Spring 2016

Practice Management & Patient Health Record System

Mohammed Abdul Qadeer Aali
Governors State University

Sameer Khan
Governors State University

Arun Pallam
Governors State University

ABSTRACT

General Physicians or Specialists who has less number of patients often cannot afford to buy high end Practice management software which often costs thousands of dollars. Online Personal health record (PHR) sharing system is an emerging patient oriented model created for health information exchange following all HIPAA regulations, Using this application patient can share his/her health record with organizations (Hospitals, Health insurance firms.. etc.). It is an application which helps connect doctors and billers together and provides hassle free experience and it does not need any additional plugins or hardware and can be accessed from any device like personal computer, laptop or tablets instantly.

In this application we are providing two registration forms, one for Patient and another for organization, unless patient or organization registers they will not be able use this application. If patient wants to share information with a third person who is not registered with the application they can do it so without any fright of exploitation of their record as the files would be encrypted and can only be accessed by the key. This key is system generated and it is generated at the time of patient sending his/her Health Record.

For the patients, it gives the pleasure of staying at ease about their scheduled appointments with the doctor also providing them a platform to save their medical history at one organized easily accessible place to share, study and review.

For the medical billers, PHR is a boon which saves the staff a lot of time to access the record and bill the patients' insurances effectively complying with HIPAA norms.

Table of Content

1. Project Description

As a child, we all might have noticed doctors taking notes and keeping manual records to keep track of patient's diagnosis and illness. This helps doctor understand the patient better as he by looking at the records the doctor can understand the basic nature of patient's illness and then provide the necessary medications. This copy most of the time is with doctor unless patient needs the records for some purposes. In early 90's, it was not really necessary for the patient to keep their medical records.

But with the growing technology and with the advancement in medical field it became necessary to involve several organizations when it comes to patient health records. Organizations like hospitals, insurance companies, law offices and even the patient needs health records to better understand the reasons particular diagnostic condition. This Patient Health Record (PHR) system is designed to be patient oriented meant to exchange patient health related information via web at a single place. This model allows patient to manage personal health related data with options of creating, updating and deleting information and due to which it is easier to store, share and retrieve the health information efficiently. As mentioned earlier that this is patient oriented design, it gives the full access control to patients to share personal health information with the persons we authorize to share our personal information with, for example: father, mother, brother, sister, wife etc. and also organizations such as doctor's office, hospitals, insurance companies etc.

With the increasing costs pertaining to medical industry, many healthcare providers choose a source which is reliable and cost effective. Most of the time organizations end of outsourcing these services in order to reduce the load of information security and risk associated with it along with cost. But often these services cannot provide login to normal users or patients and it is only provided to organizations. Designing a handy Patient Health Records sharing system looks really interesting but it comes with many security risks associated with it.

This PHR system will make sure that it is convenient to use for all type or organizations including patient itself along with necessary security requirements and HIPAA being one of the standard kept in mind while designing this system.

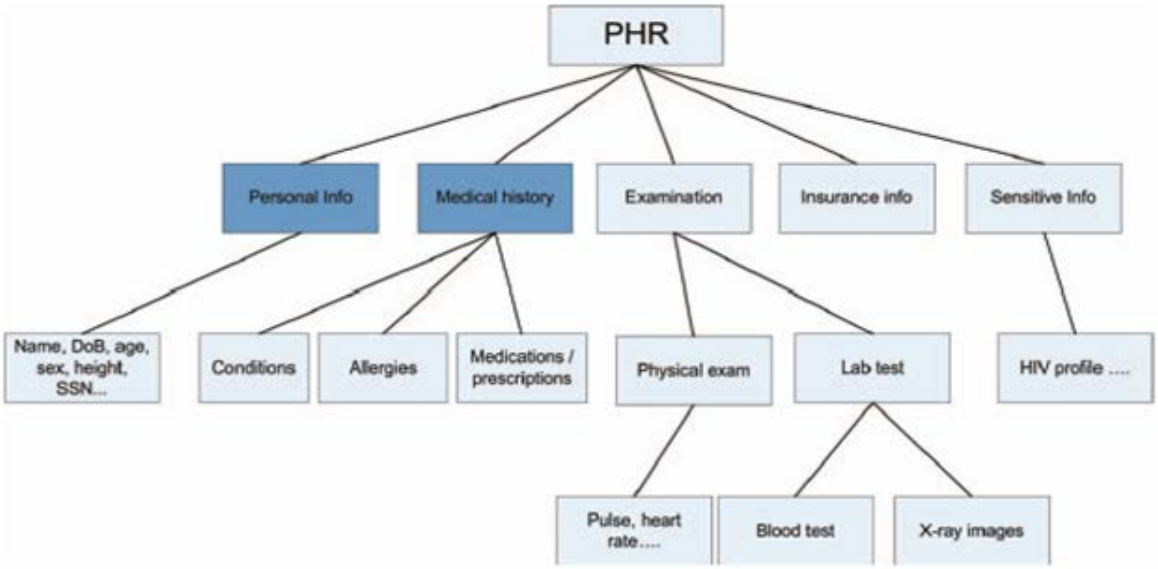
1.1 Competitive Information

Organization Module:

Organizations have to ensure that they maintain the confidential data of the patients complying with HIPAA. And to ensure the confidentiality we utilize encryption. Upon sharing the document, to ensure the safety both the recipient and sender use encryption keys.

Attribute based Access Policy Module:

In our framework, there is opportunity for multiple Organizations, multiple owners, multiple Health companies, and multiple users. We term the users having read and write access as data readers and contributors, respectively.



1.2 Relationship to Other Applications/Projects

Amid investigation, the emphasis is on what should be done, autonomous of how it is finished. Amid outline, choices are made about how the issue will be comprehended, first at abnormal state, then at progressively point by point levels.

Framework outline is the principal plan stage in which the essential way to deal with taking care of the issue is chosen. Amid framework plan, the general structure and style are chosen. The framework engineering is the general association of the framework into parts called subsystems.

The engineering gives the setting in which more itemized choices are made in later outline stages. By settling on abnormal state choices that apply to the whole framework, the framework originator parcels the issue into subsystems so that further work should be possible by a few creators working freely on various subsystems.

The framework creator must settle on the accompanying choices:

- Organize the framework into subsystems.
- Identify the simultaneousness intrinsic in the issue.
- Allocate subsystems to processors and assignments.

- Choose a methodology for administration of information stores.
- Handle access to worldwide assets.
- Choose the usage of control in programming.
- Handle limit conditions.
- Set exchange off needs.

Breaking the System into Subsystems:

The initial phase in framework configuration is to separate the framework into little number of segments. Every significant segment of a framework is known as a sub framework. Every subsystem envelops parts of the framework that share some basic property – comparative usefulness, the same physical area, or execution on the same sort of equipment.

A subsystem not an article nor a capacity but rather a bundle of classes, affiliations, operations, occasions, and requirements that are interrelated and that have a sensibly all around characterized and little interface with different subsystems. A subsystem normally recognized by the administrations it gives. An administration is a gathering of related capacities that share some regular reason, for example, I/O handling. A subsystem characterizes a rational method for taking a gander at one part of the issue.

Every subsystem has a very much characterized interface to whatever is left of the framework. The interface indicates the type of all collaborations and the data stream crosswise over subsystem limits yet does not determine how the sub framework is actualized inside. Every subsystem then can be outlined freely without influencing the others.

The decay of frameworks into subsystems might be composed as an arrangement of level layers or vertical segments.

A layered framework is a requested arrangement of virtual universes, each implicit terms of the ones beneath it and giving the premise of usage to the ones above it. The items in every layer can be autonomous, in spite of the fact that there is regularly some correspondence between articles in various layers. Information is one-way just: a subsystem thinks about the layers underneath it, however has no learning of the above layers. Every layer may have its own arrangement of classes and operations. Every layer is executed as far as the classes and operations of lower layers.

Layered engineering comes in two structures: shut and open. In a shut engineering, every layer is manufactured just regarding the prompt lower layer. In an open engineering, a layer can utilize elements of any lower layer to any profundity.

We decayed our framework into three subsystems as layers. The three layers are shut design structure. The three layers are GUI layer, Network layer and I/O layer. The reason for GUI layer

is to give a productive client interface to the client to cooperate with the framework. It is based upon the Network layer which gives fundamental FTP administrations. The most minimal layer is the I/O layer that gives administrations like perusing or composing document to and from neighborhood and remote frameworks.

At the point when top-level subsystems are recognized, the creator ought to demonstrate the data stream among the sub frameworks. There are a few design structures that are basic in existing frameworks. They are cluster change, persistent change, intelligent interface, dynamic reenactment, and constant framework and exchange administrator.

In the compositional structures determined over, our framework will best suit in intuitive interface engineering, subsequent to there are substantial number of associations amongst framework and client.

An intuitive interface is a framework that is overwhelmed by collaborations between the framework and outer operators, for example, people, gadgets or different projects. The outer operators are autonomous of framework, so their inputs can't be controlled, in spite of the fact that the framework may request reactions from them.

1.3 Assumption and Dependencies

Amid the investigation, all communications are appeared as occasions between articles. In any case, the framework creator must pick among a few approaches to actualize control in programming. There are two sorts of control streams in a product framework: inward and outer. Outside control is the stream of remotely unmistakable occasions among the articles in the framework. There are three sorts of control for outer occasions: strategy driven, occasion driven successive and simultaneous. Interior control is the stream of control inside a procedure.

1.4 Future Enhancements

Our framework falls in outside control stream like occasion driven control. Occasions are let go by outside specialist, for example, client in various request yet successive not simultaneous.

1.5 Definitions and Acronyms

DFS	Distributed File System
IP	Internet Protocol
DBMS	Data base management system
SQL	Structure query language
UDF	User defined function

2 Project Technical Description

In this project we have got total number of 29 pages including the response pages which will be displayed upon the execution of the query. Upon executing the project, we will have the index page which will show home page, organization sign in, User access and PHR owner sign in the menu list of the web application. In the organization sign in page there is an option to either signup or if the user is also created then only sign in is needed.

Once we sign in we can view the messages from other organizations or patients along with the reply option. The uploaded documents by the current user can also be viewed in the main menu of organization login.

There is also access for patients and patient can share documents with organizations and also with other users generating encryption key. By entering the encryption key and the email address the user can access the file.

3 Project Requirements

3.1 Identification of Requirements

This section provides a brief explanation of the use of named and enumerated requirements to identify and number requirements. For each requirement, please use the following SMART criteria as guidelines (https://en.wikipedia.org/wiki/SMART_criteria):

- **S**pecific – target a specific area for improvement.
- **M**easurable – quantify or at least suggest an indicator of progress.
- **A**chievable – specify what will be accomplished
- **R**ealistic – state what results can realistically be achieved, given available resources.
- **T**ime-related – specify when the result(s) can be achieved.

The following format is an example:

<Company-Project-id-version function-capability-“requirement number”>

Requirement text is located here (indicated by bold font).

Explanatory text related to the requirement should also be provided (indicated by non-bold font)

Examples:

<GSU-GS_SP2016-1 User-Capability-000100>

The project must allow new users to be added, updated, or deleted by the application.

Implementation: Mandatory (Note: could be “optional” in other requirements)

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

This is section to describe the capabilities/requirements you will be providing for your user or customer to administrate and maintain the usage of your project. (e.g., user data backup, fault recovery, routine maintenance ……)

3.3 Security and Fraud Prevention

In this segment, we investigate the security of proposed access control mechanisms. First, the GPSW and MA-ABE plans are ended up being secure in individually. Particularly, the scrambled information is secret to non-authorized users. Likewise, they are both impervious to client arrangement, and MA-ABE is further resistant to agreement among up to N − 2 AAs in one PUD. This infers that strong security certification is accomplished through document encryption.

Second, for the write access authorization, the restricted property of the hash chain guarantees that a writer can just get compose keys for the time frame that she is approved for.

Also, writer can scarcely fashion a mark of the hash affix end as per the non-forge ability of the hidden mark plan. Third, the renouncement scheme is secure, which is demonstrated in under standard security presumptions. Finally for the break-glass get to, an enemy is not possible to acquire TPD given ˜Eandgs2.

Performance Analysis

The execution investigation is abridged in Table. 3. We think about our solution with that of which utilizations CP-ABE, and a solitary open power is utilized. Miss the number of PUDs, while Ni is the quantity of PAAs in the PUD.

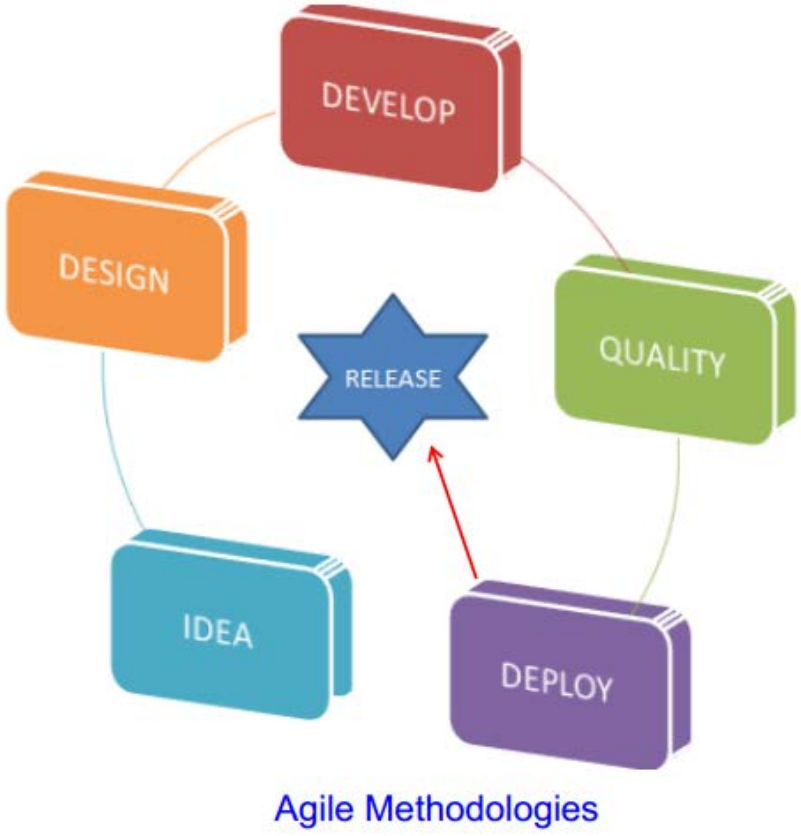
Note that the key administration many-sided quality is as far as the quantity of communications during key conveyance. For cipher text length correlation, for our plan the access policy for every PUD is confined to conjunctive structure: Ppub: =P1 ∧... ∧Pm,where every Pi is a Boolean statement comprising of "∧" and "∨". The number of cipher text parts identified with the PUDs is |AC| =Σmi=1(ΣNik=1|ACi;k|),which is straight to the quantity of PUDs and the quantity of PAAs. In practice there are typically a couple of PUDs (e.g., <5) and a couple PAAs and sorts of attributes in each of them (e.g., 5).

Thusly the extra stockpiling overhead for the server created by each cipher text (encryption of the document encryption key) is normally in the request of many gathering components, which ordinarily equivalents to a couple of hundred bytes if 160-piece ECC is embraced. This is satisfactory contrasted and the length of a PHR archive (as a rule in the request of KB).

Aside from those, for each owner, to change access approaches and empower crisis access, 2 extra group elements (s and d) should be privately put away for each encoded PHR document, which is entirely little. The cipher text length correlation depends on the same access policy as in our plan; the one in is an upper bound subsequent to there is no need to use special case.

At last, the computational overhead in our plan is low, subsequent to the decryption operation can be for the most part designated to the server. A client can present the server and just figures one bilinear matching: e(Du,E1). This is secure because the server does not know Du.

3.4 Release and Transition Plan



This project works on agile methodology and developed in continuous collaboration with all project mates. At each point integrate and test policy is applied which resulted in the successful completion of this project.

4 Project Design Description

Problem definition

We consider a PHR framework where there exist various PHR proprietors and numerous PHR clients. The proprietors allude to patients who have full control over their own PHR information, i.e., they can make, oversee and erase it. The clients incorporate readers and

journalists that may originate from different viewpoints. For instance, a companion, a caregiver or a scientist.

There is additionally a focal server having a place with the PHR service provider that stores every one of the proprietors' PHRs, where there might be a substantial number of proprietors. Clients get to the PHR archives through the server keeping in mind the end goal to read or keep in touch with somebody's PHR. The PHR documents can be composed by their classes progressively.

We consider legit yet inquisitive cloud server as those. That implies the server will attempt to discover however much mystery data in the put away PHR documents as could reasonably be expected, yet they will sincerely take after the convention all in all.

The server may likewise intrigue with a couple of malevolent clients in the framework. Then again, a few clients will likewise attempt to get to the records past their benefits. For instance, a drug store might need to get the solutions of patients for showcasing and boosting its benefits. To do as such, they may even intrigue with different clients.

Likewise, we accept every gathering in our system is preloaded with an open/private key pair, and element authentication can be finished by test reaction conventions.

In we imagine that every patient determines her own security arrangement. The proprietors need to keep the server and unauthorized users from taking in the substance of their PHR documents. Specifically, we have the accompanying destinations:

Fine-grained access control ought to be upheld, which means diverse clients can be approved to peruse distinctive arrangements of documents. Likewise, we should empower various authors to pick up compose access to contribute data to PHR with responsibility.

User denial. At whatever point it is fundamental, a client's entrance benefits ought to be revoked from future access in an effective way.

The information access arrangements ought to be adaptable, i.e., changes to the predefined approaches should be permitted, particularly under crisis situations.

Efficiency. To bolster an extensive and flighty number of clients, the system should be very versatile, as far as many-sided quality in key administration, user management, and calculation and capacity.

The proposed system

Since the cloud server is no more thought to be completely trusted, information encryption should be received which ought to implement persistent determined security arrangements. To this end, every proprietor should go about as a power that freely generates and disseminates cryptographic keys to approved clients. Be that as it may, as mentioned before, the administration complexities may increment straightly with the number of users and proprietors.

Our proposed system can tackle this issue well. The key thought is twofold. First, keeping in mind the end goal to bring down the many-sided quality of encryption and client management for every proprietor, we receive trait based encryption (ABE) as the encryption primitive.

Clients/information are grouped by traits, for example, professional roles/information sorts. Proprietors encode their PHR information under a specific access policy (or, a chose set of properties), and just clients that have legitimate sets of qualities (decoding keys) are permitted to pick up read access to those data. Second, we separate the clients in the entire PHR framework into multiple security domains (SDs), and for every SD we present one or more powers which govern attribute-based certifications for clients inside that SD.

There are two categories of SDs: open domains (PUDs) and individual domains (PSDs). Every proprietor is in charge of her PSD comprising of clients by and by associated with her. A PUD ordinarily contains countless clients, and various open attribute authorities (PAA) that distributive represents a disjoint subset of traits to remove key escrow. A proprietor encodes her PHR information so that approved users from both her PSD and PUDs may read it.

As a general rule, every PUD can be mapped to a free segment in the general public, for example, the human services, training, and government tor protection part. Clients having a place with a PUD just need to obtain credentials from the comparing open powers, without the need to interact with any PHR proprietor, which enormously decreases the key administration overhead of proprietors and users. The system is shown which includes numerous SDs, multiple owners (individual AAs), different PAAs, and various clients (authors and readers).Next, we depict the structure thoughtfully.

Key dispersion.

Clients first acquire trait based keys from their AAs. They present their personality data and acquire mystery keys that dilemma them to guaranteed traits. For instance, a doctor in it would get "hospital A, doctor, M.D., inside prescription" as her qualities, perhaps from various AAs.

Furthermore, the AAs disperse compose keys that grant clients in their SD to keep in touch with a few patients' PHR. A client needs to present the compose keys with a specific end goal to pick up compose access to the cloud server PHR Access. In the first place, the proprietors transfer ABE-scrambled PHR documents to the cloud server each of them is connected with some customized access policy enforced by encryption.

User repudiation.

There are two sorts of client repudiation. The first is revocation of a client's characteristic, which is finished by the AA that the client has a place with,

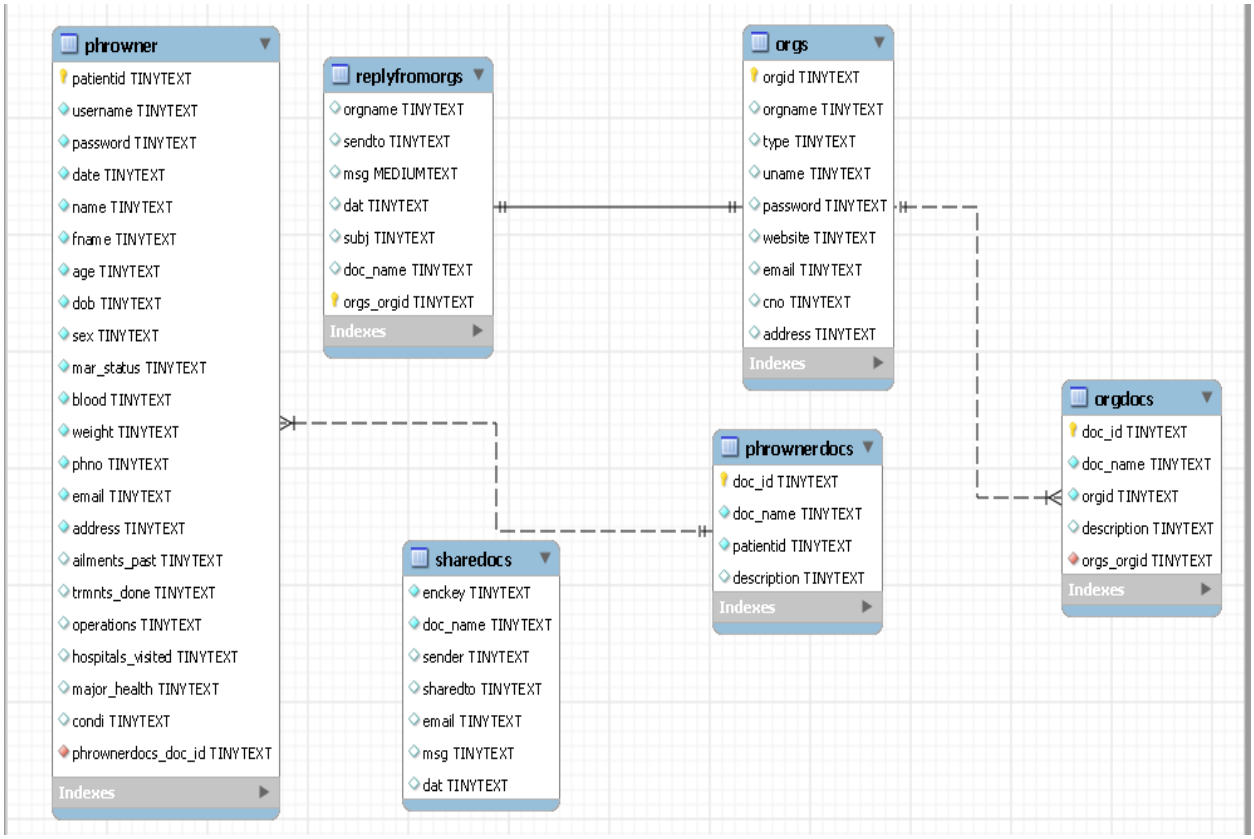
Where the genuine calculations can be assigned to the cloud server to enhance productivity. The second one is redesign of a proprietor's entrance strategy for a specific PHR record, in light of data went from the proprietor to the server.

At the point when a crisis happens, the consistent access arrangements may no longer be relevant. To handle this circumstance, break-glass access is needed to access the casualty's PHR. In our system, every proprietor's PHR's entrance right is likewise assigned to a crisis office.

To keep from abuse of break-glass choice, the crisis staff needs to contact the ED to confirm her identity and the crisis circumstance, and acquire impermanent read keys. After the crisis is over.

5. Internal/ external interface impacts and specification

Date Model



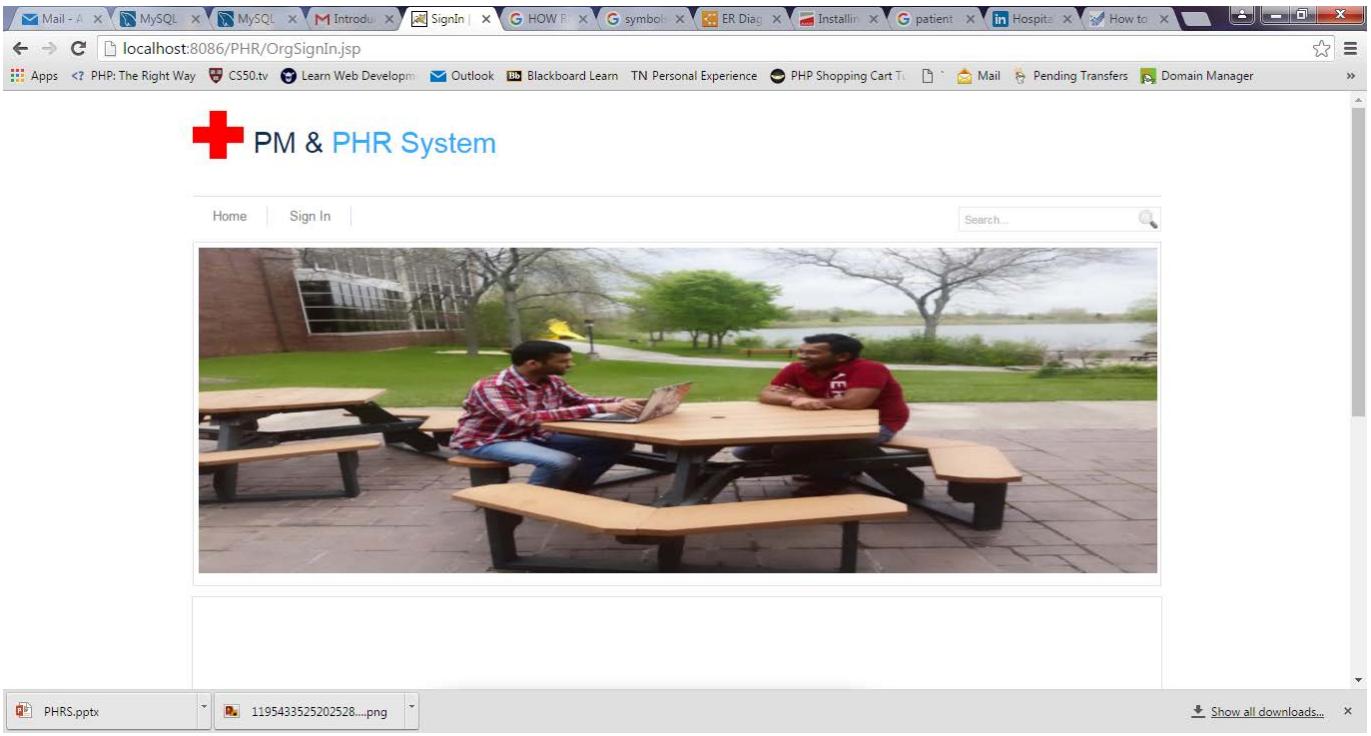
6. Design Units Impacts

6.1 Functional Area/Design

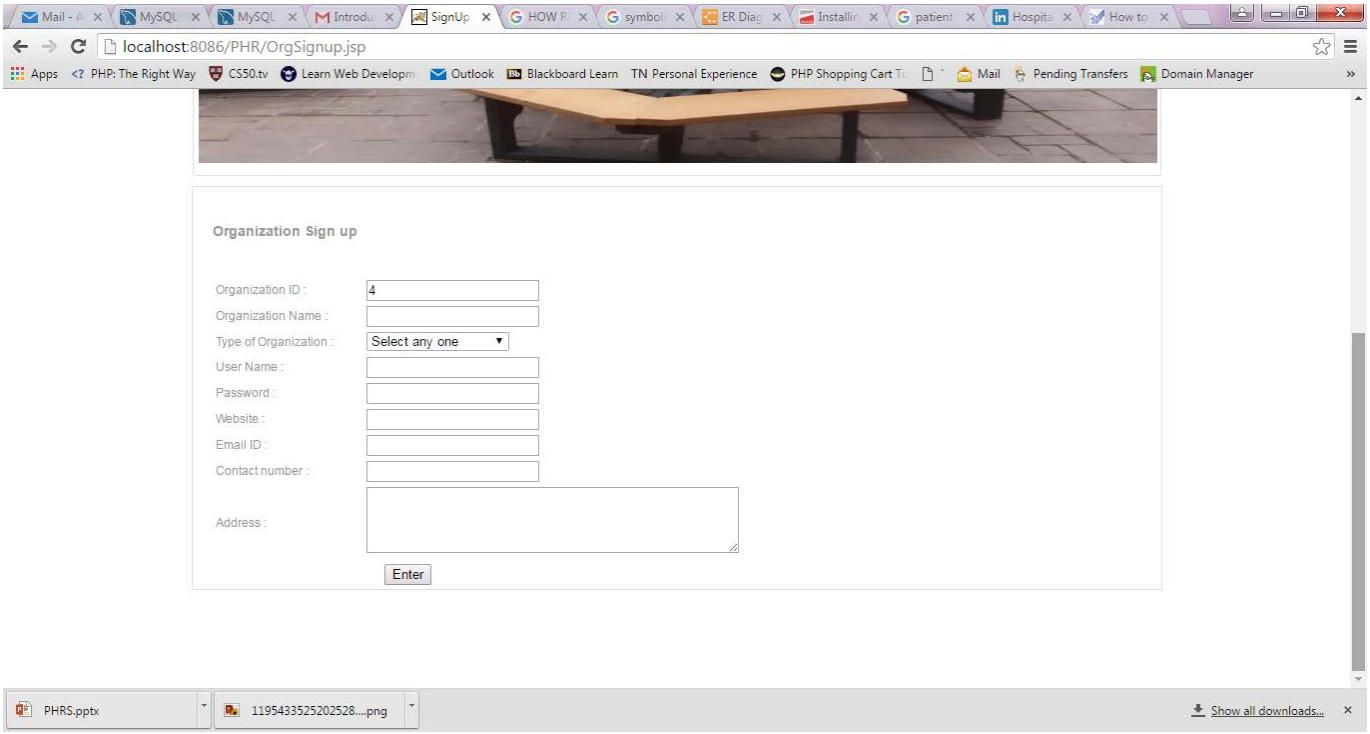
6.1.1 Functional Overview

Screen Shots:

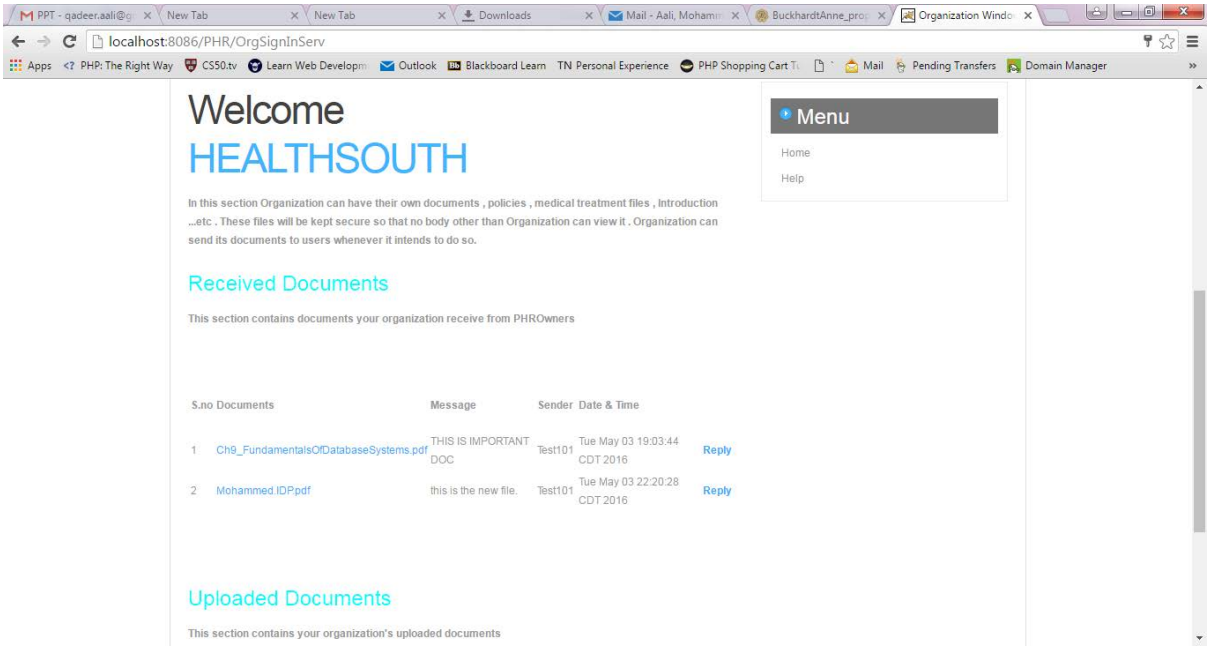
HOMEPAGE



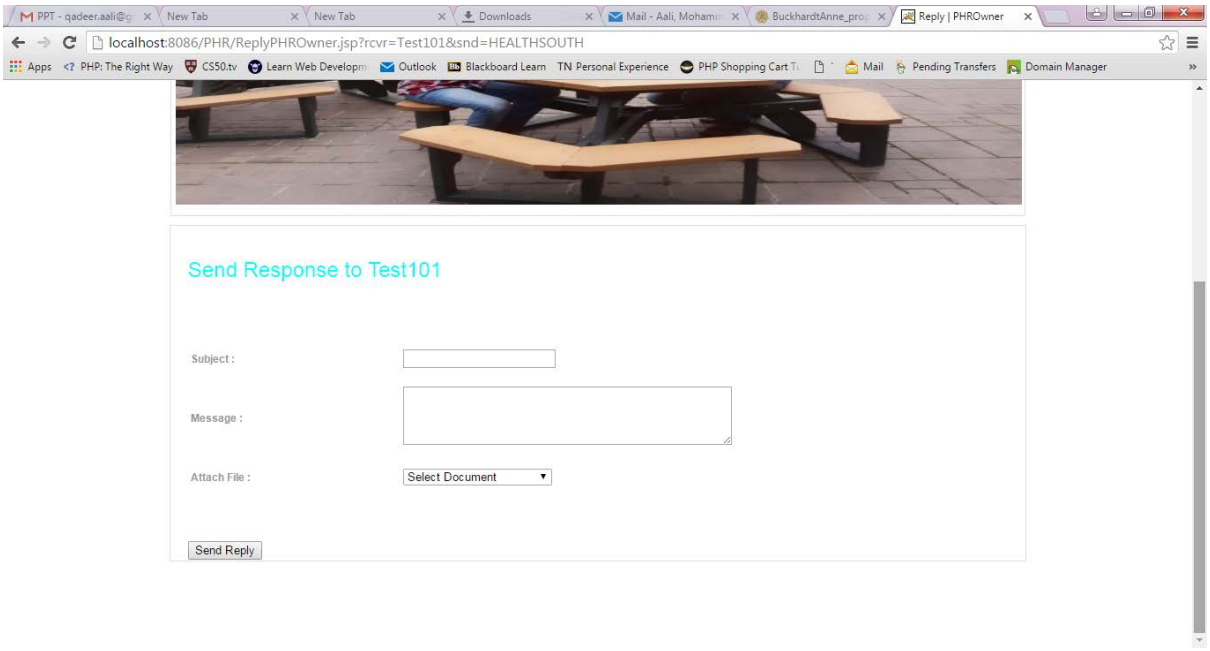
SIGNUP PAGE



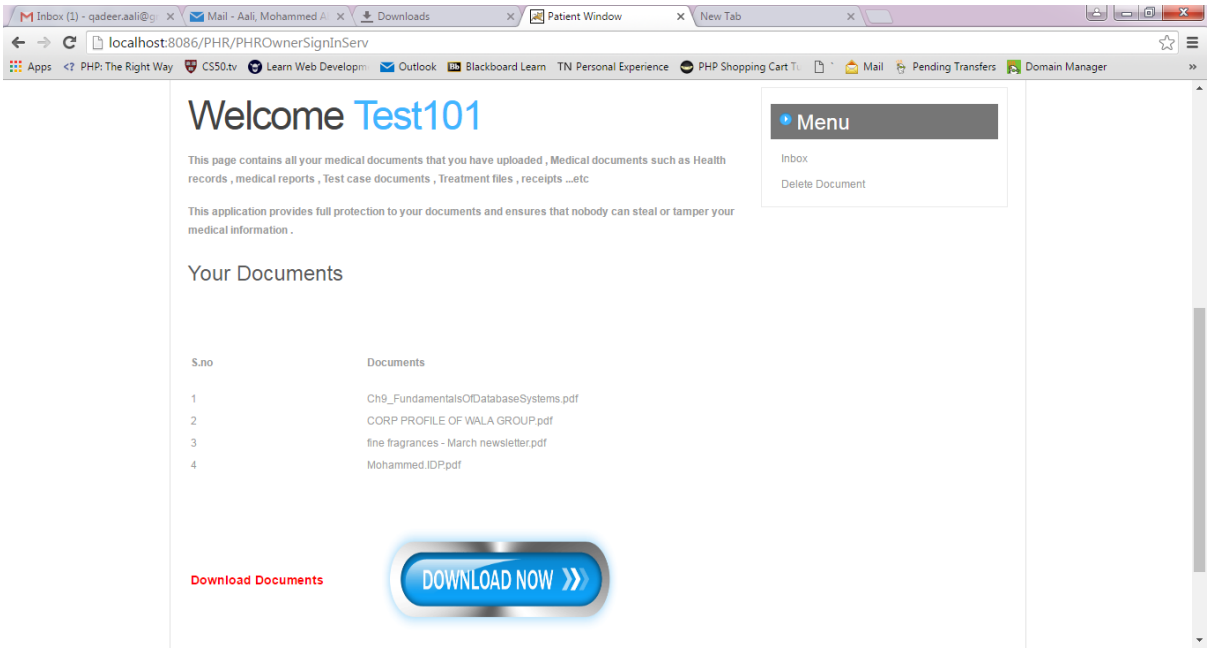
ORGANIZATION WELCOME PAGE



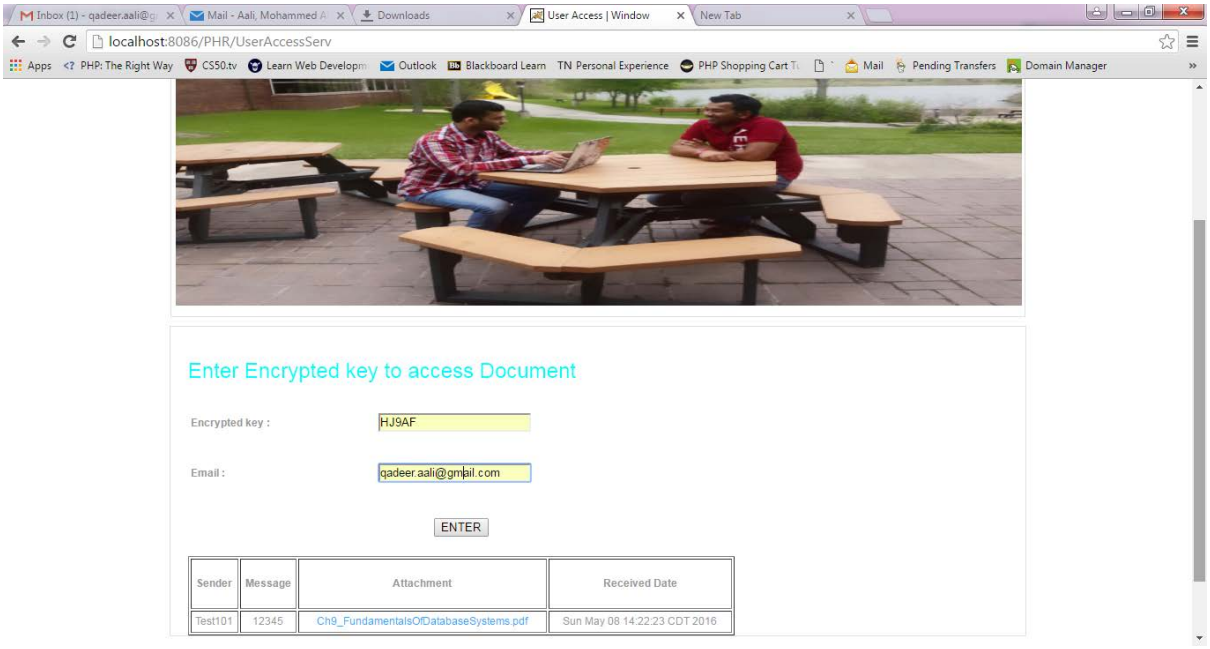
PHR OWNER REPLY PAGE



PHR DOCUMENT VIEW PAGE:



ENCRYPTED FILE ACCESS PAGE:



6.1.2 Requirements

Execution is measured regarding the yield gave by the application. Necessity particular has imperative impact in the examination of a framework. Just when the necessity particulars are legitimately given, it is conceivable to outline a framework, which will fit into required environment. It rests to a great extent with the clients of the current framework to give the necessity determinations since they are the general population who at last utilize the framework. This is on account of the prerequisites must be known amid the underlying stages so that the framework can be planned by necessities. It is extremely hard to change the framework once it has been composed and then again planning a framework, which does not take into account the necessities of the client, is of no utilization.

The prerequisite detail for any framework can be comprehensively expressed as given beneath:

The framework ought to have the capacity to interface with the current framework

- The framework ought to be exact
- The framework ought to be superior to the current framework

The current framework is totally reliant on the client to perform every one of the obligations.

SOFTWARE REQUIREMENTS:

- Operating System: Windows 7, Windows 8

- Application Server: Tomcat 8.0
- Server side Technology: Servlets
- Client Side Technology: Swing, CSS, HTML, JavaScript
- Database: MySQL

HARDWARE REQUIREMENTS:

- Intel Core I3
- 2 GB RAM
- 80 GB Hard Disk Space

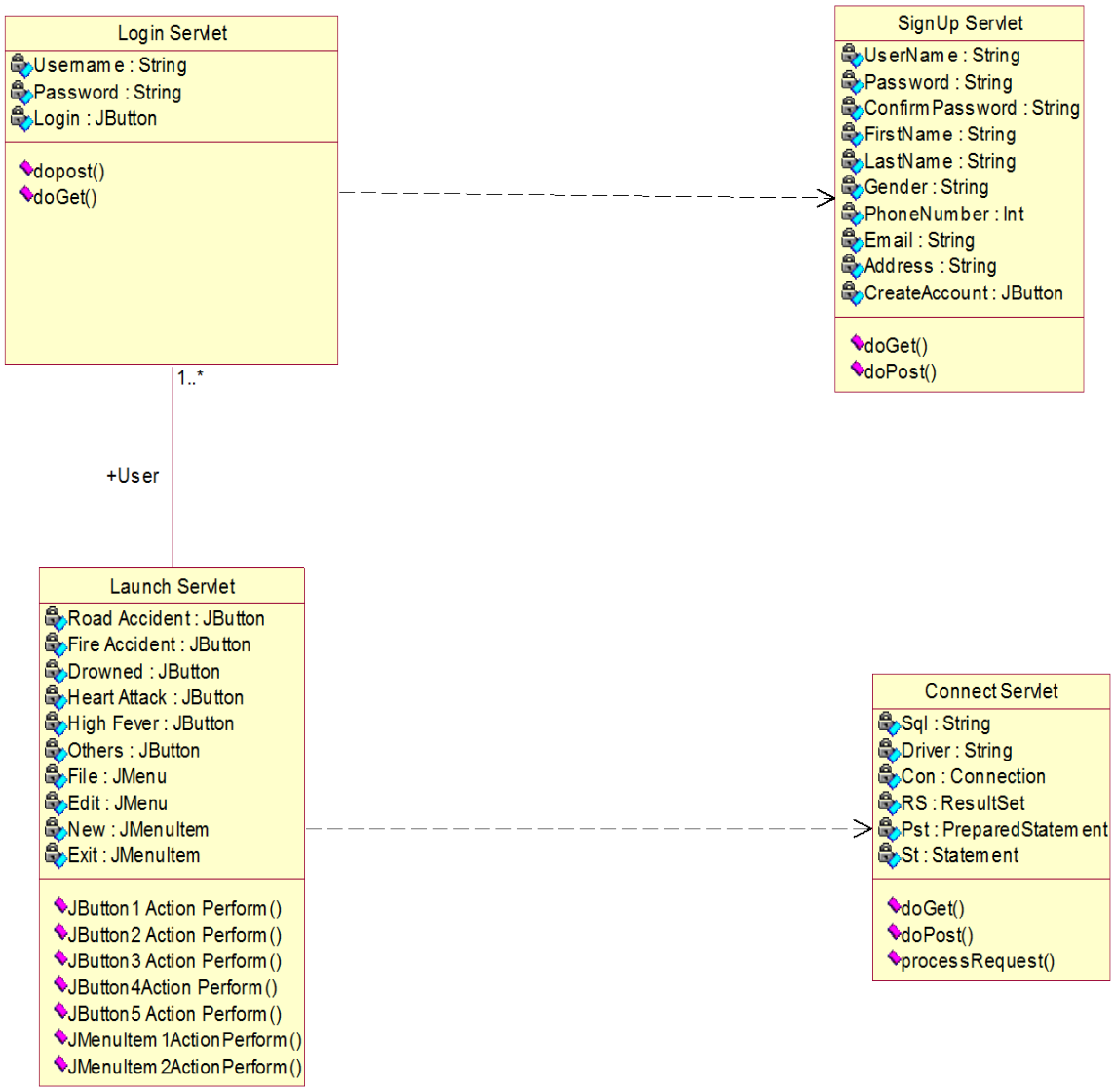


FIG: CLASS DIAGRAM

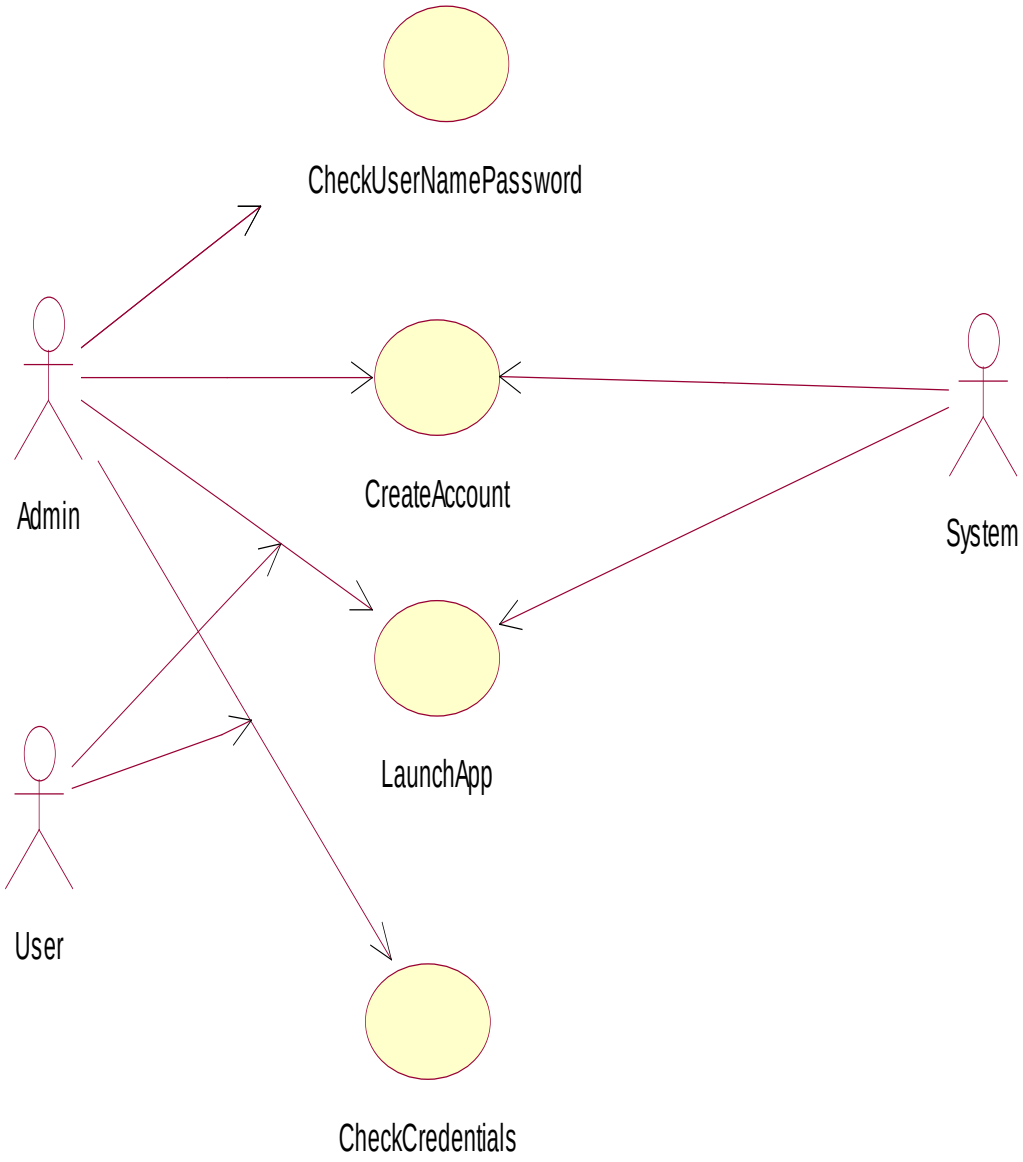


FIG: USE CASE DIAGRAM

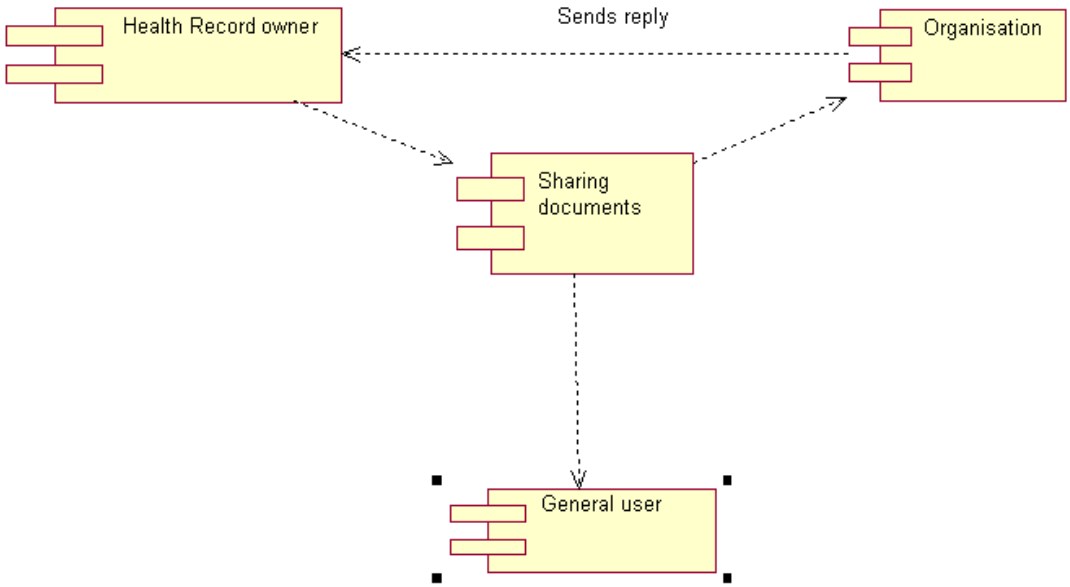


FIG: COMPONENT DIAGRAM

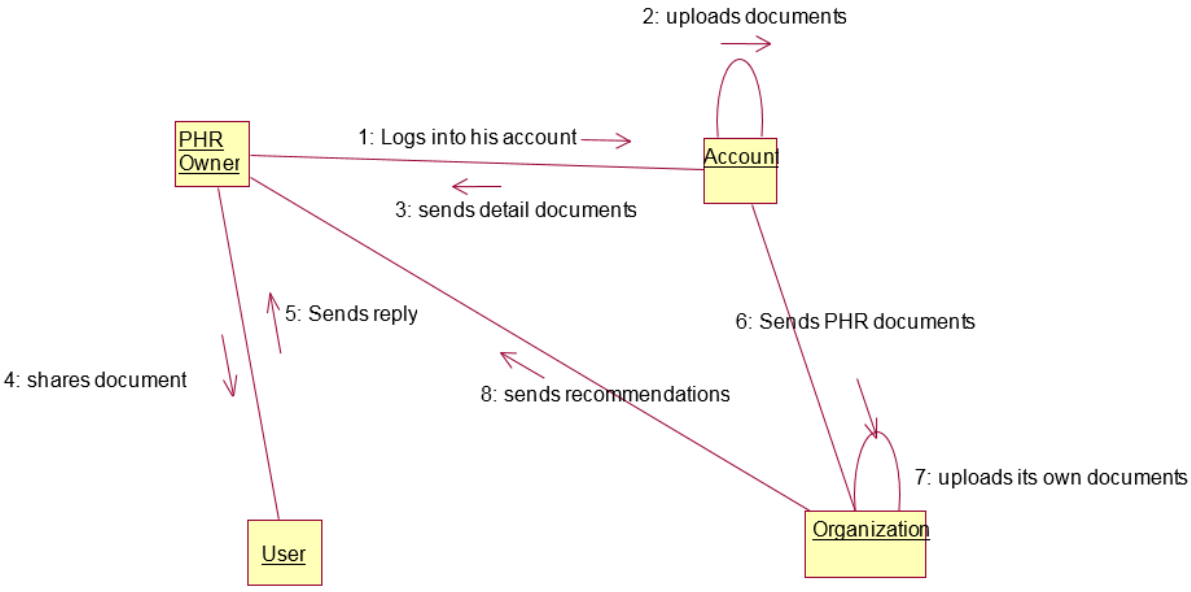


FIG: COLLABORATION DIAGRAM

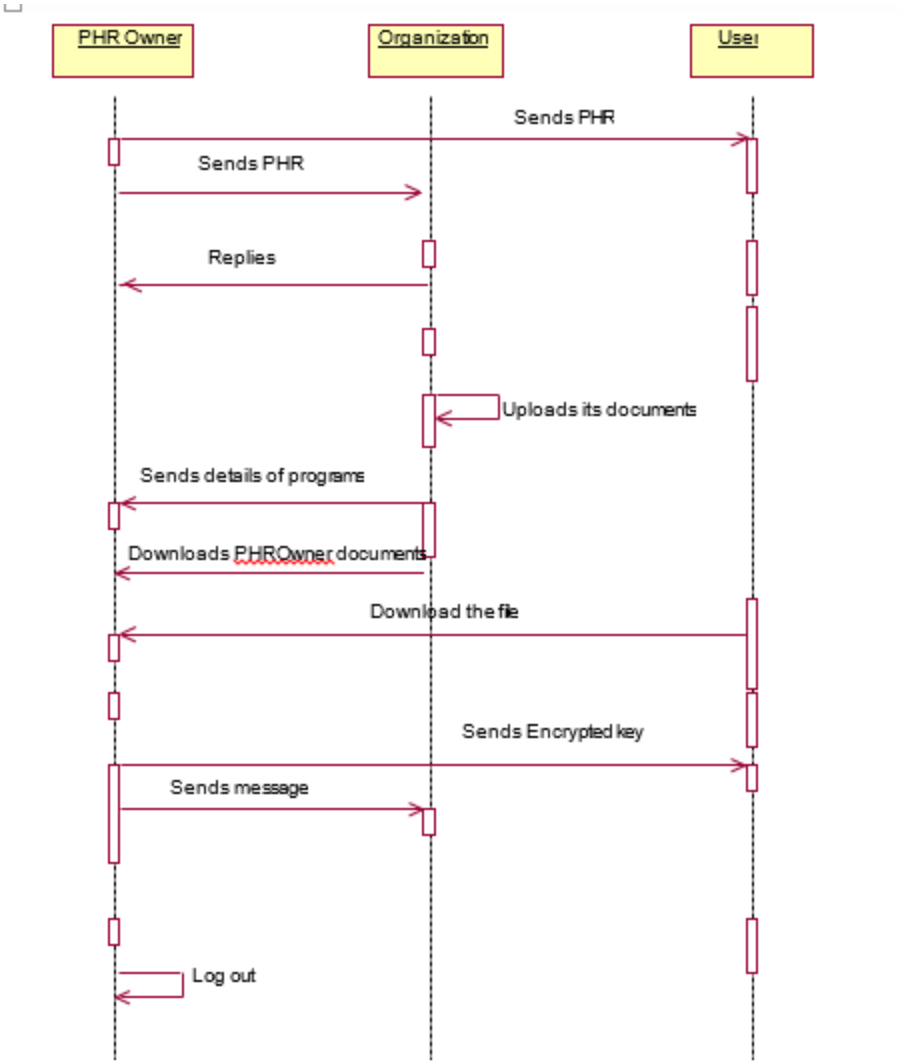


FIG: SEQUENCE DIAGRAM

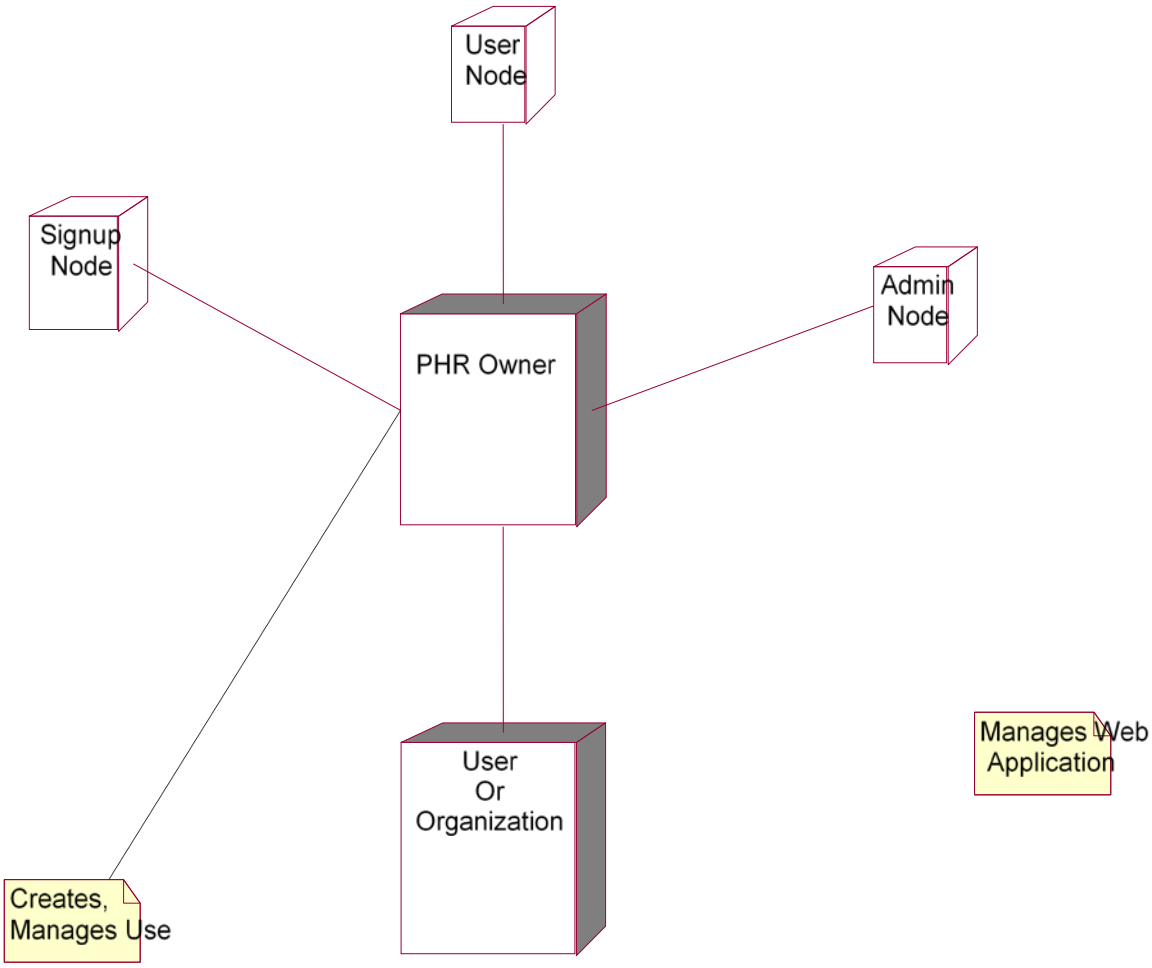


FIG: DEPLOYEMENT DIAGRAM

7 Acknowledgement

We thank our professor cum mentor Dr. Soon-Ok Park, who always supported us in each step with her wide knowledge base. Dr. Park's personal attention to each student has helped us achieve perfection from the beginning of project to submission of document. She was available to us all the time which is really appreciated.

8 References

1. Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

2. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, 2007.

3. M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, 2008.

4. C. Wang, K. Ren, W. Lou, and J. Li, "Towards Publicly Auditable Secure Cloud Data Storage Services," IEEE Network Magazine, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

5. M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS '07), pp. 1-6, 2007.

6. R. Curtmola, O. Khan, and R. Burns, "Robust Remote Data Checking," Proc. Fourth ACM Int'l Workshop Storage Security and Survivability (StorageSS '08), pp. 63-68, 2008.

7. A.L. Ferrara, M. Green, S. Hohenberger, and M. Pedersen, "Practical Short Signature Batch Verification," Proc. Cryptographers' Track at the RSA Conf. 2009 on Topics in Cryptology (CT-RSA), pp. 309-324, 2009.

8. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08), pp. 1-10, 2008.

9. G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage from Homomorphic Identification Protocols," Proc. 15th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT), pp. 319-333, 2009.

10. F. Sebe, J. Domingo-Ferrer, A. Martı´nez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient Remote Data Possession Checking in Critical Information Infrastructures," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 8, pp. 1034-1038, Aug. 2008.