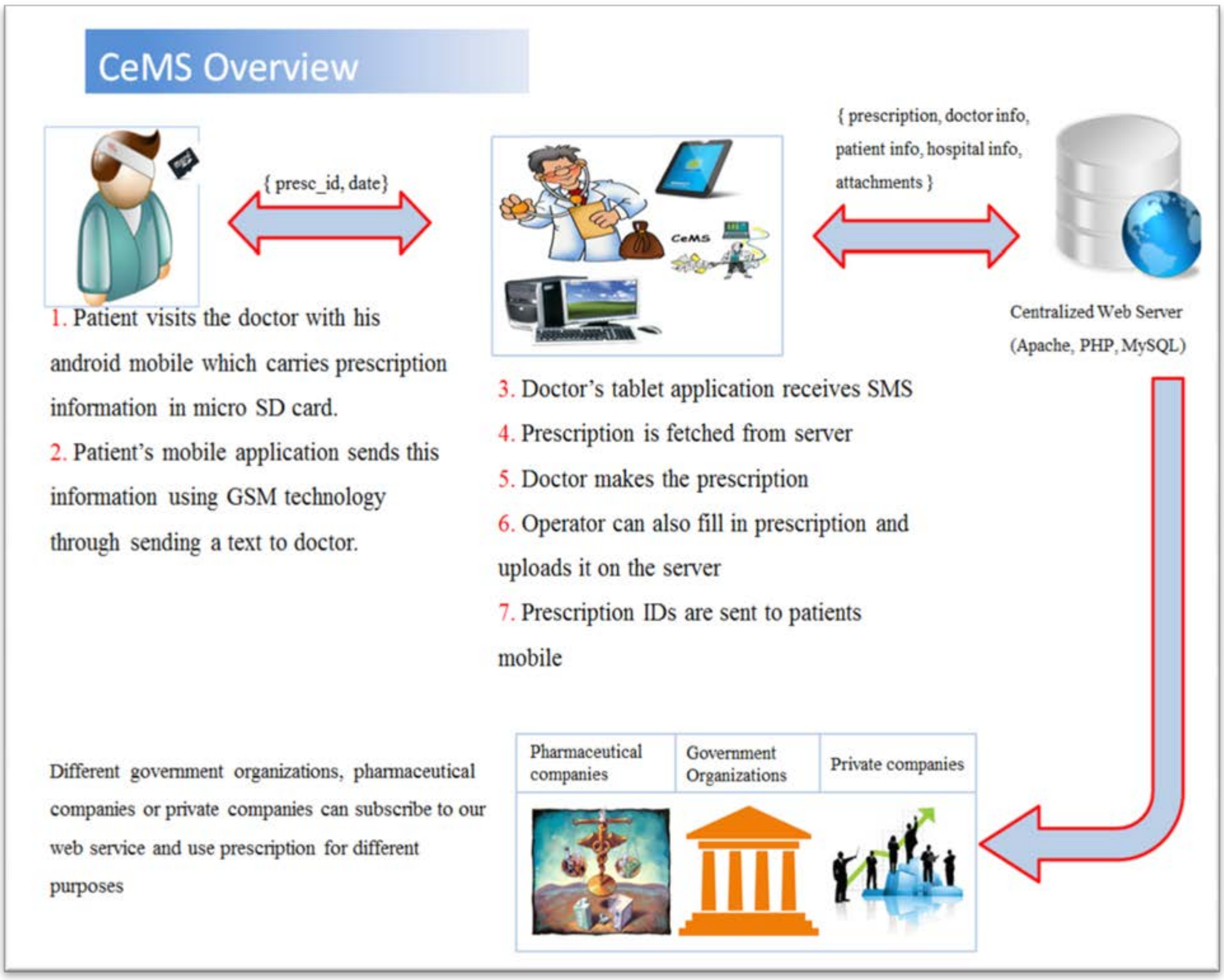
And above all being a patient it is very difficult to manage all your medical records of your life time along with you. It is not that easy to handle then well and proper. Also the growth rate of population in Pakistan is increasing at quite a pace and so does the diseases. So it is becoming the greater challenge to cope with the medical records maintenance using the current paper based medical records keeping systems.

1.4 Solution

Analyzing all above problems we have made a sincere effort to develop a system (including web service, android application and desktop application) which can make an individual's medical records easily available for him, doctors, medical laboratories, pharmaceutical companies and hospitals as shown in "**figure 4**".

The key idea is to use "micro SD cards" of patients' mobile phones for storing their complete medical records. Patients' complete medical records or their prescriptions, either made by doctors or clinics, hospitals, labs, will also be saved on centralized database server and on their micro SD cards. Using Android tablet, the doctor's prescription will be synchronized with server. All the medical records will be maintained in central database which can be made easily available to any doctor or any hospital anywhere in Pakistan.

Now patients have to keep their mobile phones (CeMS Application enabled) along with them which will carry a unique CeMS system ID for identifying them. They can go everywhere without worries of managing their medical records. And if you are brought in a hospital unconscious, your record will be found through unique CeMS system ID.


CeMS Overview
{ presc_id, date}
{ prescription, doctor info, patient info, hospital info, attachments }
CeMS
Centralized Web Server (Apache, PHP, MySQL)
Pharmaceutical companies
Government Organizations
Private companies

1. Patient visits the doctor with his android mobile which carries prescription information in micro SD card.
2. Patient's mobile application sends this information using GSM technology through sending a text to doctor.
3. Doctor's tablet application receives SMS
4. Prescription is fetched from server
5. Doctor makes the prescription
6. Operator can also fill in prescription and uploads it on the server
7. Prescription IDs are sent to patients mobile

Different government organizations, pharmaceutical companies or private companies can subscribe to our web service and use prescription for different purposes

Figure 1-4: CeMS Overview

1.5 Scope and Deliverables

The scope of CeMS includes a tested, documented, and functional system no longer dependent on paper-based medical records only.

CeMS includes following deliverables:

1.5.1 Document Deliverables

a. Software Project Proposal

b. Software Project Charter

c. Software Project Plan

d. SRS (Software Requirements Specifications)

e. System Detailed Design

f. Final Thesis

1.5.2 Software Deliverables

a. Desktop Client Application for Computer System helping to upload the medical records on Server. This application will be developed in Java using Netbeans IDE for all operating systems having JRE (Java Runtime Environment). This application will provide the users with easy and user friendly GUIs allowing them to update the medical records on the server.

b. Android-based tablets' Client Application for Doctors. It will help the Doctors to view and analyze the medical records. Doctors will be able to download and upload medical records from the centralized server.

c. AWeb Service (including Database Server) for the medical records of patients, information of hospitals, clinics, doctors and medical stores.

d. Android-based mobile phones' Client Application for Patients. It will help the patients to communicate their medical records with their doctors, medical stores and etc. and to carry their medical records in their external storages.

The work breakdown structure for CeMS is given below in figure 5:

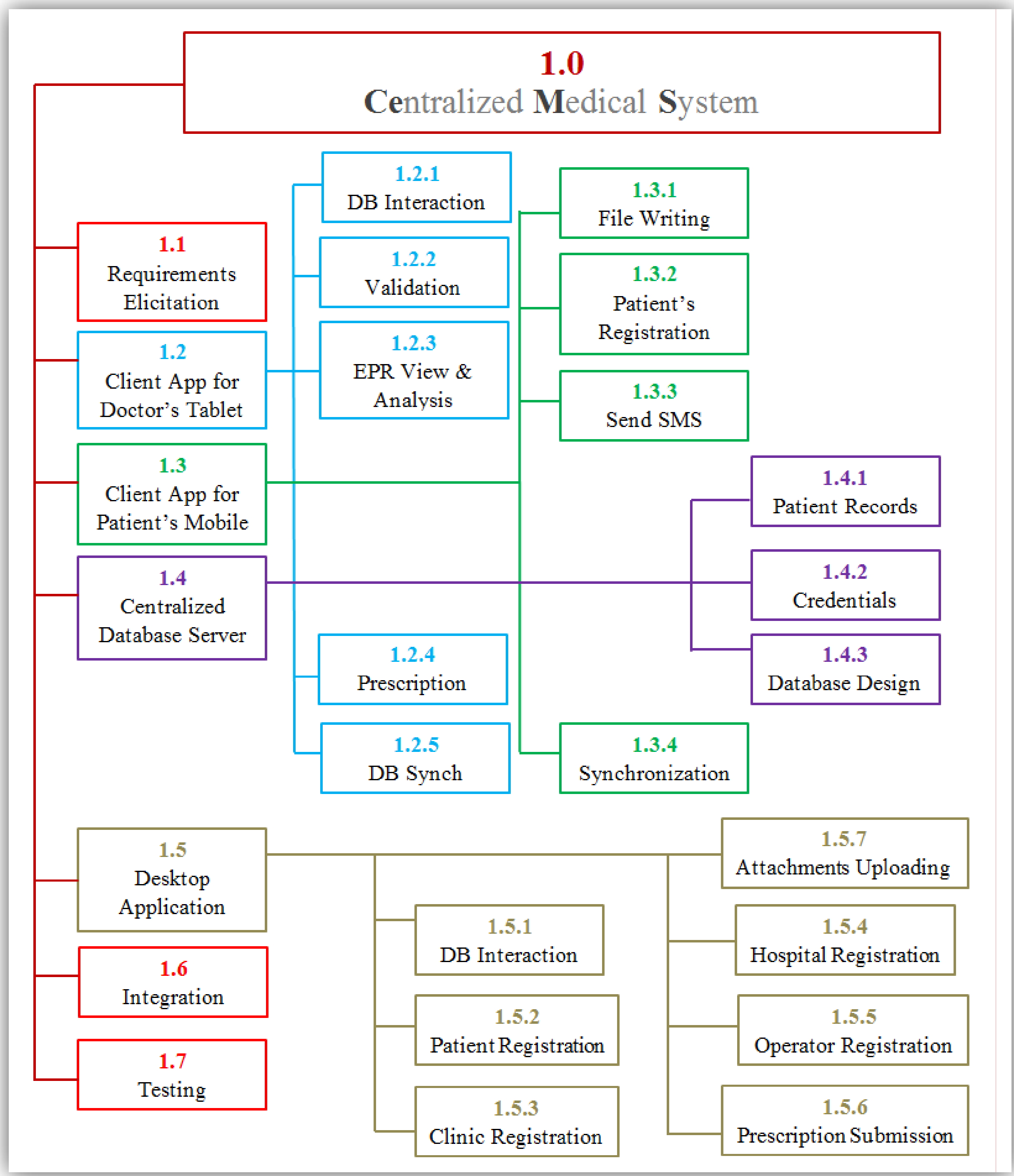

1.0
Centralized Medical System
1.1 Requirements Elicitation
1.2 Client App for Doctor's Tablet
1.2.1 DB Interaction
1.2.2 Validation
1.2.3 EPR View & Analysis
1.2.4 Prescription
1.2.5 DB Synch
1.3 Client App for Patient's Mobile
1.3.1 File Writing
1.3.2 Patient's Registration
1.3.3 Send SMS
1.3.4 Synchronization
1.4 Centralized Database Server
1.4.1 Patient Records
1.4.2 Credentials
1.4.3 Database Design
1.5 Desktop Application
1.5.1 DB Interaction
1.5.2 Patient Registration
1.5.3 Clinic Registration
1.5.4 Hospital Registration
1.5.5 Operator Registration
1.5.6 Prescription Submission
1.5.7 Attachments Uploading
1.6 Integration
1.7 Testing

Figure 1-5: Work Breakdown Structure of CeMS

1.6 Objectives and Goals

The objectives and goals of CeMS have been divided into their academic objectives, covering learning perspectives from the degree education point of view, and end goal or application objectives covering the market perspectives from the end user needs point of view.

1.6.1 Academic Objectives

a. Learning the application development on Android platform

b. Integration of databases with different platforms(Windows, Linux or Android) and their synchronization

c. Understanding of communication protocols of different technologies between different technologies, i.e. Bluetooth, Wi-Fi and GPRS

d. Understanding of medical record keeping system standards (HL7, DICOM etc.)

e. Development of user friendly interfaces for computer literate/illiterate people

f. Understanding of web services development

g. Practicing and exercising the knowledge and skills, learned in the whole degree, to a practical project

1.6.2 End Goals/Application Objectives:

End goal objectives are given in following table.

GOALS	OBJECTIVES
1. CeMS will provide an improved system for maintaining the medical records	System will replace the old paper based system for keeping the patients records, improving the efficiency for accessing, searching and maintain the medical records
2. The medical records will be available on centralized database server all the time as web service	Patients' records and details will be stored on centralized server. Records will be made available only to authenticated requests
3. CeMS will increase the ease of access of medical records for the Doctors	Well engineered Graphical User Interfaces will be developed for the doctors to make them easy to use for the Doctors
4. CeMS will provide data integrity, confidentiality and availability of medical records	Well known security mechanism will be applied to system to ensure the data integrity, confidentiality and availability
5. CeMS will be flexible enough to get integrated with existing standards and also flexible to evolve	We will be developing the system with the intention to make it flexible to accommodate the future evolution and also compatible with existing medical standards

Table 1-3: CeMS Objectives and Goals

1.7 Summary

This chapter described the details about problem statement its objectives and project goals to be implemented. Moreover it highlighted an overview of the project scope of work, its structural diagram and the key parameters which needs to be considered. Research work related to the project and the main objectives and structural analysis has been discussed.

Chapter 2: Literature Review

In this chapter different medical records keeping systems and facilities along with their basic objectives and short comings are discussed. Also, approaches for implementing client and server application and different development and integration approaches are analyzed for in depth knowledge.

2.1 Problem Domain

The first question that comes to mind, naturally, is that ‘Why to keep records?’ Reasons are given as below:

a. It helps in communication among various service medical service providers and users
b. It can anticipate future health problems
c. It can record standard preventive measures
d. Identifies deviations from the expected prescriptions and diagnosis.
e. It is a basis for clinical research.
f. Medical science is an Information industry as shown by following facts:
    i. 35-39% of hospital operating costs are due to professional (doctors, druggists, nurses, paramedics etc.) and patient communications i.e. paper/file based record keeping/maintaining, inter office/departments letters
    ii. Physicians spend 38% and nurses 50% of their time charting
    iii. Exponential growth of medical knowledge and literature

The “***weaknesses or disadvantages”*** of paper-based medical records keeping systems are given as below:

a. Content of data is missing, illegible, inaccurate i.e. one hospital study shows that 11% of tests were repeats to replace lost information

b. Too thick too carry (1.5 lbs. avg.)

c. Fail to capture rationale for diagnosis, prescription, tests etc.

d. Incomprehensible to patients and families as 75% of face sheets had no discharge disposition

e. Agreement between encounter (witnessed) and record is just 29%

f. 20.8% of Medicare discharges coded and interpreted incorrectly

g. Unavailable at up to 30% of patient visits because of any one of following reasons:

   i. Two clinic visits in a day

   ii. Patients keeping record docs in their office

   iii. Failure by doctor to deliver docs to patient while writing opinion in absence of patient

   iv. Misfiled in file room

h. Discontinuity across some institutions in terms of In/outpatient records being maintained separately

2.2 Previous Work Done

Previously no work has been done in Pakistan on such a system and on such an idea. However there are some related projects, which are helpful for developing this system. as motioned below:

2.2.1 Standards for Medical Record Keeping

Keeping in view great importance and difficulty level of maintaining the medical records, their standards were introduced. Record keeping standards can be sub-divided into two categories:

a. Generic standards for good practice and specific standards to define
b. The structure and content in specific clinical contexts

2.2.1.1 Generic Standards

Generic medical record keeping standards apply to all medical notes and address the broad requirements for clinical note keeping.

2.2.1.2 Standards for Structure and Content

Standards are also needed so that records are structured appropriately and clinical information is recorded in the right place. Content standards apply to the format and definition of what is recorded in this structure.

2.2.2 The Electronic Patient Record

There are two principal components to the electronic patient record programs for hospitals in England: the Summary Care Record and Detailed Care Records

2.2.2.1 SUMMARY CARE RECORD (SCR)

The Summary Care Record contains only basic information such as major diagnoses and procedures, current medications, adverse reactions and allergies. It is the single common set of clinical information about patients which will be accessible to all authorized health care professionals treating them anywhere in the country. It is being constructed from the data in primary care records.

2.2.2.2 Detailed Care Records

Over the next few years, as the Electronic Patient Record system develops, NHS organizations which normally work together in a local area – such as hospitals, clinics and general practices - will develop and begin to link and access detailed electronic records for each patient. It is intended that medical records will become increasingly paper free (POINT TO BE HERE IS THAT ALTHOUGH IT WILL BE INCREASINGLY PAPER FREE BUT NOT TOTALLY FREE) and interoperable so that the validity of the data held will be preserved between systems and locations.

2.2.3 Automation

With changing modern trends and technology, it has become almost a routine to use computer for routine day to day work. With little more effort concept of networking various departments and laboratories evolved.

2.2.3.1 Advantage of Automation

Patient Records keeping and coordination for hospital has become easy.

2.2.3.2 Disadvantage of Automation

Patient is not at all benefitted because his handwritten slip has been replaced with a computerized slip i.e. PAPER ENVIRONMENT HAS NOT COME TO AN END.

2.2.4 HL7 Standard

Hospitals and other healthcare provider organizations typically have many different computer systems used for everything from billing records to patient tracking. All of these systems should communicate with each other (or "interface") when they receive new information but not all do so. HL7 specifies a number of flexible standards, guidelines, and methodologies by which various healthcare systems can communicate with each other. Such guidelines or data standards are a set of rules that allow information to be shared and processed in a uniform and consistent manner. These data standards are meant to allow healthcare organizations to easily share clinical information. Theoretically, this ability to exchange information should help to minimize the tendency for medical care to be geographically isolated and highly variable.

2.2.4.1 HL7 standards:

- Version 2.x Messaging Standard – an interoperability specification for health and medical transactions
- Version 3 Messaging Standard – an interoperability specification for health and medical transactions, based on RIM
- Version 3 Rules/GELLO – a standard expression language used for clinical decision support
- Arden Syntax – a grammar for representing medical conditions and recommendations as a Medical Logic Module (MLM)
- Clinical Context Object Workgroup (CCOW) – an interoperability specification for the visual integration of user applications
- Claims Attachments – a Standard Healthcare Attachment to augment another healthcare transaction
- Clinical Document Architecture (CDA) – an exchange model for clinical documents, based on HL7 Version 3
- Electronic Health Record (EHR) / Personal Health Record (PHR) – in support of these records, a standardized description of health and medical functions sought for or available
- Structured Product Labeling (SPL) – the published information that accompanies a medicine, based on HL7 Version 3

2.2.4.2 Electronic Health Record

To develop a basic Medical Record System, the best suitable standard of HL7 for our project was electronic health record (EHR) due to its specific fields which are stated below.

An **electronic health record (EHR)** is an evolving concept defined as a systematic collection of electronic health information about individual patients or populations. It is a record in digital format that is theoretically capable of being shared across different health care settings. In some cases this sharing can occur by way of network-connected enterprise-wide information systems and other information networks or exchanges. EHRs may include a range of data, including demographics, medical history, medication and allergies, immunizationstatus, laboratory test results, radiology images, vital signs, personal stats like age and weight, and billing information.

2.2.5Glide Health

Glide Health is the complete mobile desktop for patients, doctors and other healthcare professionals. Glide Health provides access to patients' healthcare information from virtually any platform or device. Doctors and other healthcare professionals can collaboratively provide comprehensive patient care from virtually any location regardless of software system or device. Patients have anytime, anywhere access to their records from their computer or mobile phone. Glide Health is compatible with Windows, Mac, Linux, Solaris, Android, Blackberry, and iPhone, Palm Pre, Symbian and Windows Mobile.

Glide Health is automatically synchronized with doctor's Electronic Medical Records (EMR) platforms and makes possible compatible and secure exchange of patient records between different systems. In addition, patients can enter their own medical history in text or audio dictation regarding their daily health enabling healthcare professionals to monitor their patient's progress on an ongoing basis. Audio dictation is automatically converted to text and stored with the patient's health records. Patients can interact with their physicians and other health professionals through Glide Health's rights based collaboration suite that includes email, group discussions and video conferencing with integrated file sharing. Glide Health ensures proper healthcare for patients when traveling with comprehensive mobile health records securely accessible from virtually any mobile phone.

2.2.6 Practice Fusion

Practice Fusion is a free, web-based electronic health record company Co-founded in 2005 by Ryan Howard and Alan Wong. Based in San Francisco, California the SaaS startup provides advertising-supported records management and related services to doctors and medical practices. Among the other services provided are charting, e-prescribing, billing, scheduling, and patient management.

2.2.7 Medix System

Medix Corporation, founded in 1991, is a premier supplier to medical laboratories, hospitals, and doctors' offices throughout the United States. The Medix selection of brand name medical supplies includes lab products, pharmaceuticals, and many general use items. Medix leverages its relationships with over 75 vendors to offer special pricing on its catalog of over 6,000 products. With over 17 years' experience in the medical supply industry and satisfied customers all over the United States, Medix is intimately familiar with every product that it sells and ensures that all products are tested and approved by the FDA.

Medix's first priority is always addressing the varied needs of our customers. If you are uncertain of the product that you need, Medix will work with you to find the best product, even if it means recommending a competitor. If you have a question about one of our products or about your order, please contact us by phone at 800.556.9915

2.2.8CeMS

None of the above medical records keeping system provide the functionality of storing the medical records to the patient's mobile device's external storage. It's totally a new idea to integrate different smartphone technologies with centralized server, providing a web service 24/7. CeMS provides the facility to patients to keep their records up-to-date within their external storage of their mobile devices being care free from maintaining all the paper based medical records.

CeMS offer following salient features:

- Paperless Medical Records System
- Patients Records on microSD Cards
- 24/7 access to Patient Records
- Ease of Use for Doctors
- Role Based Security Model
- Statistical Analysis, Reports, Charts
- DICOM Images Analysis
- Reports Attachments

2.3 Summary

Literature overview has been discussed in this chapter. Different methodology and description of techniques have been discussed. It described the proposed approaches along with the mechanisms used and the communication protocol details through which CeMS can initiate its interaction with web server and android devices.

Chapter 3: System Requirements Specifications

System requirements specification for CeMS comprising of functional and non-functional requirements are discussed in this chapter. Use-case diagrams with user specific descriptions highlight requirement elicitation and analysis. Graphical user interface requirements for CeMS enhances the application aesthetic look and user's ease of interaction with CeMS application. Non-functional requirements describes the application security measures, stress handling, congestion control and other quality attributes which must be fulfilled in order to achieve robust and efficient application.

3.1 Product Functions

Main functionalities the system will perform include:

a. Data should be stored on external storage of patient's mobile device
b. Automated data updating and synchronization on patient's mobile storage and doctor's tablet
c. Provide GUI for the Doctor's tablet to view and analyze Patient records and also for prescription
d. Data synchronization on centralized database server, on Doctor's Tablet and on patients' external storage

Top level data flow diagram is given below in figure 6:

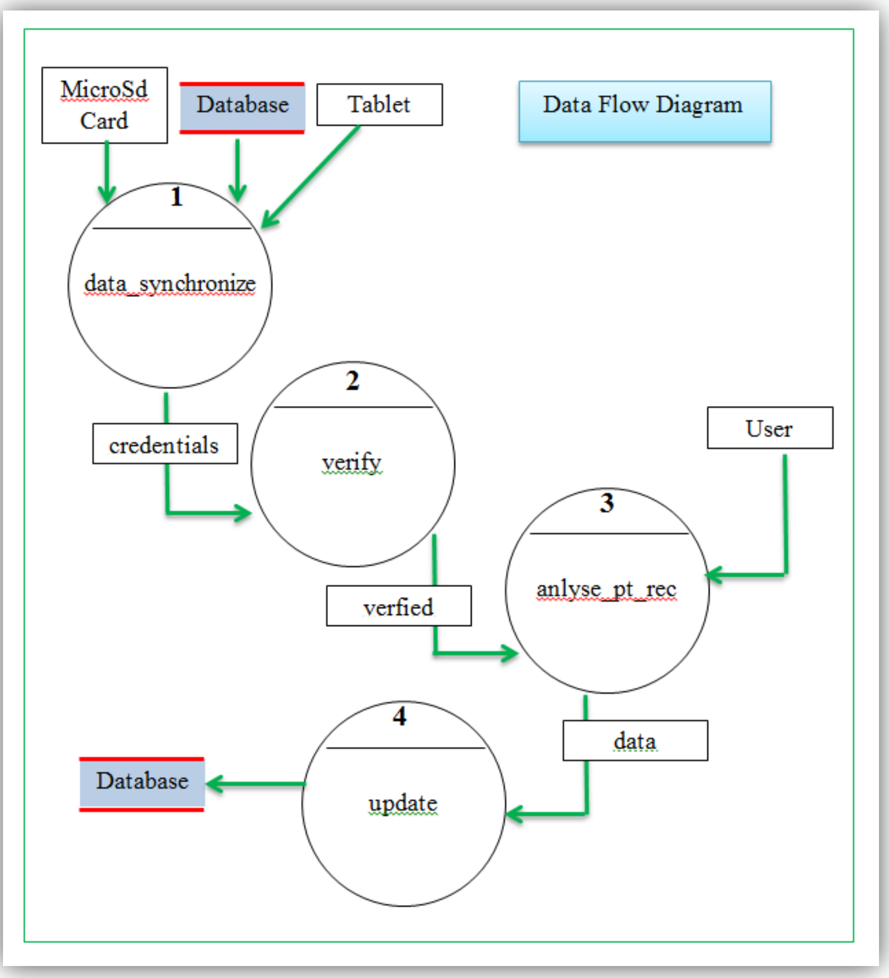


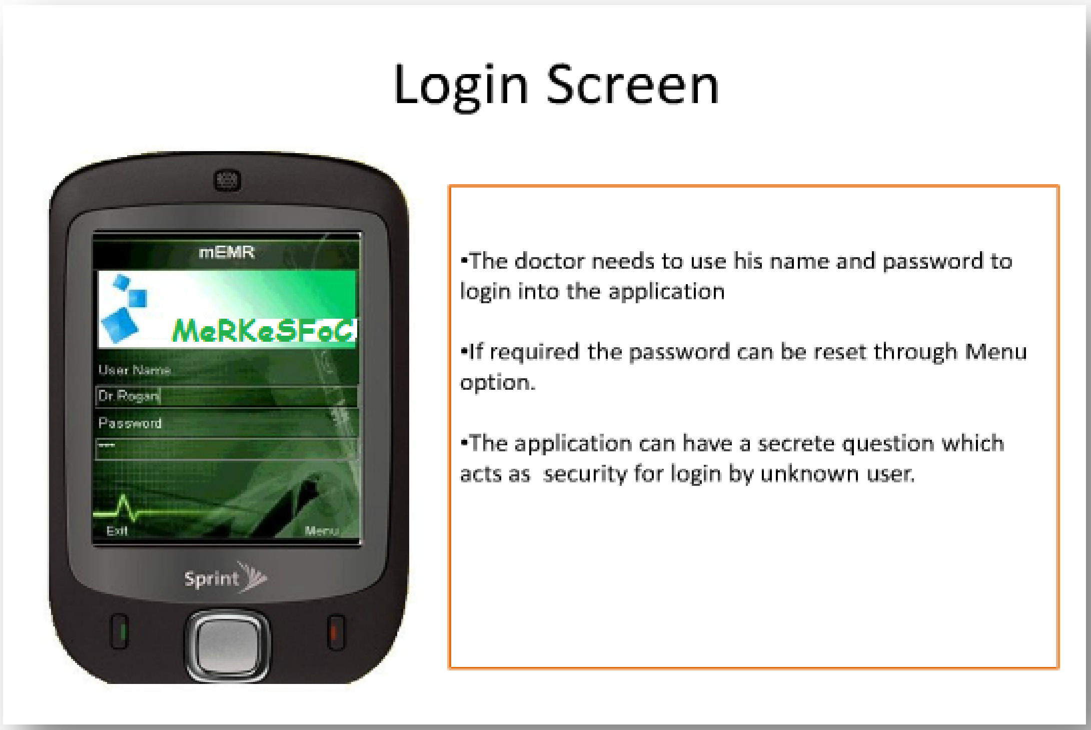
Figure 3- 1: CeMS Data Flow Diagram

3.2 External Interface Requirements

3.2.1 User Interfaces

Some of the GUI design mockups are given below:

.



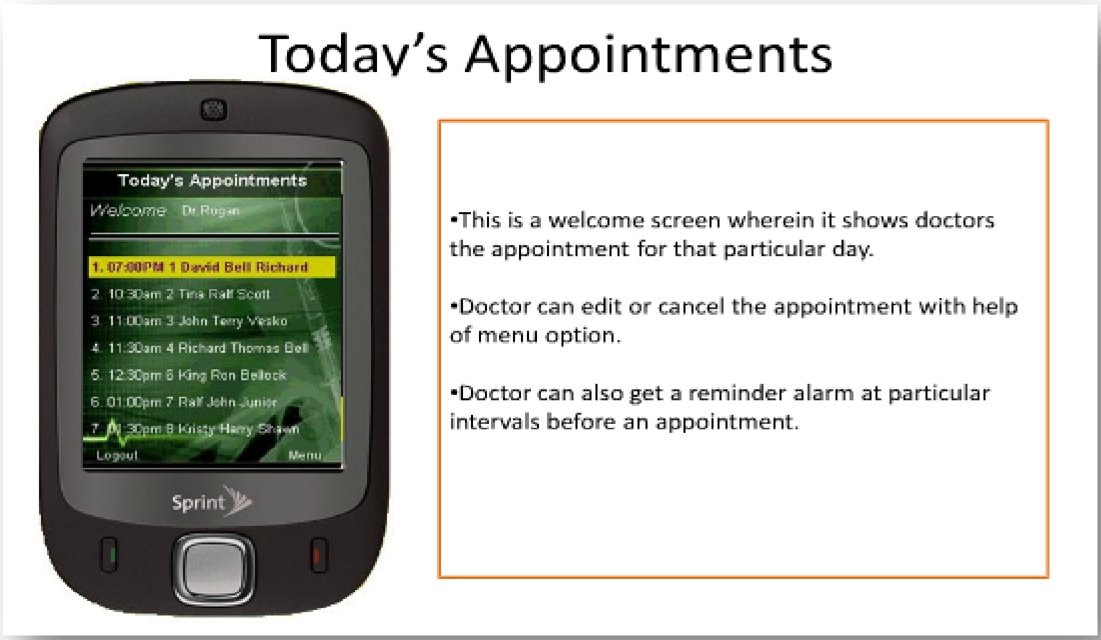
Login Screen

•The doctor needs to use his name and password to login into the application

•If required the password can be reset through Menu option.

•The application can have a secrete question which acts as security for login by unknown user.

Figure 3- 2: CeMS Android Application Login Screen



Today’s Appointments

Today's Appointments

Welcome Dr. Rogan

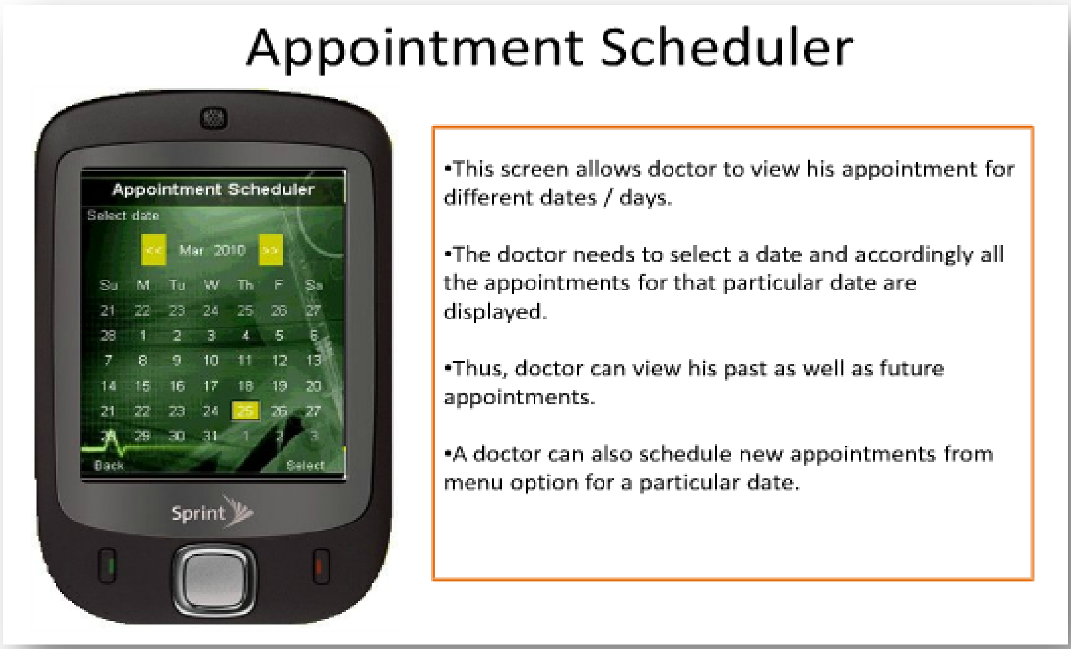
1. 07:00PM 1 David Bell Richard
2. 10:30am 2 Tina Ralf Scott
3. 11:00am 3 John Terry Vesko
4. 11:30am 4 Richard Thomas Bell
5. 12:30pm 6 King Ron Bellock
6. 01:00pm 7 Ralf John Junior
7. 01:30pm 8 Kristy Harry Shawn

•This is a welcome screen wherein it shows doctors the appointment for that particular day.

•Doctor can edit or cancel the appointment with help of menu option.

•Doctor can also get a reminder alarm at particular intervals before an appointment.

Figure 3- 3: CeMS Android Application Appointment Screen



Appointment Scheduler

Appointment Scheduler

Select date

<< Mar 2010 >>

Su	M	Tu	W	Th	F	Sa
21	22	23	24	25	26	27
28	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

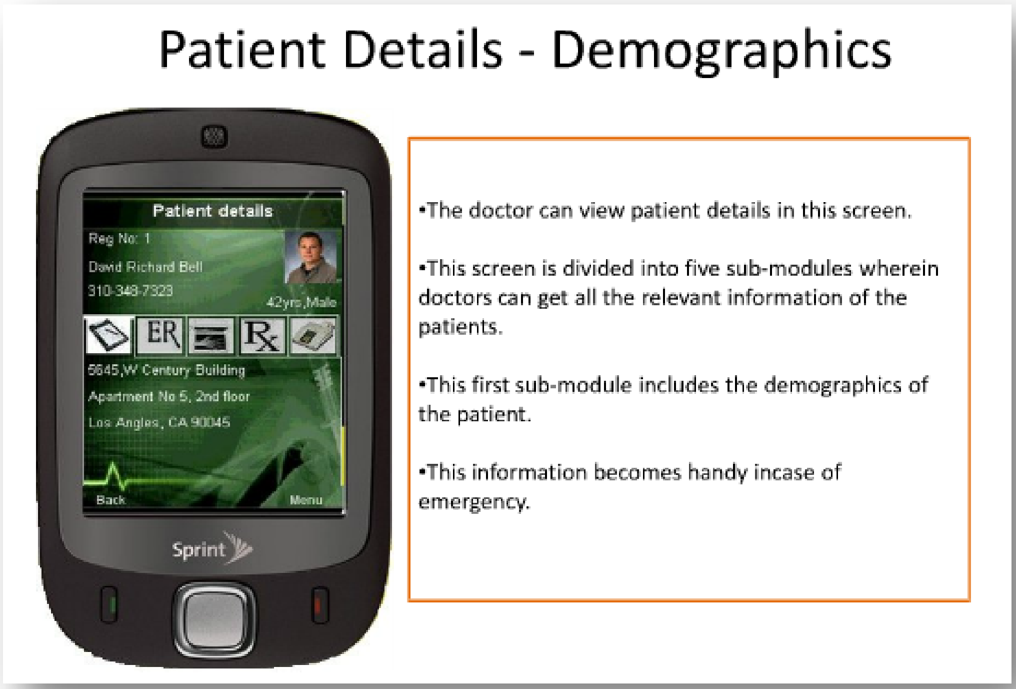
•This screen allows doctor to view his appointment for different dates / days.

•The doctor needs to select a date and accordingly all the appointments for that particular date are displayed.

•Thus, doctor can view his past as well as future appointments.

•A doctor can also schedule new appointments from menu option for a particular date.

Figure 3- 4: CeMS Android Application Schedule Screen



Patient Details - Demographics

Patient details

Reg No: 1

David Richard Bell

310-348-7323

42yrs,Male

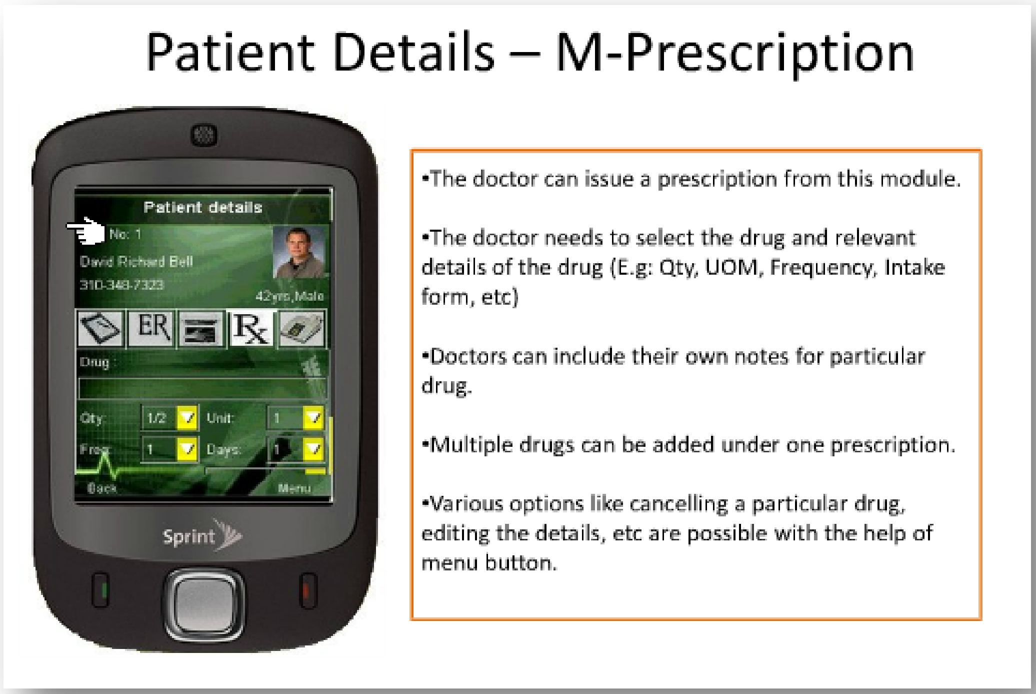
5645,W Century Building

Apartment No 5, 2nd floor

Los Angles, CA 90045

- The doctor can view patient details in this screen.
- This screen is divided into five sub-modules wherein doctors can get all the relevant information of the patients.
- This first sub-module includes the demographics of the patient.
- This information becomes handy incase of emergency.

Figure 3- 5: CeMS Android Application Patient Details Screen



Patient Details – M-Prescription

Patient details

No: 1

David Richard Bell

310-348-7323

42yrs,Male

Drug :

Qty: 1/2 Unit: 1

Freq: 1 Days: 1

- The doctor can issue a prescription from this module.
- The doctor needs to select the drug and relevant details of the drug (E.g: Qty, UOM, Frequency, Intake form, etc)
- Doctors can include their own notes for particular drug.
- Multiple drugs can be added under one prescription.
- Various options like cancelling a particular drug, editing the details, etc are possible with the help of menu button.

Figure 3- 6: CeMS Android Application Prescription Screen

3.2.2 Hardware Interfaces

Required hardware interfaces for the system are:

- Bluetooth
- Wi-Fi
- Database Server
- Micro SD Card

3.2.3 Software Interfaces

Required software interfaces for the system are:

- SQLite3
- MySQL
- NetBeans-7.0
- Eclipse-Helios

3.3 User Classes and Characteristics

CeMS is a product to facilitate doctors for accessing the records of patients and for patients to keep their records safe. Apart from them, hospital management, medical stores and department of health can also get an ease in their responsibilities by using this project.

a. Qualified Doctors having knowledge to use android tablets are mandatory for efficient usage of CeMS. Doctors can easily access data of a patient from the server and can prescribe new medicine just using his/her notepad.
b. Patients just need an android phone with an external memory card to store their data in it, which will enable them to move around free of worried being their patient records get lost
c. Hospital Management can use it to organize their departments according to the needs.
d. Department of Health can use it to do surveys nationwide using data stored on centralized server.

3.4 Operating Environment

CeMS works totally on java platform, using android mobile phones and android tablets and also PHP for server side programming. Databases will be stored using MySQL and SQLite. Visual C# and Java understanding is required. Understanding of Visual Studio 2008 and Eclipse IDEs is required. Mentioning about hardware, any mobile or tablet, having android operating system version 2.2(at least) will work for this project. Data transfer and synchronization between tablet and mobile devices should preferably be done through the Bluetooth Tech. Medical Patient Record standards (HL7) and Databases understanding is required.

3.5 Design and Implementation Constraints

Most important constraint of our project is that we need our mobile phones and tablets having android operating system version 2.2 as of minimum, lower versions cannot work. We should be having access to hospital database so we can have patients record to develop our own database using MySQL and SQLite. Centralized server should be efficient enough that it can handle all the tasks without any delay. Android programming will be done using Eclipse framework only. Security of data includes Confidentiality and Integrity.

3.6 System Features (Functional Requirements)

System Features are given below with priority from High to Low:

3.6.1 Core System Functional Requirements

3.6.1.1 Centralized Database Server for Medical Records (Web Service)

Electronic Medical Records of the patients will be saved in the centralized database server, which will make the Patient Records available 24/7 on the internet, serving as web service

3.6.1.2 Medical Records Storage on Micro SD

Medical records will be stored on the microSD card of Android Mobile of Patients. This feature of CeMS will enable the patient records get automatically synchronized between the centralized server and doctors' tablet.

3.6.1.3 Medical Records Presentation

Graphical user interfaces will be designed for the tablets of Doctors in a way that they must be able to view analyze and edit the patient's medical records with ease of use, for example viewing the BRAIN MRIs of patients, X-rays etc.

3.6.1.4 Updating and Synchronization

All medical records will be synchronized and up-to-date between patient's device, doctor's tablet and centralized server. And the records will be available any time on the server. After the role of requesting person gets verified, then it can get the requested medical record.

3.6.2 Additional Requirements

If the above requirements are implemented then we will add the following features to our system.

- Data confidentiality and integrity
- System integration with existing international standards
- DICOM images support on android for doctors to view patients X-ray, brain MRIs etc. pictures

3.7 Use Case Description

CeMS functional requirements are given below as Use Case Descriptions along with figure 12: use case diagram of CeMS:

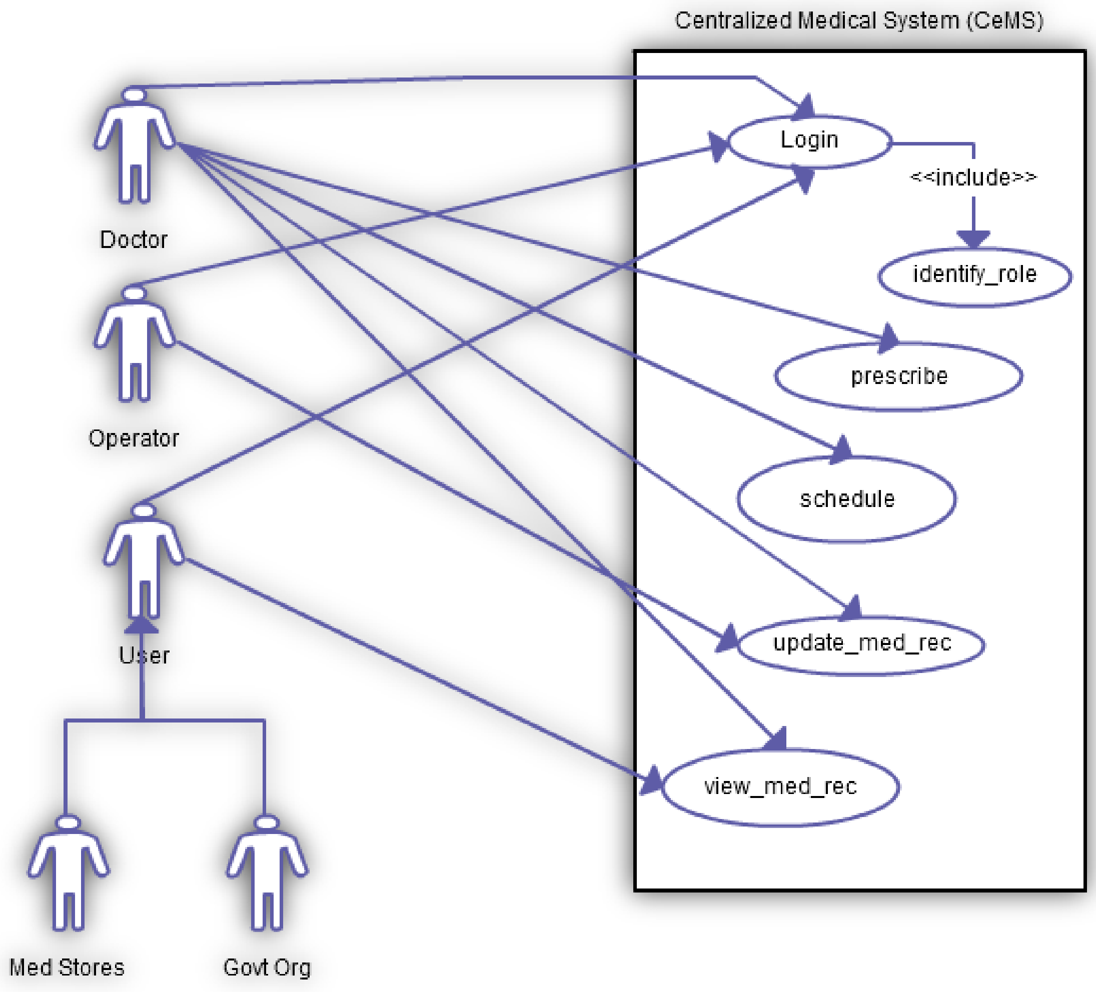


Figure 3- 7: Use Case Diagram of CeMS

3.7.1 Login

Name	Login
ID	UC-L-01
Version	1
Author(s)	Maj. Humayun, GC Mohsin, NC Hafiz M Hassan
Primary Actor(s)	Doctors, Operator, Users
Date	4/10/11
Summary	This use case details the login steps of a Doctor, operator or users; it is necessary to gain access to other functionality of the system.
Basic Path	1. The system displays prompts for the User.id and User.passwords; The LOGIN button is displayed as well 2. User enters the User.id 3. User enters the User.passwords 4. User presses the LOGIN button 5. The system locates the User's object 6. The system verifies the User's access rights and identifies its role 7. The system displays the overview of User's accounts
Alternative Paths	None
Exception Paths	▪ If in 5. the User's object cannot be located, the system displays the UserNotFoundMessage and terminates the use case ▪ If in 6. the system fails to verify User's access rights (Customer.paswwords is different than provided), the system displays the UnkownUserMessage and terminates the use case.

Extension Points	
Triggers	The User elects to use the system
Assumption	
Pre-Condition	
Post-Condition	The Customer object is available to other use cases available to the same User

Table 3- 1: Login Use Case Description

3.7.2Prescribe

Name	Prescribe
ID	UC-Pr-02
Version	1
Author	Maj. Humayun, GC Mohsin, NC Hafiz M Hassan
Primary Actors	Doctor
Date	4/10/11
Summary	This use case allows the Dr. to enter the prescriptions to the specific patient.
Basic Path	1. Doctor selects ‘prescribe’ option 2. The system displays prompts for the prescription 3. Doctor enters the data 4. Doctors selects ‘ok’ button

Alternative Paths	▪ In 4, Doctor can select add more prescription
Exception Paths	
Extension Points	
Triggers	The patient records prescription data form
Assumption	
Pre-Condition	Doctor is logged in to the system
Post-Condition	System displays the summary of prescribed data

Table 3- 2: Prescribe Use Case Description

3.7.3 View/Analyze Medical Records

Name	View_med_rec
ID	UC-VPR-03
Version	1
Author	Maj. Humayun, GC Mohsin, NC Hafiz M Hassan
Primary Actors	Doctors, Users
Date	4/10/11
Summary	This use case details the view/analyze steps for Doctor and Users; how a doctor or a user can view the patient records
Basic Path	1. Doctor selects the specific patient 2. The system locates the requested patient record 3. Patient's records are displayed

Alternative Paths	None
Exception Paths	▪ If in 2. the patient's records cannot be located, the system displays the RecordNotFoundMessage and terminates the use case
Extension Points	
Triggers	The patient records elects to use the system
Assumption	
Pre-Condition	Doctor is logged in to the system
Post-Condition	

Table 3- 3: view_med_rec Use Case Description

3.7.4 Update

Name	Update the system
ID	UC-Up-04
Version	1
Author	Maj. Humayun, GC Mohsin, NC Hafiz M Hassan
Primary Actors	Doctor, Operator
Date	4/10/11
Summary	This use case allows the doctor or operator to update the database or all

	the changes that just have been made
Basic Path	1. Doctor/Admin selects 'update' option 2. The system displays update options 3. Doctor/Admin selects the respective options 4. System prompts ConfirmMessage 5. System saves all the changes and update the system
Alternative Paths	
Exception Paths	▪ In 3, if Doctor selects some old passed date, then system will prompt an ErrorMessage and again displays the schedule screen
Extension Points	
Triggers	The patient schedule meeting activity screen
Assumption	
Pre-Condition	Doctor is logged in to the system
Post-Condition	System displays the summary of scheduled

Table 3-4: Update UseCase Description

3.7.5 Schedule

Name	Schedule the meting
ID	UC-SM-05
Version	1
Author	Maj. Humayun, GC Mohsin, NC Hafiz M Hassan
Primary	Doctors

Actors	
Date	4/10/11
Summary	This use case allows the Dr. to schedule an appointment or view their schedule
Basic Path	1. Doctor selects 'schedule' option 2. The system displays calendar to schedule the meeting or view appointments 3. Doctor picks up the date and set the data 4. Doctors selects 'ok' button
Alternative Paths	
Exception Paths	▪ In 3, if Doctor selects some old passed date, then system will prompt an ErrorMessage and again displays the schedule screen
Extension Points	
Triggers	The patient schedule meeting activity screen
Assumption	
Pre-Condition	Doctor is logged in to the system
Post-Condition	System displays the summary of scheduled

Table 3-5: Schedule Use Case Description

3.8 Nonfunctional Requirements

3.8.1 Performance Requirements

As the project will be running on real time basis, so system should be fast enough that there is no delay in process. Connection between mobile and tablet and data transfer as well as synchronization try not to irritate the users with delays. System must be error free and smooth.

3.8.2 Safety Requirements

N/A

3.8.3 Security Requirements

Confidentiality and Integrity are our main concerns of data security. Data include all the medical records, so its integrity and confidentiality is necessary.

3.8.4 Software Quality Attributes

CeMS is totally software based project, so it should be easily available and easily installable anywhere. This system is concerned with the medical records of patients, so it must be very reliable, error or mistake in any critical situation can cause a big damage. Because of no hardware except the very own mobiles and tablets of users, it is very portable. We have to develop it in such a manner that a person with little knowledge of using android tablets and GUIs can operate it in an efficient way.

3.8.5 Other Requirements

Other requirements include approval of using this project notion wide, need of project sponsors from different health organizations and Hospitals, in terms of cost, information and guidance.

Chapter 4: System Design Specifications

Software Design specification for CeMS comprising of key processing functionalities, high level and low level design architecture, graphical user interface design for CeMS server, Android mobile, desktop application in Java and PHP web application have been discussed in this chapter. It also focuses on Database design and schema for creating tables in MySQL for CeMS database. Furthermore it describes major design constraints related to data design, graphical interface and schema design of CeMS.

4.1 System Architecture

4.1.1 Architectural Design

CeMS consists of following basic components:

a. A Client
b. A Server

Client component is composed of end user applications including “Android Tablet Application” and “Desktop Application”. This component allows user to interact with the system and use the system functionality, downloading and uploading medical records. Client Side of CeMS exposes the system with user friendly and interactive GUIs to users to increase the system usability.

Using client side android based application for patients, patients records information can be communicated through SMS/Wi-Fi/Bluetooth with Doctor’s android tablet information. After getting patient records from the patient’s mobile, doctor’s android tablet will request the centralized server to get the complete prescription.

Using desktop application of CeMS, registration process of different users can be done easily after logging into the system. Further this client side will allow the operator to upload the prescriptions along with attachments on the server.

Server component is composed of all data management modules. This component is basically a Database Server, hiding all low level data handling details from the end users and fulfilling the request from the relevant users. Using the power of PHP, the server-side programming allows the server to process all incoming requests.

4.1.2 Client Server 2-Tier Architecture

Analyzing the system requirements and system specifications, CeMS is best architecture in “2-Tier Architecture” style and is described in:

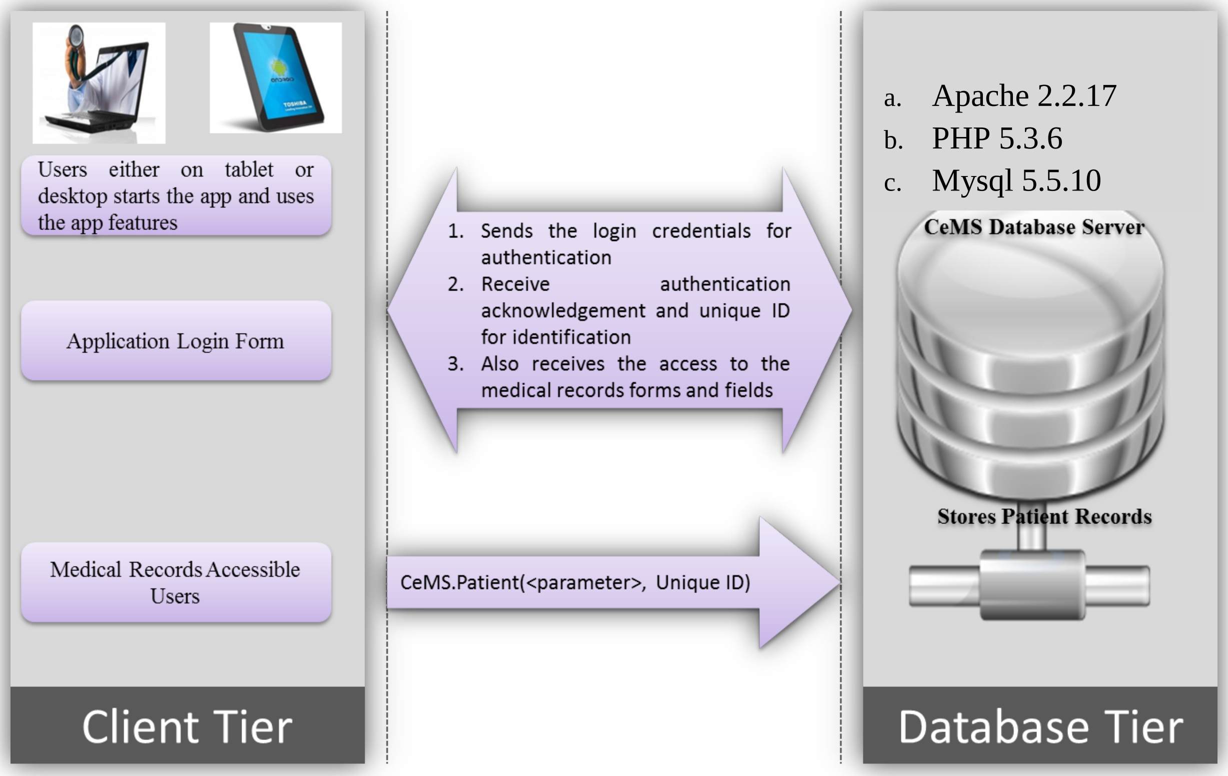

Users either on tablet or desktop starts the app and uses the app features
Application Login Form
Medical Records Accessible Users
Client Tier
1. Sends the login credentials for authentication
2. Receive authentication acknowledgement and unique ID for identification
3. Also receives the access to the medical records forms and fields
CeMS.Patient(<parameter>, Unique ID)
a. Apache 2.2.17
b. PHP 5.3.6
c. Mysql 5.5.10
CeMS Database Server
Stores Patient Records
Database Tier

Figure 4- 1: CeMS 2-Tier Architecture

The client–server architecture is a computing model that acts as distributed application which partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.Clients and servers communicate over a computer network on separate hardware. A server machine is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

In a network, the client/server architecture allows efficient way to interconnect programs that are distributed efficiently across different locations.Client/server architecture illustrates the relationship between two computer programs in which one program is a client, and the other is Server.Servers arestand-alone components that provide specific service data management. Clients arecomponents that call on the services provided by servers.

In Client Tier, user is presented with relevant forms either on Android tablet application or on Desktop application accordingly.

On Android tablet, Doctor:

a. does not communicate with database server unless access to patient records is required
b. sends his login credential to database server in order to get access to relevant patient records only

On Desktop application, Operator:

a. logins to the application first
b. fills the medical records, after getting signed in, and sends the request to CeMS Databases for uploading the records. Server sends acknowledge after successful upload operation

In Database Tier, all hidden, from end user, low level details are implemented and managed through PHP files.

4.1.2 Model View Controller (MVC) Design Pattern

Model View Controller (MVC) pattern isolates "domain logic" (the application logic for the user) of CeMS from the user interface (input and presentation), permitting independent development, testing and maintenance of each (separation of concerns).

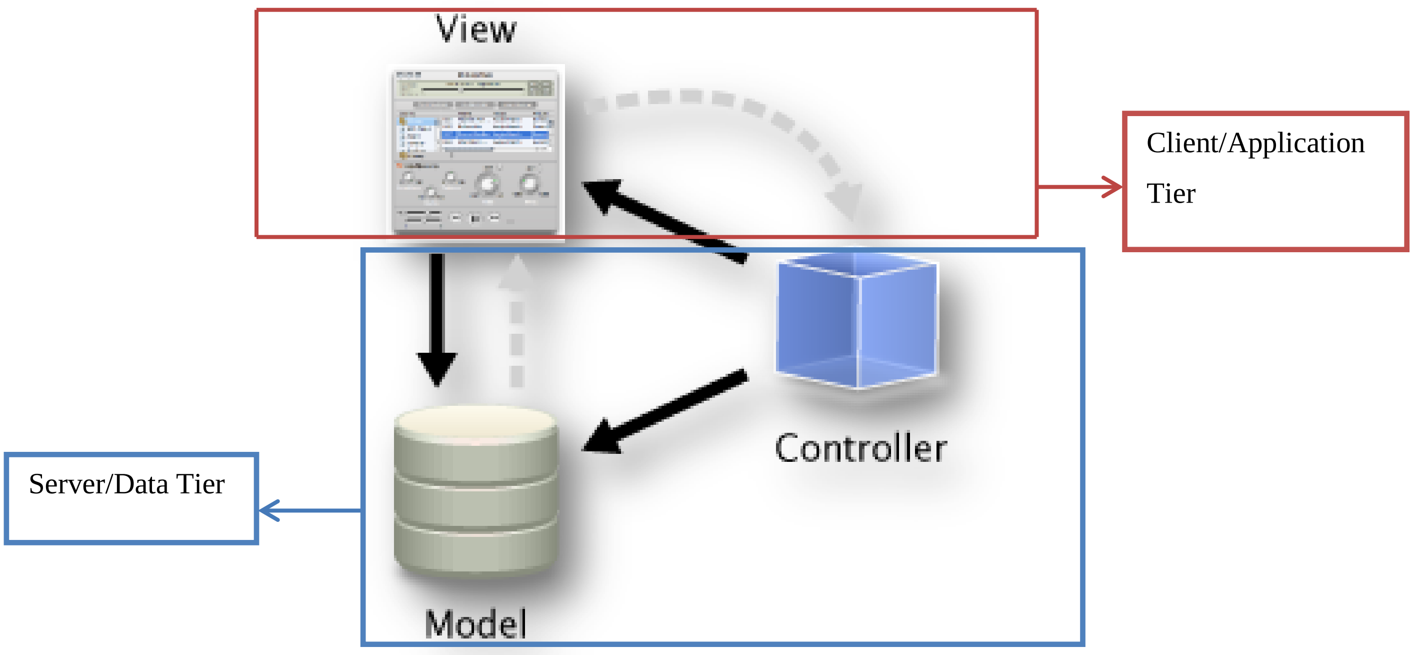


Figure 4- 2: CeMS MVC Design Pattern

MVC pattern creates applications that separate the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements. This patterns allows the CeMS developer to separate the responsibilities at Client Tier or Application Tier, managing the user requests as showing in Figure 14: CeMS MVC Design Pattern.

MVC control flow, in CeMS, is as follows:

a. The user interacts with the user interface provided on Android Tablet Application or desktop application
b. The controller handles the input event from the user interface, via a registered callback, and converts the event into an appropriate user action, understandable for

the model. The controller notifies the model of the user action, resulting in a change in the model's state.

c. The view queries the model in order to generate an appropriate user interface. The view gets its own data from the model.
d. The user interface waits for further user interactions, which restarts the control flow cycle.

The goal of MVC is, by decoupling models and views, to reduce the complexity in architectural design and to increase flexibility and maintainability of code.

Model: The model is the part of the Client Component of CeMS 2-Tier Architecture that encapsulates the application's data. It providesthe routines to manage and manipulate this data in a meaningful way in addition to routines that retrieve the data from the model.

The underlying data access technique is encapsulated in the model. This way, if the Android Tablet Application or Desktop Application is to be moved from a system that utilizes a flat file to store its information to a system that uses a database, the model is the only element that needs to be changed, not the view or the controller.

View: The view is the part of the Client Component that is used to render the data from the model in a manner that is suitable for interaction.

For our applications, the view would be Medical Records that are returned. The view pulls data from the model (which is passed to it from the controller) and feeds the data into a template which is populated and presented to the user. The view does not cause the data to be modified in any way; it only displays data retrieved from the model.

Controller: The controller is responsible for responding to user actions.

In our application, a user action is medical records access request. The controller determines that what request is being made by the user and respond appropriately by triggering the model to manipulate the data appropriately. The controller does not display the data in the model, it only triggers methods in the model which modify the data, and then pass the model into the view which displays the data.

4.2 Decomposition Description

CeMS decomposition description includes:

a. Behavioral Model
b. Object-Oriented Design (OOD)
c. Data-Oriented Design (DOD)

4.2.1 Behavioral Model

Behavioral Model is used to describe the overall behavior of the CeMS and includes

a. Data Processing Model
b. State Machine Model

a. Data Processing Model

Data Processing Model shows how data is processed as it moves through the system and is described with the help of Data Flow Diagram (DFD).

DFD is used to model the CeMS' data processing and describes the processing steps as data flows through our system.It shows end-to-end processing of data and models the system from a functional perspective. DFD is given below:

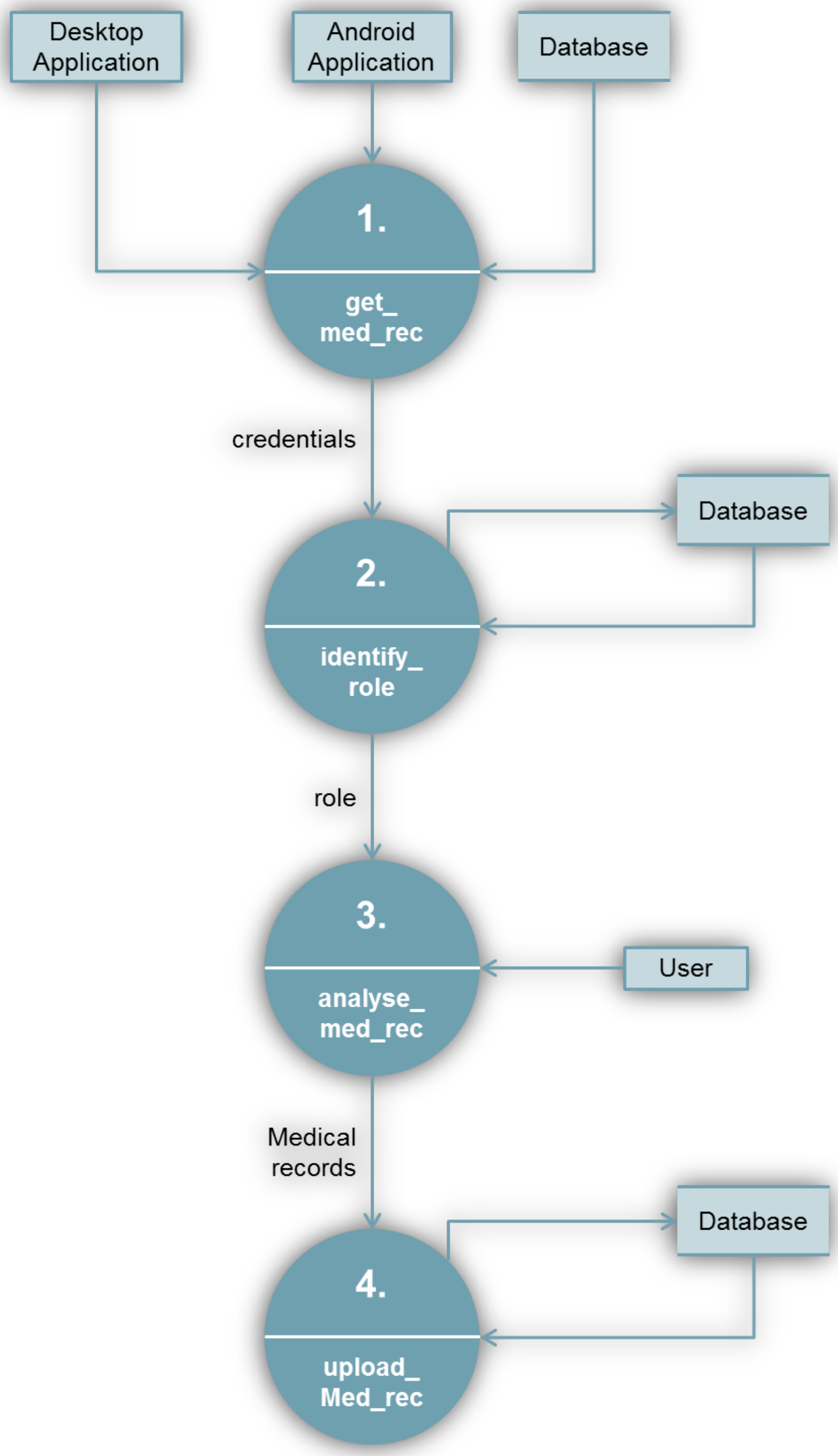


Figure 4- 3: CeMS Flow Diagram

b. State Machine Model

SMM shows the systems response to events and is described with the help of “State Charts”.

State Charts describes the movement of CeMS from one state another state, when an event occurs and brief description of the actions is included in each state. State Chart for CeMS is given below in:

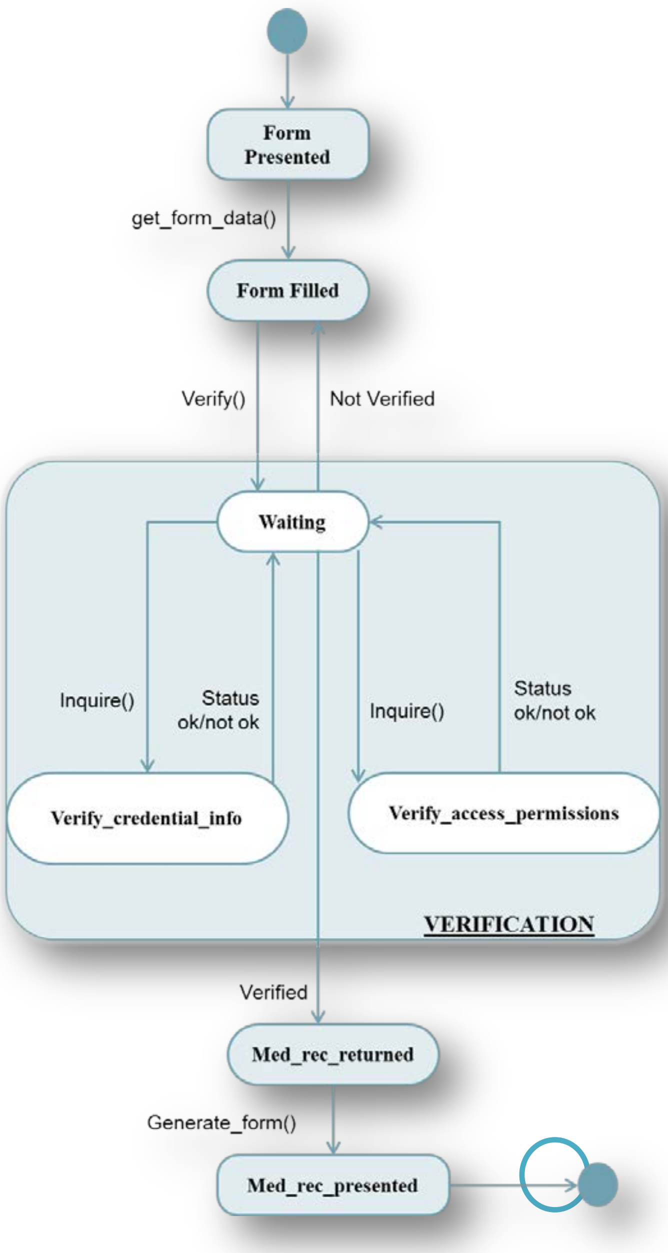


Figure 4- 4: CeMS State Machine Model

4.2.2 Object-Oriented Design

OOD provides the abstractions of CeMS entities and manage them. CeMS functionality is expressed in terms of object services and these objects communicate by message passing to each other.

CeMS OOD includes:

a. Static Model
b. Dynamic Model

a. Static Model

Static models describe the static structure of the CeMS in terms of the system object classes and their relationships and include:

a. Subsystem Model
b. UML Class Diagram

Subsystem models show how the CeMS design is organized into logically related groups of objects and is shown in following figure:

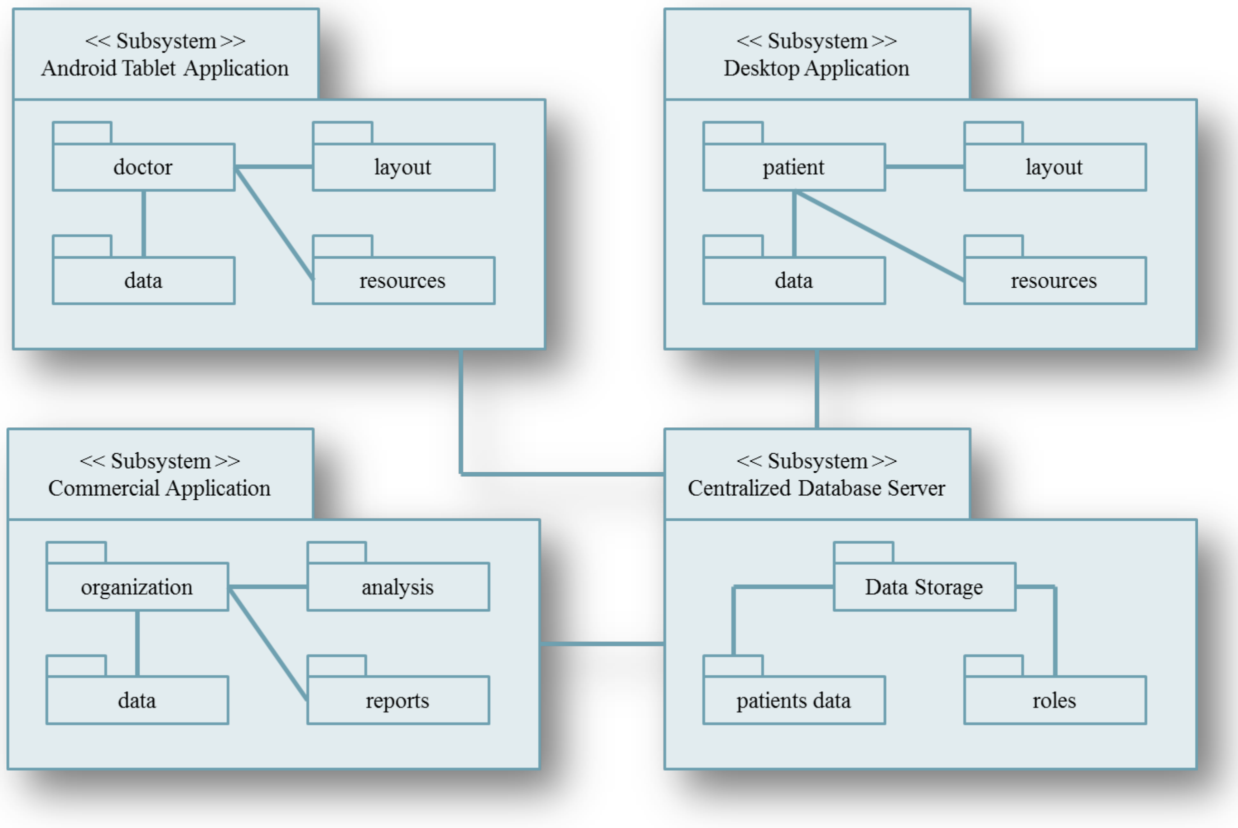


Figure 4- 5: CeMS Subsystem Model

UML Class Diagram: CeMS class diagram is the static structure diagram that describes the structure of the system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes.

Class Diagram of CeMS is given below:

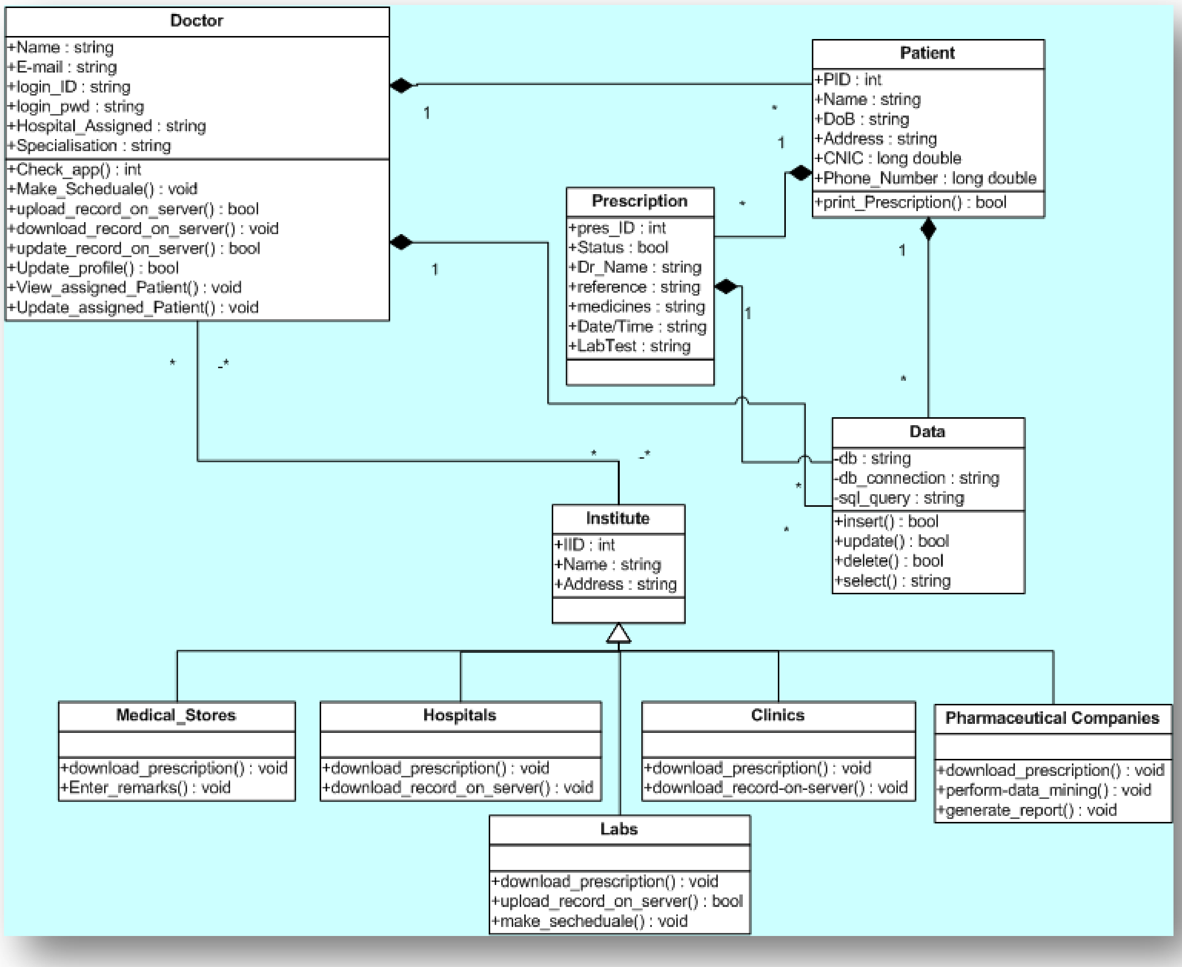


Figure 4- 6: CeMS Class Diagram

b. Dynamic Model

Dynamic model that describe the dynamic structure of the CeMS and that shows the interactions between the systems objects (not the object classes). Interactions are documented including the sequence of service requests made by objects and the way in which the state of the system is related to these objects interactions.

It shows the interactions between objects to produce some particular system behavior that is specified as a use-case. Sequence diagrams (or communication diagrams) in the UML are used to model interaction between objects.

Sequence diagrams show the sequence of object interactions that take place in the system. There are two types of sequence diagrams.

a. System-level sequence diagram
b. Service-level sequence diagram

System-level sequence diagram (SSD) represents the interaction between the actors and the system. During this interaction an actor generates system events to a system, usually requesting some system operation to handle the event.

SSD from the Doctor's view is given below:

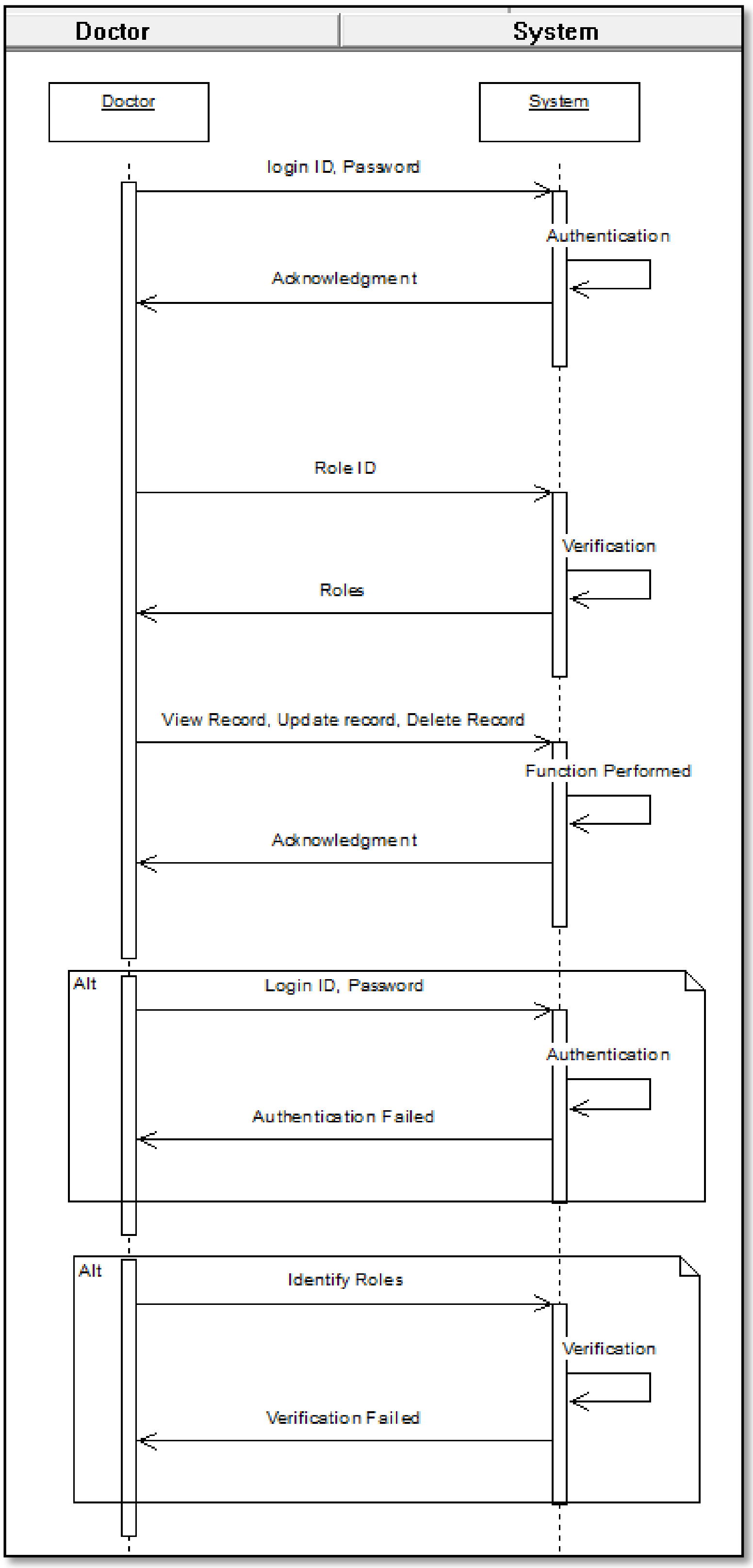


Figure 4- 7: CeMS SSD for Doctors

SSD from the Operator's view is given below:

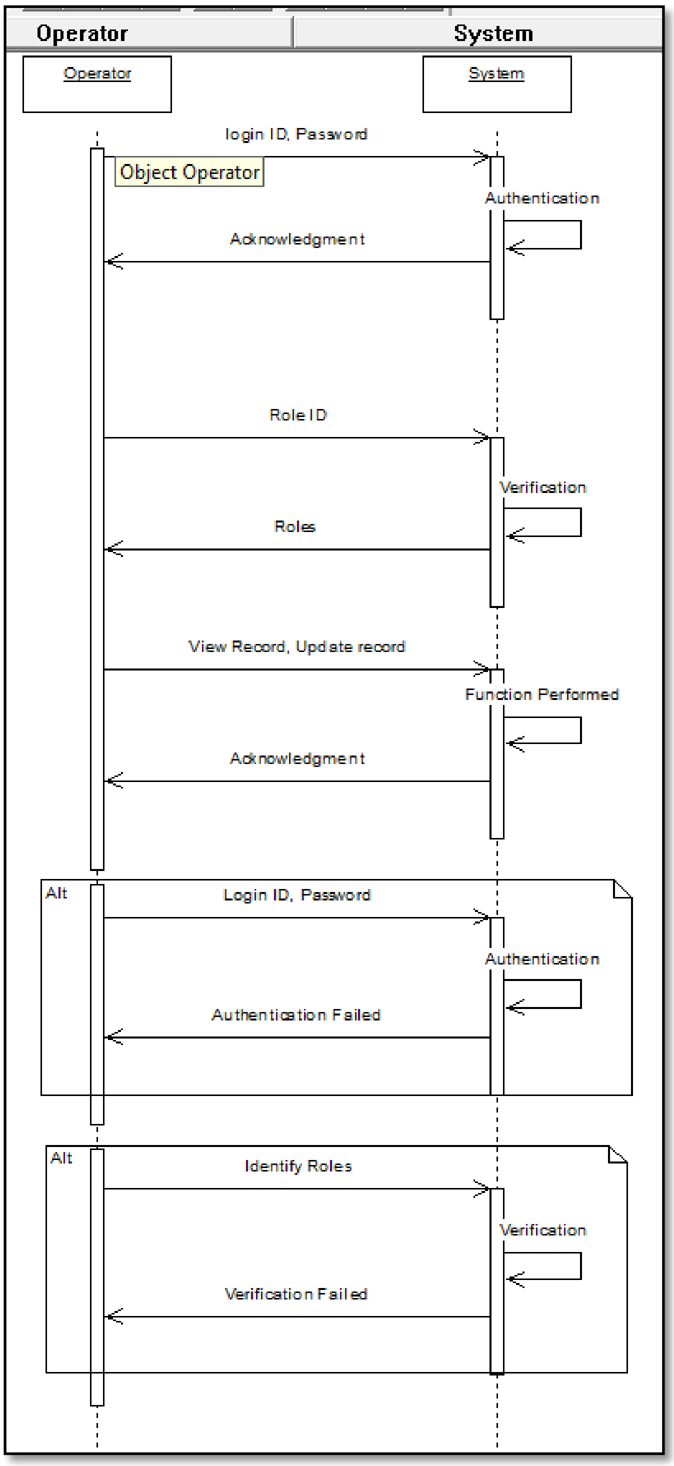


Figure 4- 8: CeMS SSD for Operator

SSD from the Operator's view is given below in figure 21:

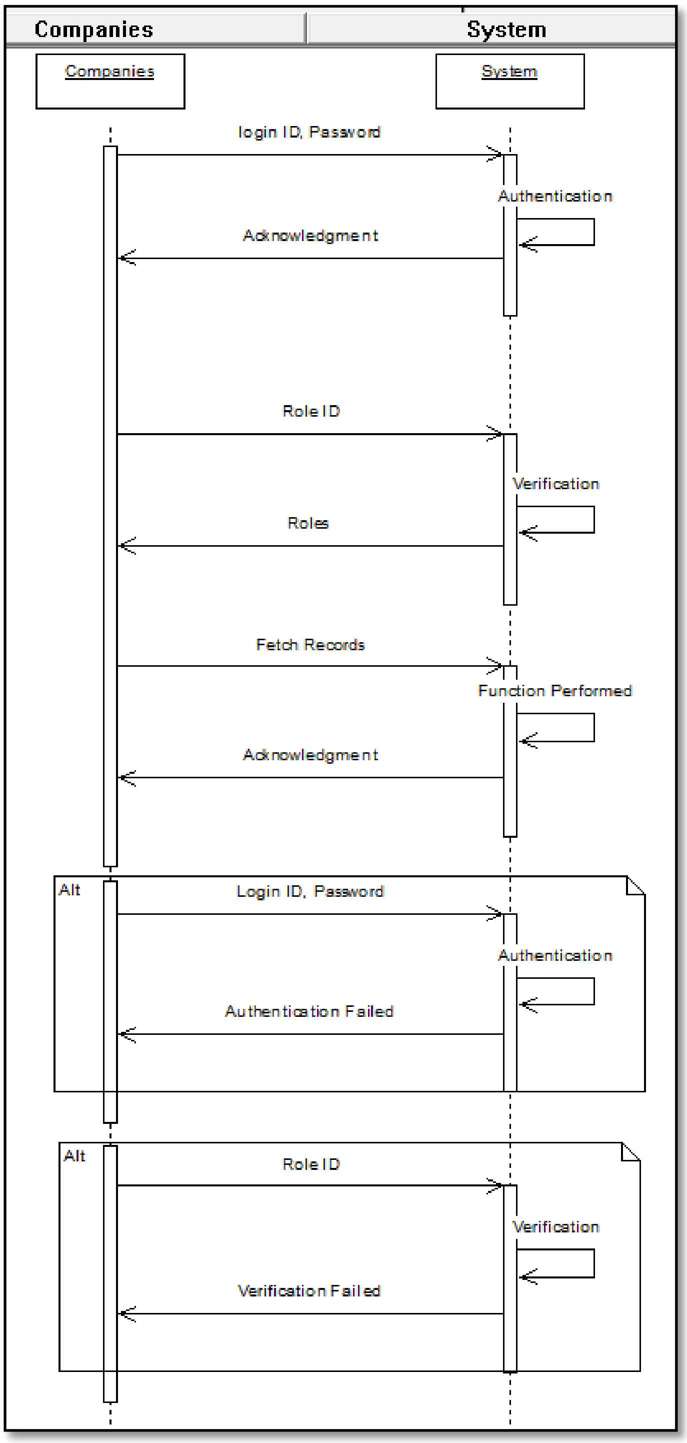


Figure 4- 9: CeMS SSD for Organizations

SSD from the Operator’s view is given below:

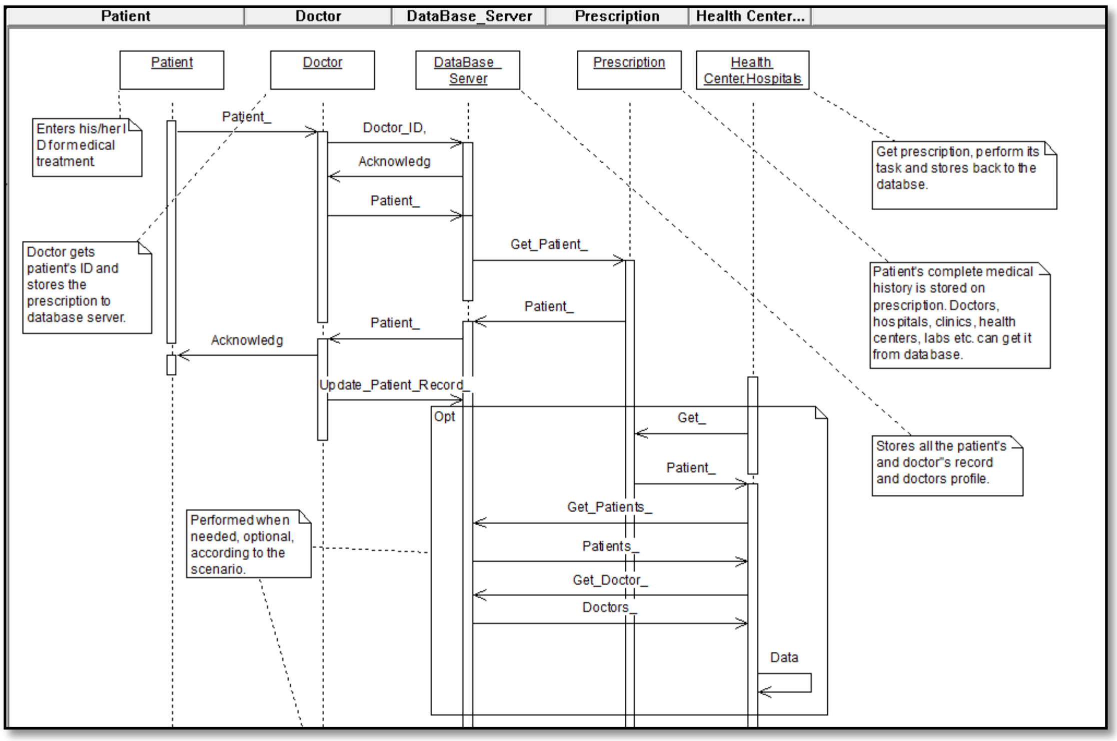

Patient
Doctor
DataBase_Server
Prescription
Health Center...
Patient
Doctor
DataBase_Server
Prescription
Health Center,Hospitals
Enters his/her ID for medical treatment
Patient_
Doctor_ID,
Acknowledg
Patient_
Get_Patient_
Doctor gets patient's ID and stores the prescription to database server.
Patient_
Patient_
Acknowledg
Update_Patient_Record_
Opt
Get_
Patient_
Get_Patients_
Patients_
Get_Doctor_
Doctors_
Data
Performed when needed, optional, according to the scenario.
Get prescription, perform its task and stores back to the databse.
Patient's complete medical history is stored on prescription. Doctors, hospitals, clinics, health centers, labs etc. can get it from database.
Stores all the patient's and doctor's record and doctors profile.

Figure 4- 10: CeMS SSD-1

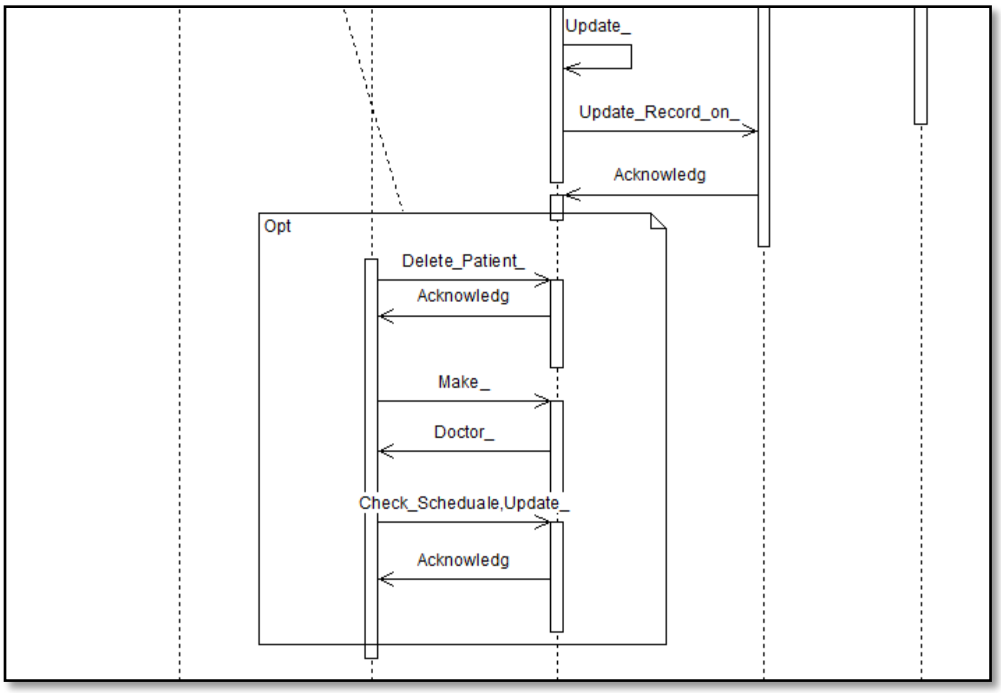


Figure 4- 11: CeMS SSD-2

4.2.3 Data-Oriented Design (DOD)

Using DOD, first the entities are determined for CeMS Server, and then entity inter-relationships are examined to develop the additional entities needed to support the relationships. CeMS constitutes of databases and rapid transfer of data between objects, so implementation of DOD is a better approach for the development of system.

CeMS DOD includes

a. Entity Relationship Diagram (ERD)
b. Relational Model

a. Entity Relationship Diagram (ERD):

ERD model identifies the entities that exist in our system and the relationships between those entities. It is used as a way to visualize a relational database server: each entity represents a database table, and the relationship lines represent the keys in one table that point to specific records in related tables. It is given below in figure 24:

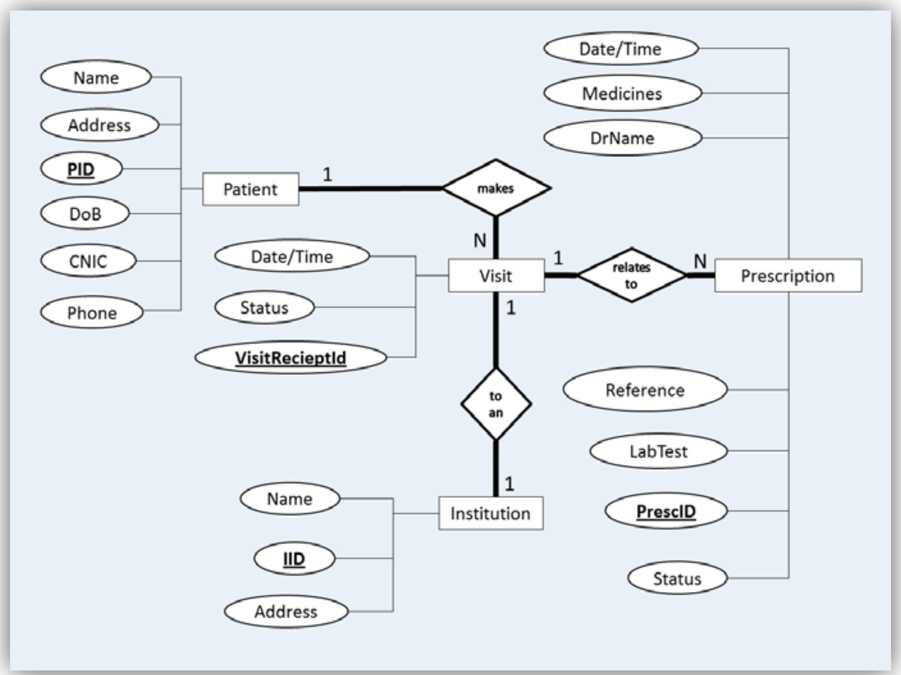


Figure 4- 12: CeMS Abstract ERD

Detailed ERD is given in Figure 15: CeMS Detailed ERD

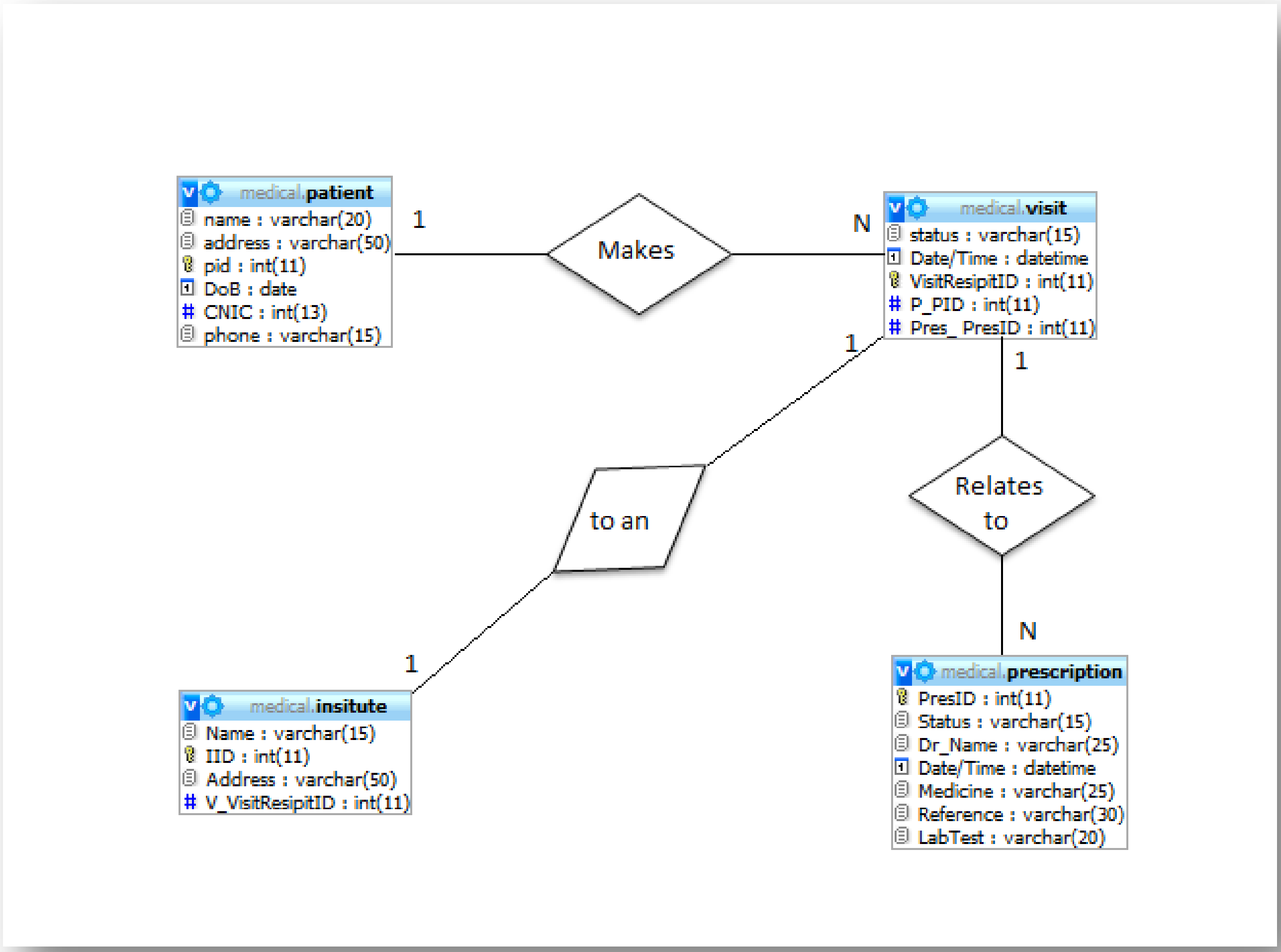


Figure 4- 13: CeMS Detailed ERD

b. Relational Model

The CeMS relational model describes the database as a collection of predicates over a finite set of predicate variables, describing constraints on the possible values and combinations of values. Relational Model is given below:

Create table Patient

```
(
        Name  varchar(20),
    Address  varchar(50),
    PID  integer not null primary key,
    DOB  date,
    Cnicvarchar(15),
)
```

Create table phone

```
        (
        P_PID  integerFOREIGN KEY references Patient (PID),
        Phone 1  varchar(15),
        Phone 2  varchar(15)
)
```

Create table VISIT

```
(
    Status  varchar(15),
    Date/time  datetime,
        VisitResipitID  integer not null primary key,
        P_PID integer references Patient(PID),
        Pres_PresID integer FOREIGN KEY reference Prescription(PresID)
    )
```

```
    Create table INSITUTE
    (
        Name  varchar(15),
        IID  integer not null primary key,
        Address  varchar(50),
        V_VisitResipitIDinteger FOREIGN KEYeferences VISIT(VisitResipitID)
    )
```

```
    Create table Prescription
            (
            PresID  integer not null primary key,
            Status  varchar(15),
            DrNamevarchar (25),
            Date/time  datetime,
            Medicine  varchar(25),
            Reference varchar(30),
            LabTestvarchar(20)
)
```

4.3 Design Rationale

The motivations behind selecting Model View Controller (MVC) design pattern include:

a. To support cohesive model definitions that focus on the domain processes, rather than on user interfaces.
b. To allow separate development of the model and user interface layers.
c. To minimize the impact of requirements changes in the interface upon the domain layer.
d. To allow new views to be easily connected to an existing domain layer, without affecting the domain layer.
e. To allow multiple simultaneous views on the same model object, such as both a tabular and business chart view of sales information.
f. To allow execution of the model layer independent of the user interface layer, such as in a message-processing or batch-mode system.
g. To allow easy porting of the model layer to another user interface framework.

For CeMS Architecture and Design, following critical issues and trade/offs that are considered:

a. Internet is available
b. Web Server is running
c. All user of CeMS are registered

Chapter 5: System Implementation

Implementation details of the project have been discussed in this chapter. The coding for the Central Server Web Server has been done in the PHP language in Adobe Dreamweaver CS5 on Apache Web Server and MySQL, while for Android Mobile Client Application for patients and doctors' tablet have been done in Android Software Development Kit (Java, XML, SQLite) in Eclipse IDE (version Helios) and the Desktop Application has been developed in Java in Netbeans (7.1) IDE. As illustrated in the prior chapter, the major applications of the system are Central server, android mobile client and desktop application divided into different Modules for flexible development and integration of inter-dependent Modules. This chapter presents the implementation details of the project.

5.1 Implementation

System implementation has been highlighted in the Figure 26: showing the overall architecture and design implementation of CeMS.

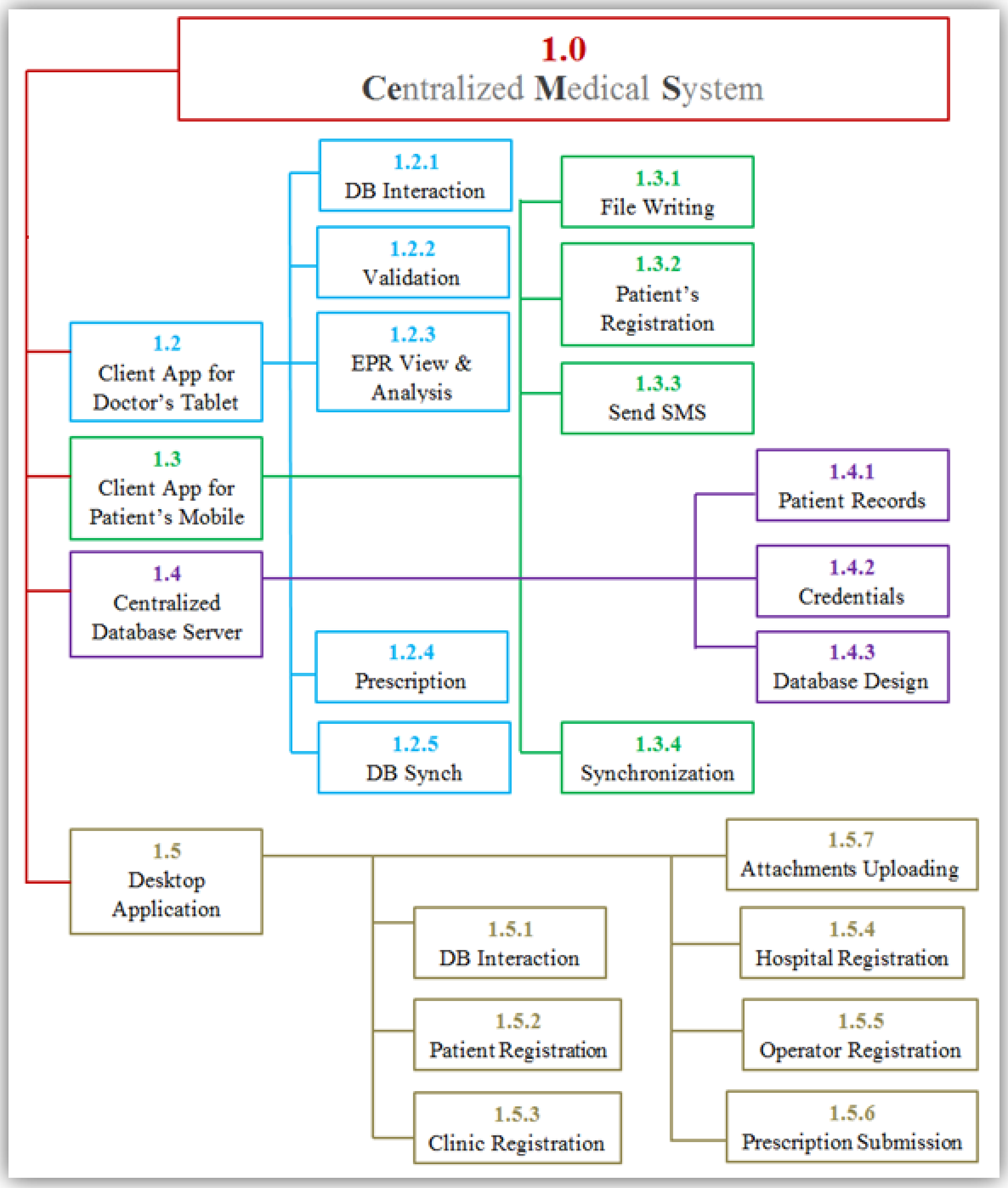


Figure 5- 1: Implementation Structure of CeMS

5.2 Implementation Language

The implementation language which has been chosen for the CeMS is JAVA. It has been preferred over other languages because:

a. Java is an object oriented language.
b. It is compiled to an intermediate language (Java Byte Code) independently of the Language it was developed or the target machine architecture and operating System
c. Automatic garbage collection
d. Open Source
e. Platform Independent
f. Pointers no longer needed (but optional)
g. Reflection capabilities
h. Definition of classes and functions can be done in any order
i. Declaration of functions and classes not needed
j. Non-existing circular dependencies
k. Classes can be defined within classes (Inline Classes)
l. There are no global functions or variables, everything belongs to a class
m. All the variables are initialized to their default values before being used
n. You can't use non-Boolean variables (integers, floats...) as conditions. This indicates it is much cleaner and less error prone language.
o. Application can be run on any OS (JRE is required), including Windows, Linux, UNIX, Mac OS, etc.

The implementation platform which has been chosen for the CeMS Mobile Phones and tablets is Android. It has been preferred over other languages because:

a. Android is an open source
b. Android does inherit all Java power into it
c. Android uses its own virtual machine, called, Delvik Virtual Machine (DVM) for optimizing the code for the mobiles
d. Automatic garbage collection
e. Android uses Java as main source coding language for the development of business logic while it uses XML for designing interactive GUIs and Layouts.

The implementation language which has been chosen for the CeMS Server side programming is PHP and for Databases “MySQL” has been used.

5.3 CeMS Modules Distribution

5.3.1 Desktop Application

This application helps administrators and operators to complete the registration process with the CeMS server of different roles/users.

5.3.1.1 Application Login

The first thing, the user gets prompted with is Application Login form asking users for the input their credentials. This module includes: admin login and operator login. This module is responsible for verifying the user through the web server.

5.3.1.2 DB Interaction

This module handles all the communication between the desktop application and server allowing the authenticated users to get the requested medical records. This module is responsible establishing the connection between the desktop application and web server.

5.3.1.3 Role Verification

Role based security model has been used to make sure the data confidentiality and integrity. In this security model every user has to have a defined role whenever trying to communicate with the web server. This module includes server side programming where users’ request have checked and processed properly. Every time a user makes a request to server, it sends a role along with the request which then first verified in the database and then on successful verification the requested contents get returned.

5.3.1.4 Patient Account Registration

This module allows the administrator to register a patient account and in result assigning a unique CeMS patient ID for identification purposes while storing all the necessary information on the server. This module includes interactive GUI for the ease of use.

5.3.1.5Operator Account Registration

This module allows the administrator to register anoperator account and in result assigning a unique CeMS operator ID for identification purposes while storing all the necessary information on the server. This module includes interactive GUI for the ease of use.

5.3.1.6 Clinic Account Registration

This module allows the administrator to register a clinic account and in result assigning a unique CeMS clinic ID for identification purposes while storing all the necessary information on the server. This module includes interactive GUI for the ease of use.

5.3.1.7Hospital Account Registration

This module allows the administrator to register a Hospital account and in result assigning a unique CeMS hospital ID for identification purposes while storing all the necessary information on the server. This module includes interactive GUI for the ease of use.

5.3.1.8Organization Account Registration

This module allows the administrator to register anorganization account and in result assigning a unique CeMS patient ID for identification purposes while storing all the necessary information on the server. This module includes interactive GUI for the ease of use.

5.3.1.9 Forget Password

This module allows the administrator and operators to reset their credentials in case they forget their credentials. But for the security purposed this module includes the users to input their email or CNIC to reset it.

5.3.1.10 Error Handling

This module makes the application more interactive and user friendly by not making users panic but guiding them in case of some wrong step or wrong input. A bottom label with red color has been used to prompt the user with appropriate message after some wrong step or input but also appropriate message dialogs have been used to display important information.

5.3.2 Android Application for Doctors' Tablet

This application facilitates the doctors in their daily patient checkups by reducing man effort of reading from paper-based medical records which be torn, in poor condition, hard to read etc. This application enhances the doctors' experience of viewing their patients' medical records, enabling them to view the all medical images on their tablets and allowing them prescribe their patients from their tablets.

5.3.2.1 Application Login

The first thing, the doctor gets prompted with is Application Login form asking users for the input their credentials. This module includes: admin login and operator login. This module is responsible for verifying the doctor through the web server, whether it is a registered doctor or not.

5.3.2.2 DB Interaction

This module handles all the communication between the doctors' tablet and server allowing the authenticated users to get the requested medical records. This module is responsible establishing the connection between the desktop application and web server.

5.3.2.3 Role Verification

Role based security model has been used to make sure the data confidentiality and integrity. In this security model every user has to have a defined role whenever trying to communicate with the web server. This module includes server side programming where users' request have checked and processed properly. Every time a doctor makes a request to server, it sends a role along with the request which then first verified in the database and then on successful verification the requested contents get returned.

5.3.2.4 Patients Basic Information and Prescription

This module presents user-friendly GUI to doctor on their tablets for viewing the patient's basic information and their last prescription.

5.3.2.5 Patients Attachments

This module presents user-friendly GUI to doctor on their tablets for viewing the patient's attachments if there are any including their X-ray images, brain MRIs, heart CT scans and so on.

5.3.2.6 Prescription

This module presents user-friendly GUI to doctor on their tablets for prescribing their patients.

5.3.2.7 SMS Receiver

This module listens for the incoming texts or SMS, in the background, from the patients mobile and on receiving the SMS, its reads it and if it is the appropriate intended text, then it extracts patient information from text and forwards it for further processing

5.3.2.8 DICOM images handling

This module handles the incoming DICOM images from the server attached to a particular patient and presenting them appropriately on the tablet

5.3.3 Android Application for Patients' Mobiles

This application presents the user with the interfaces to interact with their doctors by sending them text message.

5.3.3.1 File Writing

This module is responsible for writing the patients information into their micro SD cards in a file including their prescription ID and patient ID, last checkup date and whether attachments are required or not.

5.3.3.2 SMS/Text Messages Handler

This module is responsible for sending the patient information through send a text message after reading from the file to their doctors.

5.3.4 CeMS Centralized Server

This application has been implemented using a set of technologies. Apache has been used as our web server making our CeMS web Service available to its users 24/7. For the database server, MySQL has been used. And finally for server side programming handling all the requests from different platforms and different users, PHP has been used.

5.3.4.1 Role Verification

Role based security model has been used to make sure the data confidentiality and integrity. In this security model every user has to have a defined role whenever trying to communicate with the web server. This module includes server side programming where users' request have checked and processed properly. Every time a doctor makes a request to server, it sends a role along with the request which then first verified in the database and then on successful verification the requested contents get returned. Here PHP file receives the user request and verifies it from the database and then process accordingly.

5.3.4.2Prescription

This module processes the request on the server and sends the requested prescriptions of patients.

5.3.4.3 Attachments

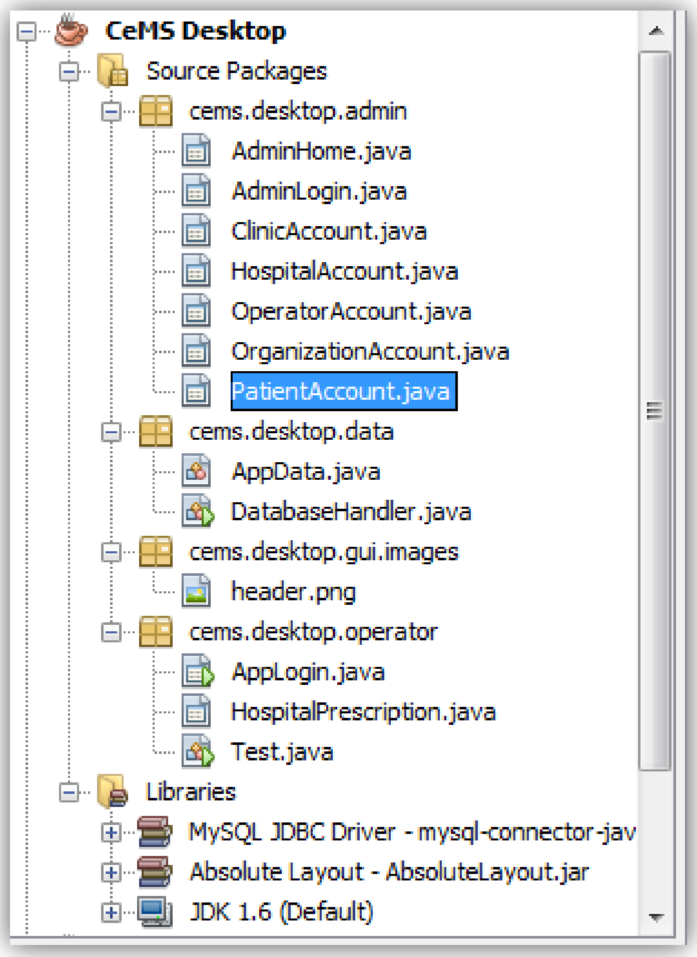
This module processes the request on the server and sends the requested attachments of patients.

5.4 Classes Distribution

Distribution of classes with respect modules distribution, as explained in 5.3, is given below.

5.4.1 Desktop Application

CeMS desktop application is composed of following classes:



```
CeMS Desktop
  Source Packages
    cems.desktop.admin
      AdminHome.java
      AdminLogin.java
      ClinicAccount.java
      HospitalAccount.java
      OperatorAccount.java
      OrganizationAccount.java
      PatientAccount.java
    cems.desktop.data
      AppData.java
      DatabaseHandler.java
    cems.desktop.gui.images
      header.png
    cems.desktop.operator
      AppLogin.java
      HospitalPrescription.java
      Test.java
  Libraries
    MySQL JDBC Driver - mysql-connector-jav
    Absolute Layout - AbsoluteLayout.jar
    JDK 1.6 (Default)
```

Figure 5- 2: CeMS Desktop Application Classes Distribution

5.4.1.1 AdminHome Class

This class is a Java jForm Form class provides the administrator with different links to navigate desired links as shown in figure below:

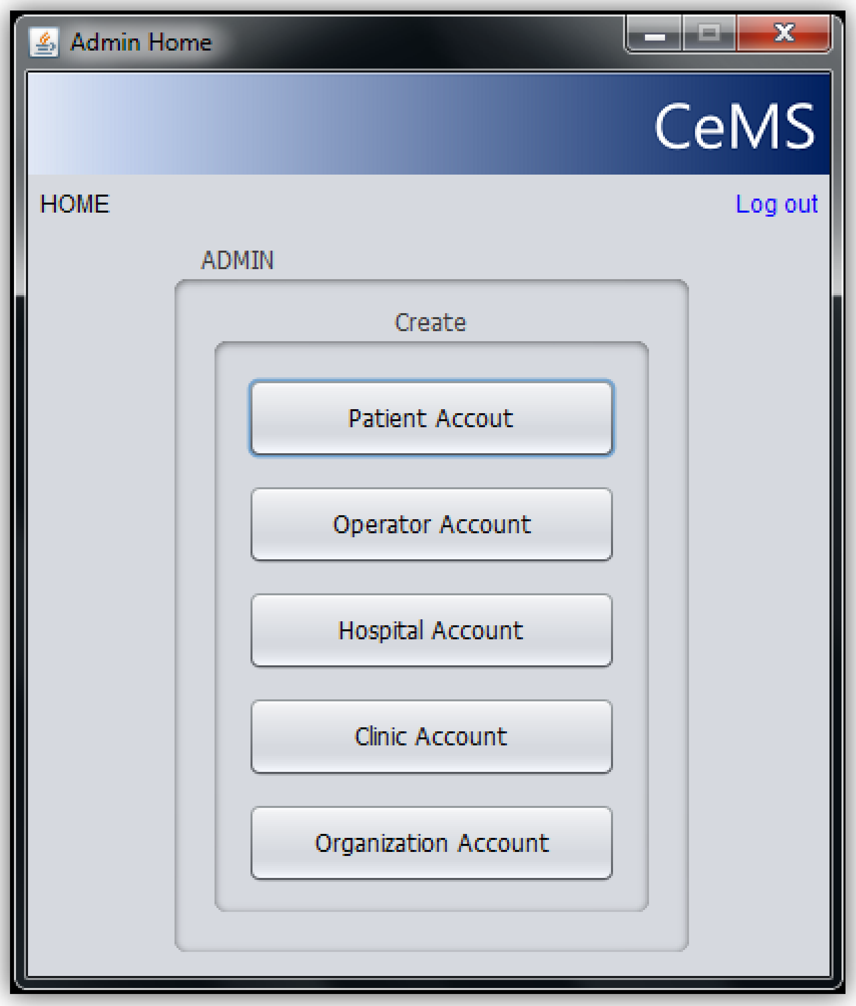


Figure 5- 3: Admin Home Class

5.4.1.2AdminLogin Class

This class is a Java jForm Form class asks the administrator to input the credentials in order to login to the system as shown in figure below:

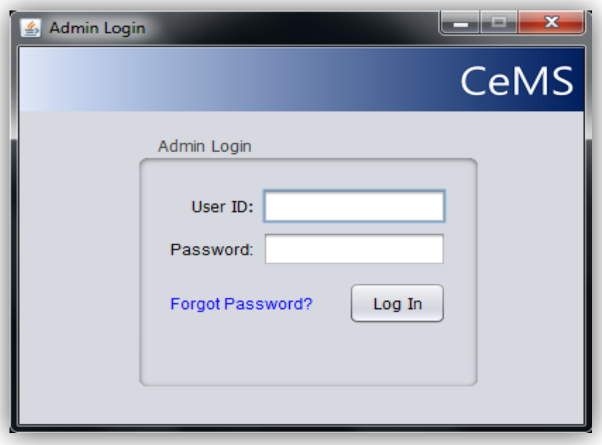


Figure 5- 4: Admin Login Class

5.4.1.3Clinic Account Class

This class is a Java jForm Form class allows the administrator to register a clinic as shown in figure below:

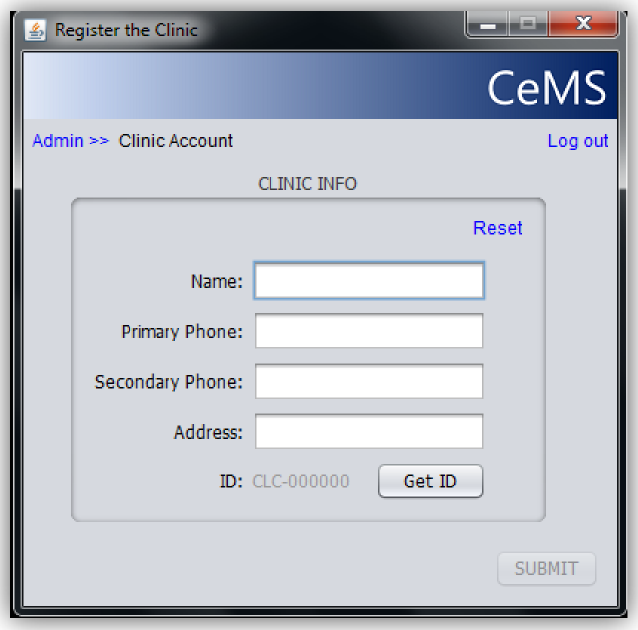
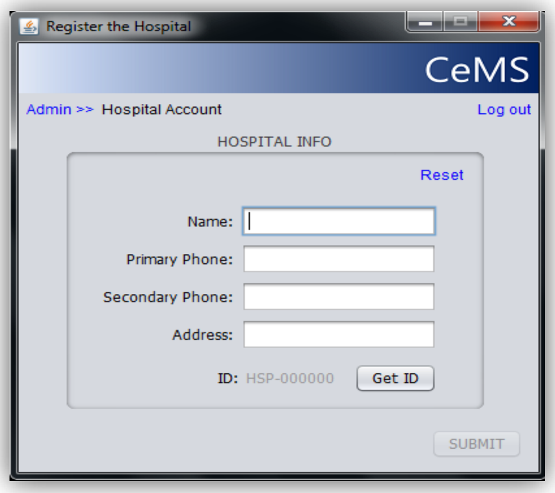


Figure 5- 5: CeMS Clinic Account

5.4.1.4 Hospital Account Class

This class is a Java jForm Form class allows the administrator to register a hospital as shown in figure below:



Register the Hospital

CeMS

Admin >> Hospital Account Log out

HOSPITAL INFO

Reset

Name:

Primary Phone:

Secondary Phone:

Address:

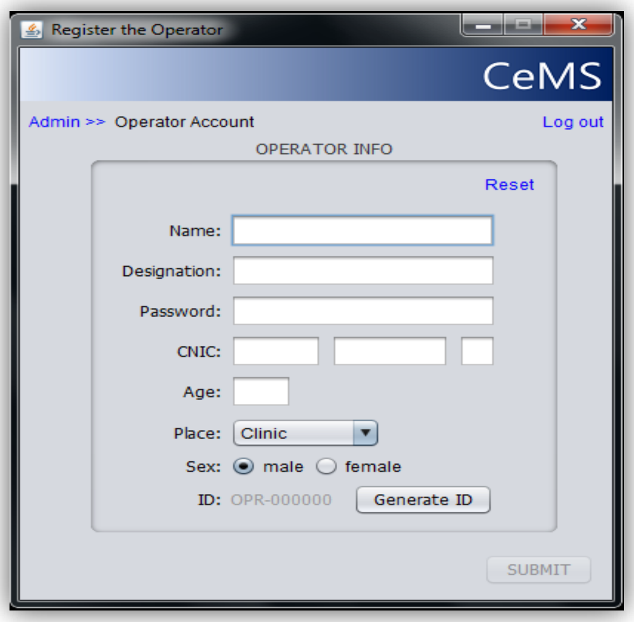
ID: HSP-000000 Get ID

SUBMIT

Figure 5- 6:CeMS Hospital Account Class

5.4.1.5Operator Account Class

This class is a Java jForm Form class allows the administrator to register an operator as shown in figure below:



Register the Operator

CeMS

Admin >> Operator Account Log out

OPERATOR INFO

Reset

Name:

Designation:

Password:

CNIC:

Age:

Place: Clinic

Sex: [✓ male] female

ID: OPR-000000 Generate ID

SUBMIT

Figure 5- 7: CeMS Operator Account Class

5.4.1.6 Organization Account Class

This class is a Java jForm Form class allows the administrator to register an organization as shown in figure below:

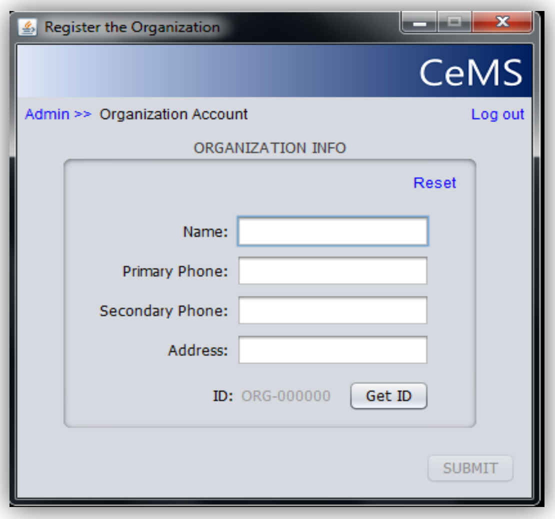


Figure 5- 8: CeMS Organization Account Class

5.4.1.7 Patient Account Class

This class is a Java jForm Form class allows the administrator to register a patient as shown in figure below:

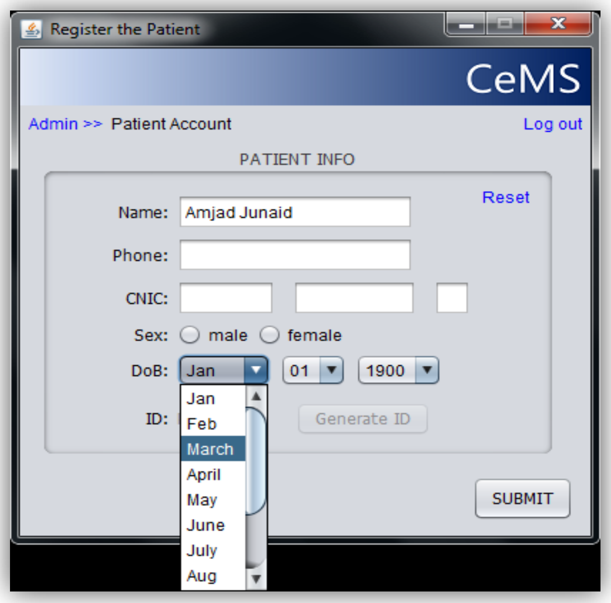


Figure 5- 9: CeMS Patient Account Class

5.4.1.8 AppData Class

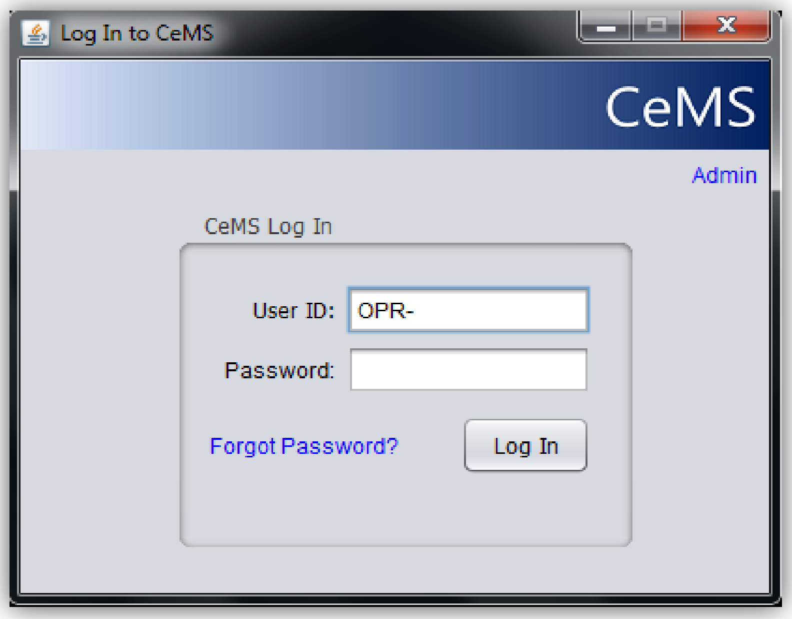
This class is a basic Java Static Class which provides the shared data between different class and procedures as well.

5.4.1.9 DatabaseHandler Class

This class is a basic Java Class which acts a handler between the GUIs and the database server. It receives the requests from the GUIs and delegate it appropriately to the server and after getting the results it returns the result to requesting GUI.

5.4.1.10 AppLogin Class

This class is a Java jForm Form class which is the main ENTRY POINT or starting point of the CeMS Desktop Application and it asks for operator to input the required credentials as shown in figure below:



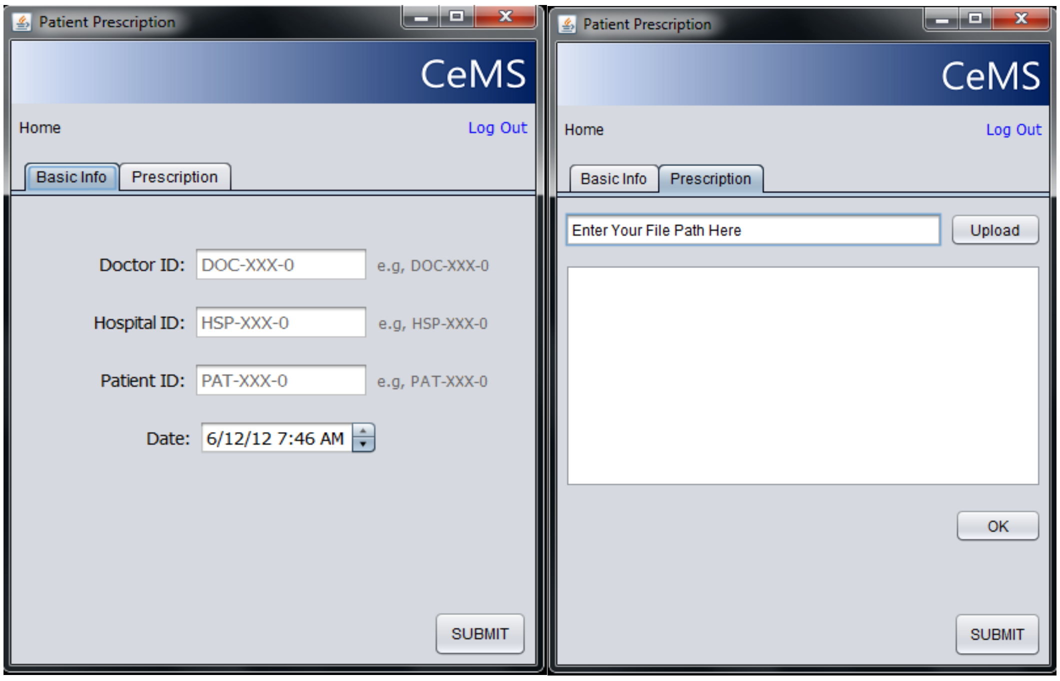
User ID: OPR-

Password:

Figure 5- 10: CeMS Application login Class

5.4.1.11 Prescription Class

This class is a Java jForm Form class which allows the operators to upload the prescriptions and attachments on the server as shown in figure below:



Doctor ID: DOC-XXX-0 e.g, DOC-XXX-0

Hospital ID: HSP-XXX-0 e.g, HSP-XXX-0

Patient ID: PAT-XXX-0 e.g, PAT-XXX-0

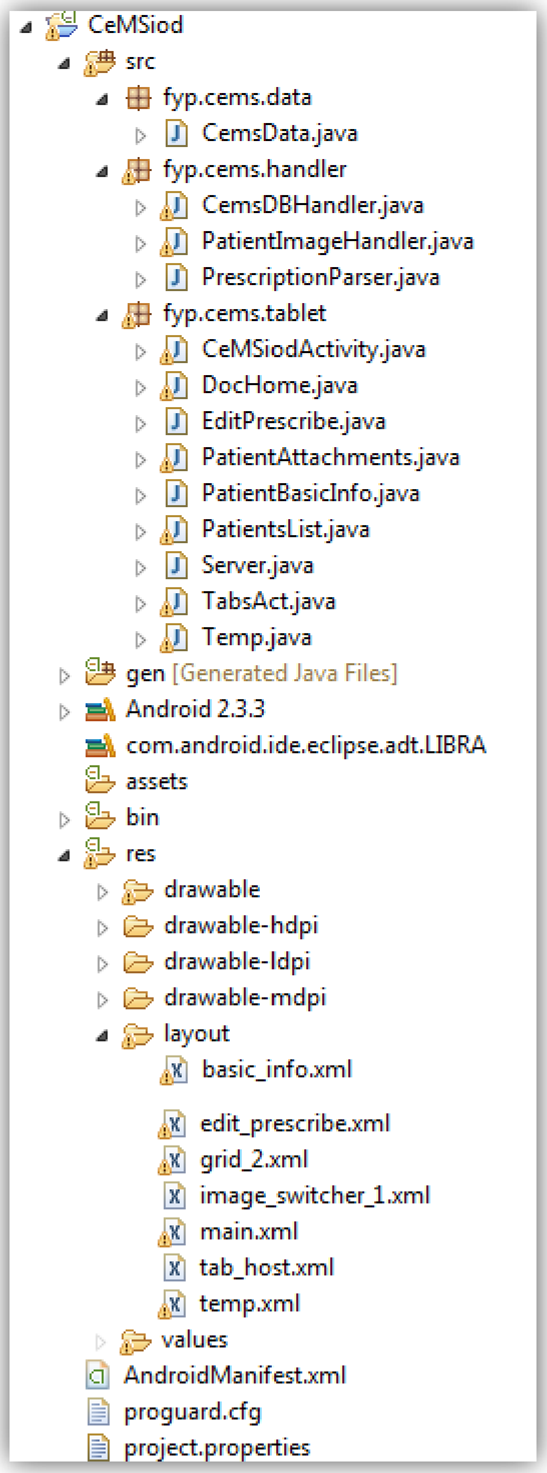
Date: 6/12/12 7:46 AM

Enter Your File Path Here

Figure 5- 11: CeMS Prescription Class

5.4.2 Android Tablet Application for Doctors

Android Tablet Application for Doctors has following classes:



```
CeMSiod
  src
    fyp.cems.data
      CemsData.java
    fyp.cems.handler
      CemsDBHandler.java
      PatientImageHandler.java
      PrescriptionParser.java
    fyp.cems.tablet
      CeMSiodActivity.java
      DocHome.java
      EditPrescribe.java
      PatientAttachments.java
      PatientBasicInfo.java
      PatientsList.java
      Server.java
      TabsAct.java
      Temp.java
  gen [Generated Java Files]
  Android 2.3.3
  com.android.ide.eclipse.adt.LIBRA
  assets
  bin
  res
    drawable
    drawable-hdpi
    drawable-ldpi
    drawable-mdpi
    layout
      basic_info.xml

      edit_prescribe.xml
      grid_2.xml
      image_switcher_1.xml
      main.xml
      tab_host.xml
      temp.xml
    values
  AndroidManifest.xml
  proguard.cfg
  project.properties
```

Figure 5- 12: CeMS Android Tablet Application Classes Hierarchy

5.4.2.1 CeMSData Class

This class is a basic Android Java Static Class which provides the shared data between different class and procedures as well.

5.4.2.2 CeMS DBHandler Class

This class is a basic Java Class which acts a handler between the Android Layouts GUIs and the database server. It receives the requests from different layouts and delegates it appropriately to the server and after getting the results it returns the result to requesting layout.

5.4.2.3Patients Images Handler Class

This class is a basic Android Java Class which acts a handler between the Android Layouts GUIs and the database server. It receives the different set of images in DICOM format and then this is class converts them appropriately and make them presentable for doctors to view images.

5.4.2.4 Prescription Parser Class

This class is a basic Android Java Class which acts a handler between the Android Layouts GUIs and the database server. It receives the formatted prescription from server response in JSON format. It then parses the received prescription and extracts the information.

5.4.2.5 CeMSoid Activity Class

This class is a basic Android Activity Java Class which is the entry point or starting activity of the application. It prompts the doctor with a layout asking for input the doctor credentials as shown below in figure 38:

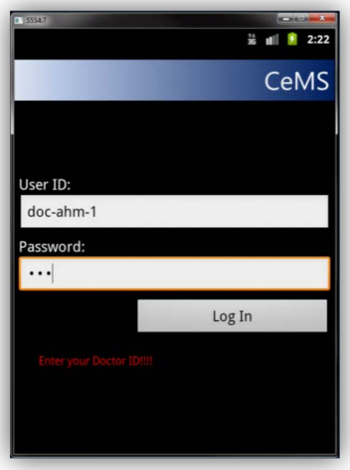


Figure 5- 13: CeMSoid Activity Class

5.4.2.6 DocHome Class

This class is a basic Android Activity Java Class which exposes its layout to the doctor for different option. But as per project requirements only check patients functionality has been implemented as shown below in figure 39:

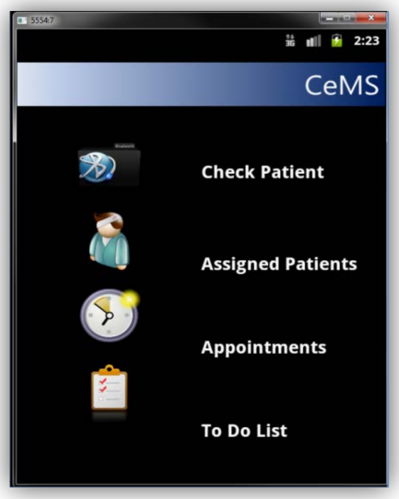
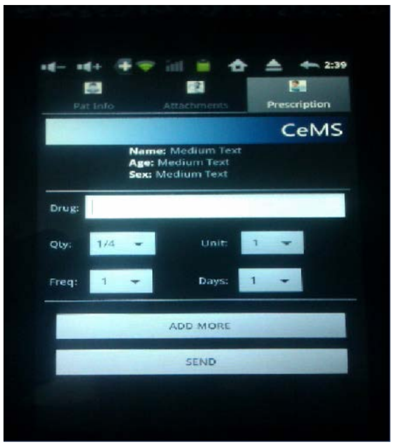


Figure 5- 14: DocHome Class

5.4.2.7 Edit Prescription Class

This class is a basic Android Activity Java Class which exposes its layout to the doctor for prescribing the patient from the tablet as shown below in figure 40:



Name: Medium Text
Age: Medium Text
Sex: Medium Text
Drug:
Qty: 1/4
Unit: 1
Freq: 1
Days: 1

Figure 5- 15: EditPrescribe Class

5.4.2.8Patients Attachment Class

This class is a basic Android Activity Java Class which exposes its layout to the doctor for viewing medical images attachments of the patient as shown below in figure 41:

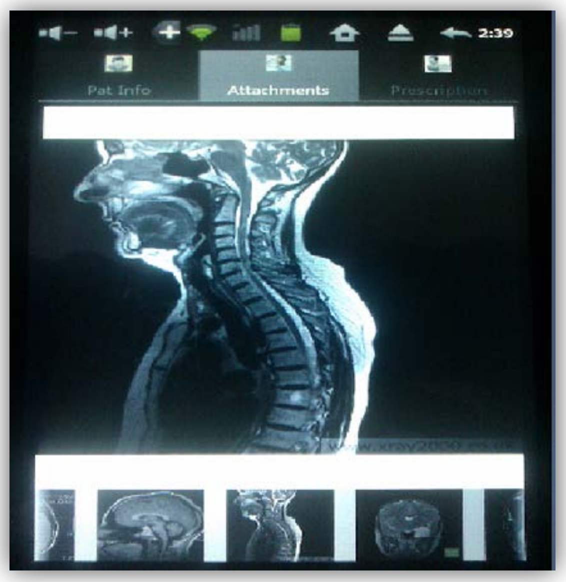
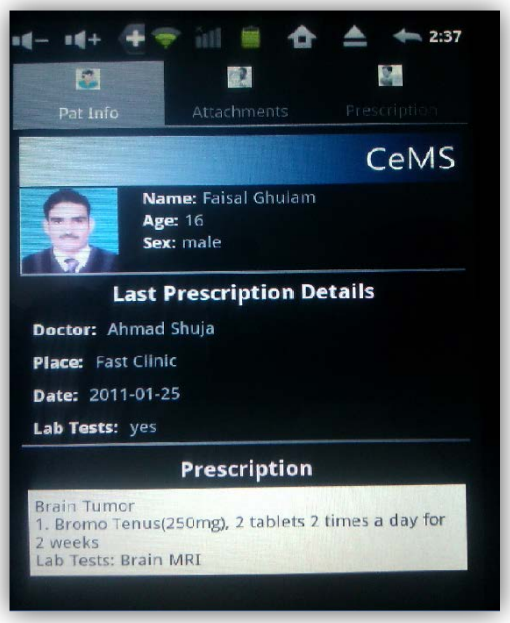


Figure 5- 16: PatientsAttachments Class

5.4.2.9 Patients Basic Info Class

This class is a basic Android Activity Java Class which exposes its layout to the doctor for viewing the basic information of the patient and previous prescriptions as shown below in figure 42:



Name: Faisal Ghulam
Age: 16
Sex: male

Last Prescription Details

Doctor: Ahmad Shuja
Place: Fast Clinic
Date: 2011-01-25
Lab Tests: yes

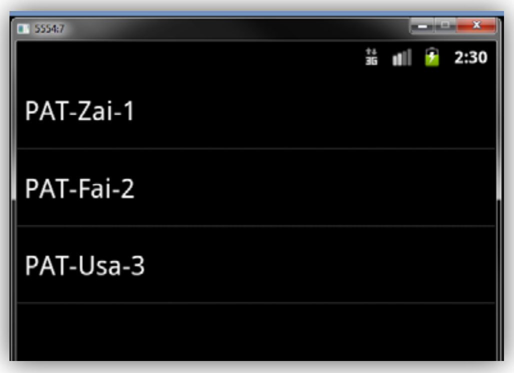
Prescription

Brain Tumor
1. Bromo Tenus(250mg), 2 tablets 2 times a day for 2 weeks
Lab Tests: Brain MRI

Figure 5- 17: PatientBasicInfo Class

5.4.2.10 Patients List Class

This class is a basic Android Activity Java Class which exposes its layout to the doctor for viewing the number patients to be checked, after receiving text messages from the patient’s android devices as shown below in figure 43.



PAT-Zai-1
PAT-Fai-2
PAT-Usa-3

Figure 5- 18: PateintsList Class

5.4.3CeMS Centralized Web Server (Web Service)

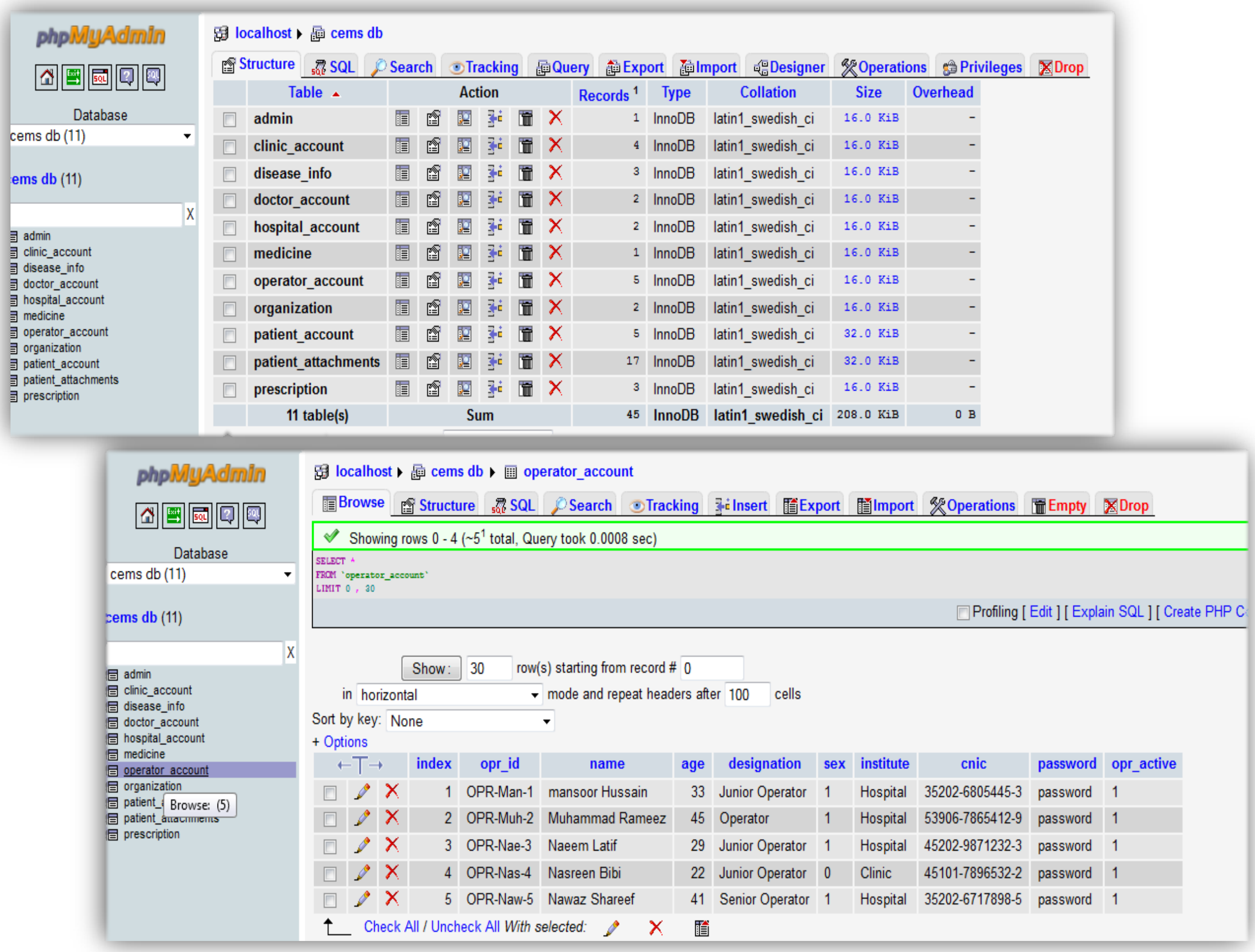
Web service is composed of:

a. MySQL database

b. Server side programming- PHP files

5.4.3.1 MySQL Database

This is basically CeMS data layer where all data or information is being stored and managed. An overview of database tables is shown below in figure 44:


localhost ▸ cems db

Structure | SQL | Search | Tracking | Query | Export | Import | Designer | Operations | Privileges | Drop

Table	Action	Records	Type	Collation	Size	Overhead
admin		1	InnoDB	latin1_swedish_ci	16.0 KiB	-
clinic_account		4	InnoDB	latin1_swedish_ci	16.0 KiB	-
disease_info		3	InnoDB	latin1_swedish_ci	16.0 KiB	-
doctor_account		2	InnoDB	latin1_swedish_ci	16.0 KiB	-
hospital_account		2	InnoDB	latin1_swedish_ci	16.0 KiB	-
medicine		1	InnoDB	latin1_swedish_ci	16.0 KiB	-
operator_account		5	InnoDB	latin1_swedish_ci	16.0 KiB	-
organization		2	InnoDB	latin1_swedish_ci	16.0 KiB	-
patient_account		5	InnoDB	latin1_swedish_ci	32.0 KiB	-
patient_attachments		17	InnoDB	latin1_swedish_ci	32.0 KiB	-
prescription		3	InnoDB	latin1_swedish_ci	16.0 KiB	-
11 table(s)	Sum	45	InnoDB	latin1_swedish_ci	208.0 KiB	0 B

localhost ▸ cems db ▸ operator_account

Browse | Structure | SQL | Search | Tracking | Insert | Export | Import | Operations | Empty | Drop

Showing rows 0 - 4 (~5 total, Query took 0.0008 sec)

```
SELECT *
FROM `operator_account`
LIMIT 0 , 30
```

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

Sort by key: None

index	opr_id	name	age	designation	sex	institute	cnic	password	opr_active
1	OPR-Man-1	mansoor Hussain	33	Junior Operator	1	Hospital	35202-6805445-3	password	1
2	OPR-Muh-2	Muhammad Rameez	45	Operator	1	Hospital	53906-7865412-9	password	1
3	OPR-Nae-3	Naeem Latif	29	Junior Operator	1	Hospital	45202-9871232-3	password	1
4	OPR-Nas-4	Nasreen Bibi	22	Junior Operator	0	Clinic	45101-7896532-2	password	1
5	OPR-Naw-5	Nawaz Shareef	41	Senior Operator	1	Hospital	35202-6717898-5	password	1

Figure 5- 19: CeMS Database Server

Chapter 6: Testing and Results Analysis

In this section of the document, functional testing, integration, system testing and load balancing test has been performed using different test cases. Software testing can also be stated as the process of validating and verifying that a software program/application/product:

- meets the business and technical requirements that guided its design and development;
- works as expected; and
- Can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

6.1 Functional Testing

“**S**oftware development organizations with an effective functional testing practice have a fast and objective way to determine whether each functional requirement is actually implemented in the code. With functional testing, the team translates functional requirements into executable test cases that confirm how well the code satisfies the requirements at any given time. It provides unprecedented objective insight into requirement status and prevents the missing or incorrect functionality implementations that can lead to countless rewrites (and then budget overruns and missed deadlines), user dissatisfaction, and project failure.”

(Functional Testing By Adam Kolawa, Co-Founder of Parasoft)

Functional testing typically involves five steps:

a. The identification of functions that the software is expected to perform.
b. The creation of input data based on the function's specifications.
c. The determination of output based on the function's specifications.
d. The execution of the test case.
e. The comparison of actual and expected outputs.

6.1.1 Functional Testing Plan

6.1.1.1Testing Goals

To validate application functionality, many of the features will be tested to ensure that those functions provide the expected output. Testing will be completed over two consecutive days. Functional testing will complete when all features have passed all associated test cases – no exceptions.

6.1.1.2 Team Members Assignment

The following resources will be completely or partially dedicated to the testing effort. The roles each will play in the testing phase.

Name	**High-level Testing Assignments**
NC Hafiz Hassan	Test case writing and execution for features
Maj. Humayun	Test case writing and execution for features
GC Mohsin Mumtaz	Test case writing and execution for features
NC Adnan Munir	Test Cases Execution
NC Zaigham Abbas	Test Cases Execution

Table 6- 1: CeMS Functional testing Team Members

6.1.1.3 Scope

This section details the features that will be included in the functional testing phase(s) and those that will be excluded.

Feature ID	**Name**
1	Login Page
2	Admin Home
3	Patient Account
4	Operator Account
5	Hospital Account
6	Clinic Account
7	Organization Account
8	Forget Password
9	Patients Prescription-Basic Info

10	Patients Prescription-Prescription

Table 6- 2: Features Set for Functional Testing of CeMS Desktop Application

6.1.1.4 Testing Approach and Tools

The following approach and tools will be used to test the application.

***T**esting will be executed with the aid of automated functional testing tool. Test cases will be developed and maintained in the tool.*

***T**he process for receiving the application for testing, and communicating errors to developers, is as follows:*

The project is using an XP approach. At the end, the application will be moved to the testing environment for functional testing. The testing cycle will be two days long. Results will be recorded using appropriate template andfaults will be communicated to the development team at the end of the testing cycle using the given form.

6.1.1.5 Functional Test Cases and their Execution

The following test cases will be created and executed against the application, given in forms of tables.

S#	Test Data	Expected O/P	O/P	Status
1	OPR-Man-1, password	Log in	Log in	Success
2	OPR-man-1,125	Invalid username or password	Invalid username or password	Success
3	Asd , password	Invalid username or password	Invalid username or password	Success
4	PAT-Adm-12,146	Invalid username or password	Invalid username or password	Success
5	Forget Password	Forget Password Page	Forget Password Page	Success

Table 6- 3: Feature ID-1(Operator Login) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	admin, 123	Admin Home	Admin Home	Success
2	admin,125	Invalid username or password	Invalid username or password	Success
3	Asd,123	Invalid username or password	Invalid username or password	Success
4	Adm,146	Invalid username or password	Invalid username or password	Success

Table 6- 4: Feature ID-2(Admin Home) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Log out	Login Page	Login Page	Success
2	Admin	Admin Page	Admin Page	Success
3	Generate ID	Enter Your Name First	Enter Your Name First	Success
4	Reset	Clear all entries	Clear all entries	Success
5	Omer, 03324567890,12345-1234567-1,male,dob, submit	ID Generated	ID Generated	Success
6	Omer,12345-1234567-1,male,dob, submit	Required Fields are empty	Required Fields are empty	Success
7	Omer, 03324567890,12345-1234567-1,male , submit	Required Fields are empty	Required Fields are empty	Success
8	Omer, 03324567890, ,male, dob, submit	Required Fields are empty	Required Fields are empty	Success
9	Nm, Generate ID	Invalid name	Invalid name	Success

Table 6- 5: Feature ID-3(Admin Login) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Log out	Login Page	Login Page	Success
2	Admin	Admin Page	Admin Page	Success
3	Generate ID	Enter Your Name First	Enter Your Name First	Success
4	Reset	Clear all entries	Clear all entries	Success
5	Health clinic, 03324567890, 03451234567, mcs Rawalpindi, submit	ID Generated	ID Generated	Success
6	Health clinic, 03324567890, mcs Rawalpindi, submit	Required Fields are empty	Required Fields are empty	Success
7	03324567890, 03451234567, mcsrawalpindi , submit	Required Fields are empty	Required Fields are empty	Success
8	Nm, Generate ID	Invalid name	Invalid name	Success

Table 6- 6: Feature ID-4(Patient Account) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Log out	Login Page	Login Page	Success
2	Admin	Admin Page	Admin Page	Success
3	Generate ID	Enter Your Name First	Enter Your Name First	Success
4	Reset	Clear all entries	Clear all entries	Success
5	Omer,operator, 123,12345-1234567-1,43,clinic,male, submit	ID Generated	ID Generated	Success
6	Omer, 123,12345-1234567-1,organization, male, submit	Required Fields are empty	Required Fields are empty	Success
7	operator, 123,12345-1234567-1 ,43,clinic,submit	Required Fields are empty	Required Fields are empty	Success
8	Nm, Generate ID	Invalid name	Invalid name	Success

Table 6- 7:Feature ID-5(Operator Account) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Log out	Login Page	Login Page	Success
2	Admin	Admin Page	Admin Page	Success
3	Generate ID	Enter Your Name First	Enter Your Name First	Success
4	Reset	Clear all entries	Clear all entries	Success
5	Shifa Hospital, 03324567890, 03451234567, mcs Rawalpindi, submit	ID Generated	ID Generated	Success
6	Shifa Hospital , 03324567890, mcs Rawalpindi, submit	Required Fields are empty	Required Fields are empty	Success
7	03324567890, 03451234567, mcsrawalpindi , submit	Required Fields are empty	Required Fields are empty	Success
8	Nm, Generate ID	Invalid name	Invalid name	Success

Table 6- 8:Feature ID-6(Hospital Account) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Log out	Login Page	Login Page	Success
2	Admin	Admin Page	Admin Page	Success
3	Generate ID	Enter Your Name First	Enter Your Name First	Success
4	Reset	Clear all entries	Clear all entries	Success
5	Institute of Health, 03324567890, 03451234567, mcs Rawalpindi, submit	ID Generated	ID Generated	Success
6	Institute of Health , 03324567890, mcs Rawalpindi, submit	Required Fields are empty	Required Fields are empty	Success
7	03324567890, 03451234567, mcsrawalpindi , submit	Required Fields are empty	Required Fields are empty	Success
8	Nm, Generate ID	Invalid name	Invalid name	Success

Table 6- 9:Feature ID-7(Clinic Account) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Logout	Log in Page	Log in Page	Success
2	DOC-XXX-1, HSP-XXX-1, PAT-XXX-1, Date, submit	Prescription is submitted	Prescription is submitted	Success
3	DOC-XXX-1, PAT-XXX-1, submit	Some Fields are missing	Some Fields are missing	Success
4	HSP-XXX-1, PAT-XXX-1, Date, submit	Some Fields are missing	Some Fields are missing	Success
5	Prescription	Prescription tab is opened	Prescription tab is opened	Success

Table 6- 10:Feature ID-8(Operator Account) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Cancel	Log in Page	Log in Page	Success
2	12345-1234567-1. OK	Lost password is generated	Lost password is generated	Success
3	OK	Invalid CNIC	Invalid CNIC	Success
4	12345-14567-1. OK	Invalid CNIC	Invalid CNIC	Success

Table 6- 11:Feature ID-9(Forget Password) Test Cases and Result

S#	**Test Data**	**Expected O/P**	**O/P**	**Status**
1	Logout	Log in Page	Log in Page	Success
2	Upload file, description, OK, submit	Prescription is submitted	Prescription is submitted	Success
3	OK	Some Fields are missing	Some Fields are missing	Success
4	Upload file, OK, submit	Some Fields are missing	Some Fields are missing	Success
5	Basic Info	Basic Info tab is opened	Basic Info tab is opened	Success

Table 6- 12:Feature ID-10(Patient Prescription) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Cancel	Admin Log in Page	Admin Log in Page	Success
2	asdfg@hotmial.com OK	Lost password is generated	Lost password is generated	Success
3	OK	Invalid email Address	Invalid email Address	Success
4	amSSdsa@sdnc@dma OK	Invalid email Address	Invalid email Address	Success

Table 6- 13:Feature ID-9(Forget Password) Test Cases and Result

For ***android tablet application for doctors***we have following feature set:

Feature ID	Name
1	Login Page
2	Check Patient
3	Patient’s Attachments
4	Patient’s Prescription

Table 6- 14: Features Set for Functional Testing of Android Application for Doctors' Tablet

S#	Test Data	Expected O/P	O/P	Status
1	DOC-Man-1, password	Log in	Log in	Success
2	DOC-man-1,125	Invalid username or password	Invalid username or password	Success
3	Asd , password	Invalid username or password	Invalid username or password	Success
4	PAT-Adm-12,146	Invalid username or password	Invalid username or password	Success

Table 6- 15:Feature ID-1(Doctor Login) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Panadol,1,1,3,3,send	Message sent to Patient	Message sent to Patient	Success
2	Panadol,1,1,3,3, add more, send	Message sent to Patient	Message sent to Patient	Success

Table 6- 16:Feature ID-4(Patient Prescription) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Click on attachment (when there are attachments)	Show attachments	Show attachments	Success
2	Click on attachments(where there are not any attachment)	No attachments	No attachments	Success

Table 6- 17:Feature ID-3(Patient Attachments) Test Cases and Result

S#	Test Data	Expected O/P	O/P	Status
1	Patient ID	Patients Info	Patient Info	Success

Table 6- 18:Feature ID-2 (check patient) Test Cases and Result

For ***android mobile application for patients*** we have following feature set:

Feature ID	**Name**
1	Send text
2	Registration

Table 6- 19: Feature Sets for Functional testing of android mobile application of patients

S#	**Test Data**	**Expected O/P**	**O/P**	**Status**
1	03321234567,PAT-Man-1, Prescription ID, date	Message send to doctor	Message send to doctor	Success

Table 6- 20:Feature ID-1(send text) Test Cases and Result

S#	**Test Data**	**Expected O/P**	**O/P**	**Status**
1	03321234567,PAT-Man-1, Prescription ID	Message send to operator for registration	Message send to operator for registration	Success

Table 6- 21:Feature ID-2(registration of patient) Test Cases and Result

6.2 Integration Testing

6.2.1 Modules to Be Integrated

We have following modules to integrate:

- Login
- Prescribe(desktop/android)
- Schedule
- Update medical record
- View medical record

6.2.2 Entry Criteria

- Code is reviewed and approved.
- Unit test plan for each module is written, reviewed and executed.
- All of the unit tests passed.

6.2.3 Exit Criteria

- **A**ll code is completed and frozen and no more modules are to be integrated.
- **A**ll of the system integration tests passed.
- **N**o major defect is outstanding.
- **A**ll the moderate defects found in SIT phase are fixed and retested.

6.2.4 System Integration Method

Big Bang technique has been used to integrate the system. In the big-bang approach, first all the modules are individually tested. Next, all those modules are put together to construct the entire system which is tested as a whole.

6.2.5 Test Specification for Integration

Following are given test specifications for integration testing:

REQ-1	
1.1	Testing Login of users
Purpose:	To test whether system has the functionality to log in the users
Pre-Requisite:	Server is running
Test Data:	Any character from a-z and A- Z in any amount Integers (0-9).
Steps:	Run the application
Status:	Success

Table 6- 22: Integration Test Specification for Requirement of proper login functionality

REQ-2	
2.1	To prescribe the patient’s treatment
Purpose:	To test whether system has features for doctor’s android application and operators desktop application to write prescriptions.
Pre-Requisite:	Doctor and Operator must be registered to the system and they are logged in as well
Test Data:	Any set of characters
Steps:	Prescriptions are then transferred to the server database
Status:	Success

Table 6- 23:Integration Test Specification for Requirement of proper login functionality

Chapter 7: Conclusion and Future Work

This chapter describes the possible future enhancements in the project. As it is a desktop application comprising of three basic components, there are a lot of enhancements that can be done in order to make the system more effective and efficient in the face of providing flexibility and ease of inter-organization communication.

7.1 Commercial Application and Uses

a. It will provide the facility, to patients, of keeping their medical records safe in their micro SD cards of their mobile saving the cost of extra purchasing of files and folder, X-Rays cards and reports etc.
b. CeMS Tablet Application for Doctors provides the facility to view medical records, their X-Rays, Brain MRIs and other lab tests reports with ease of use on their tablets, saving their time and reduces the pain of analyzing old, torn and poor conditioned medical prescriptions
c. With having a centralized server, there is no need to be worried about the loss of patient records which are paper-based.
d. If deployed properly, then most of medical institutes (hospitals, clinics, medical universities and colleges, medical labs, medical stores etc.) in Pakistan can be integrated together, collaborating with each other and accessing medical records for different purposes.
e. With the CeMS unique mechanism of interoperating using CeMS IDs makes it reliable and trustworthy for data confidentiality and integrity. This provides the facility of medical records available 24/7
f. With CeMS flexible architecture, it is very convenient to get integrate the functionality with International Medical Standards

7.2 Conclusion

Currently, in Pakistan, medical records are being maintained on papers. Patients lose their records very frequently or they get old after a specific time. Usually when patients visit their family doctor or new doctor, they are not in possession of their previous medical reports, including X-Rays, medical tests etc. Also patients, doctors and hospital staff find it very difficult to process the paper based medical records including their maintenance.

SoCeMS is a sincere effort to develop a system which makes an individual's medical records easily available. Patients' complete medical records will be saved on the Centralized Server. Also in the external storage of patient's mobile; Perception ID, Date are stored for making it a carrier of keys for accessing the prescriptions. From the external storage, it will be transferred to doctor's tablet (Android-based) through sending a text message using GSM technology. Similarly the doctor's prescription will be synchronized with the server and in micro SD card only relevant IDs get stored. All the medical records will be maintained in central database which can be made easily available to any doctor or any hospital anywhere in Pakistan. So just take along your mobile while going to doctor and all will be ok.And if you are brought in a hospital unconscious, your record will be found through CeMS Unique IDs.

7.3 Future Work

7.3.1 Statistical Analysis

CeMS can be extended in a way to help the

- Government institutes
- Commercial organizations
- Educational institutes
- Pharmaceutical companies

to conduct or perform different kind of statistical analysis on patient records or prescriptions of patients use data or also can conduct a research on various aspects of health related issues.

Some examples of such analysis:

a. Some pharmaceutical companies can plot different graphs of analysis describing in what exact quantity their medicine has been used in past or being used at present in specific areas etc.
b. Government organization can conduct analysis like how many death have been occurred due to dengue Disease in some year say 2010 in Lahore.

7.3.2 Prediction of Future Disease Trends

With having a CeMS web service as Centralized database server can help to predict future of some disease,keeping in view rapid increase in no of patients of a certain disease.

7.3.3 Availability of data

Central data can be used by various organizations to tailor their use as colleges for training purpose,drug store keepers to observe demand and supply factor etc.

7.3.4 Incorporation of HL7

Present CeMS can be incorporated wit HL7, for implementing a complete hospital management system, acting as a module for maintain prescriptions.

7.3.5 Integrity and Confidentiality

Well known security mechanisms and web server security mechanisms can be applied to system to ensure the data integrity, confidentiality and availability.

7.3.6 VirtualizedStorage for Medical Records:

Virtual storage technology can ease the burden of medical images placed on CeMS database server.

7.3.7 Use of Storage Monitoring Tools

Storage monitoring tools from Virtual Instruments can be used to help alleviate application crashes and failures.

7.3.8 Reducing Disk Costs

Compel lent Data Progression automated making tiers software expands data storage through cheaper tier 3 SATA disks; offers making tiers and sub-tiers level options.

7.3.9 Cloud Services

Cloud services let health care providers move storage, data processing and other systems onto the Internet, provided that security and data ownership concerns are addressed.

7.3.10 Alternates for micro SD Card

Subscribers Identity Module (SIM)can also be used as memory storage device instead of micro SD, which does have space of 60K. Similarly RFID Cards (read by barcodes easily) can also be used at patient records carrier.

Appendix A: Glossary

EHR:	An electronic health record (EHR) is digital documentation of an individual's medical history that is maintained by health professionals and official agencies.
EPR:	Electronic patient record is defined as the file stored on a computer, which records all the vital information about the patient's current health and history.
HL7:	Accredited standards for electronically defining clinical and administrative data in the healthcare industry from Health Level Seven International
PR:	Patient Record is a collection of documents that provides an account of each episode in which a patient visited or sought treatment and received care or a referral for care from a health care facility
micro SD:	A type of removable flash memory card designed specifically for mobile phones.
Wi-Fi:	Short for "Wireless Fidelity." Wi-Fi refers to wireless networking technology that allows computers and other devices to communicate over a wireless signal.
Web Service:	A method of communication between two electronic devices over the Web (Internet).
Centralized Server:	A type of network where all users connect to a central server, which is the acting agent for all communications.
JRE:	Java Virtual Machine (JVM) enables a set of computer software programs and data structures to use a virtual machine model.

JDK:	The Java Development Kit (JDK) is a Sun Microsystems product aimed at Java developers.
MRI:	Magnetic Resonance Imaging, uses magnetic signals to create image "slices" of the human body.
Netbeans:	Netbeans is a free IDE that can be used to create Java and JavaFX applications.
IDE:	Integrated Drive Electronics is a standard electronic interface used between a computer motherboard's data paths or bus and the computer's disk storage.
SCR:	It is an electronic record which contains information about the medicines you take, allergies you suffer from and any bad reactions to.

Appendix B: References

Books:

[1] Java How to Program, 6th edition, By Deitel and Deitel

[2] Learning Java, 3rd Edition, By Jonathan Knudsen, Patrick Niemeyer

[3] Java generics and Collections, By Maurice Naftalin, Philip Wadler

[4] Java Network Programming by Richard Blum

[5] Pro Android 3, by SatyaKomatineni, Sayed Y. Hashimi

[6] Android User Interface Development: Beginner's Guide, By Jason Morris

[7] The Busy Coder's Guide to Android Development, By Mark L. Murphy

[8] Hello Android Introducing Google's Mobile Development Platform, 3rd Edition by Ed Burnet

[9] PHP and MySQL Web Development (4th Edition)by Luke Welling and Laura Thomson.

[10] Learning PHP, MySQL, and JavaScript: A Step-By-Step Guide to Creating Dynamic Websites (Animal Guide)by Robin Nixon.

FORMS:

[11] http://www.anddev.org/sdk-adt-emulator-problems-f16/http://www.daniweb.com/forums/thread91293.html

[12] http://bytes.com/topic/c-sharp/answers/225292-how-trap-x-button-windows-form-c

[13] http://www.dreamincode.net/forums/index.php?showtopic=90279

SERIALIZABLE:

[14] http://www.osix.net/modules/article/?id=692

[15] http://social.msdn.microsoft.com/Forums/en-US/csharpgeneral/thread/59f1eccd-4c9b-4c4e-bfc8-9bc685719964

FILING:

[16] http://devhood.com/Tutorials/tutorial_details.aspx?tutorial_id=400

[17] http://www.codeproject.com/KB/dotnet/Surrogate_Serialization.aspx

MINIMIZE TO TRAY:

[18] http://www.dreamincode.net/forums/index.php?showtopic=116283

SOCKET ERROR PROBLEM:

[19] http://bytes.com/topic/net/answers/625867-c-udp-only-one-usage-each-socket-address-protocol-network-address-port

LISTBOX:

[20] http://www.java2s.com/Code/CSharp/GUI-Windows-Form/ClearallitemsinaListBox.htm

TIMESPAN:

[21] http://www.dotnetspider.com/resources/458-How-find-difference-between-two-Dates-C-or.aspx

LOCKING:

[22] http://www.albahari.com/threading/part2.aspx

PHP:

[23] http://www.w3schools.com/php/

[24] http://phpbuilder.com/columns/scott_clark09242009.php3?aid=1618

[25] http://www.devshed.com/c/a/PHP/Socket-Programming-With-PHP/

[26] http://social.msdn.microsoft.com/Forums/en-US/netfxnetcom/thread/c18dae07-5f7e-4338-9202-2e7690a0cc3a

[27] http://www.ibm.com/developerworks/opensource/library/os-php-ide/index.html

[28] http://www.killersites.com/dreamweaver/dreamweaver-cs4.php

[29] http://www.adobe.com/devnet/dreamweaver/articles/setting_up_php.html

[30] http://www.adobe.com/devnet/dreamweaver/articles/first_dynamic_site_pt2_03.html

[31] http://www.php.net

APIs JAVA:

[32] http://www.andrejkoelewijn.com/wp/2008/12/30/using-google-talk-from-java-example/

[33] https://developer.skype.com/wiki/Java_API

[34] http://jymsg9.sourceforge.net/

[35] http://sourceforge.net/apps/trac/java-jml

MYSQL:

[36] http://www.tizag.com/mysqlTutorial/mysqltables.php

[37] http://sql-info.de/mysql/examples/CREATE-TABLE-examples.html

Java MYSQL:

[38] http://bitdaddys.com/MySQL-ConnectorJava.html

[39] http://www.geekpedia.com/tutorial139_Connecting-to-MySQL-with-Csharp-and-ODBC.html

[40] http://dev.mysql.com/downloads/connector/net/1.0.html 5.0.9

DATATABLES:

[41] http://dotnetperls.com/datatable-use

[42] http://social.msdn.microsoft.com/forums/en-US/csharpgeneral/thread/342c39b3-43fb-44dc-8501-d8104e677692

DISABLE BUTTON:

[43] http://www.victorchen.info/disable-a-button-after-a-click-in-c/

XML SERIALIZER ISSUE:

[44] http://www.boards.ie/vbulletin/showthread.php?p=63524587

[45] http://social.msdn.microsoft.com/Forums/en-US/netfxcompact/thread/6f0d849f-61b1-4cb1-9801-ca38bd769009

[46] http://social.msdn.microsoft.com/forums/en-US/netfxremoting/thread/40b4df3a-2fcb-44b9-b3f7-09a01a2ca6f9

SMS Interceptor:

[47] http://msdn.microsoft.com/en-us/library/bb932385.aspx

SMS SEND:

[48] http://www.devx.com/wireless/Article/38571/1954

SMS INTERCEPTOR SAMPLE:

[49] http://suniljagadish.wordpress.com/category/windows-mobile/

[50] http://hi.baidu.com/pinggudian/blog/item/4f871122bdc2e9ae4723e841.html

LOCK MYTH:

[51] http://www.dotnetconsult.co.uk/weblog/PermaLink.aspx/870417fa-dc59-4c8d-9dad-a68b98b24457

Problem Solution Reference:

[52] http://32feet.net/forums/t/2518.aspx

YAHOO MESSENGER:

[53] http://www.ycoderscookbook.com/

SMS INTERCEPTOR PROBLEM STOPPING:

[54] http://social.msdn.microsoft.com/Forums/en-SG/netfxcompact/thread/b63a8ca6-f7a6-40d9-99a0-9bc1315c1481

[55] http://www.eggheadcafe.com/software/aspnet/32007167/messageinterceptor-in-wm5.aspx

[56] http://social.msdn.microsoft.com/Forums/en/vssmartdevicesvbcs/thread/f2b9378a-66ba-4095-ba1a-bab9d0a26902

[57] http://mavenlog.wordpress.com/2009/03/24/no-vpc-network-adapter-enumerated-or-no-host-network-adapter-with-provided-mac-address-found/

[58] http://blogs.msdn.com/b/akhune/archive/2005/11/16/493329.aspx

BAD ERROR PROBLEM IN ENCRYPTION:

[59] http://windows-tech.info/13/1533b5d3f9268d44.php

[60] http://www.eggheadcafe.com/community/aspnet/2/10052950/rsa-asymmetric-bad-data.aspx

XML SERIALIZATION DESERIALIZATION:

[61] http://bytes.com/topic/net/answers/177728-deserialize-abstract-object

[62] http://bytes.com/topic/net/answers/177700-how-do-i-serialize-object-string-instead-stream

[63] http://social.msdn.microsoft.com/forums/en-US/csharplanguage/thread/5d08bc28-5b61-4c5a-8c4b-4665b1c929ea

DATED FILE NAMES:

[64] http://dotnetperls.com/filename-datetime

FORMAT EXCEPTION:

[65] http://stackoverflow.com/questions/710853/base64-string-throwing-invalid-character-error