

CS 475 - Senior Project

*Secure Electronic Medical Record System
(SEMRS)*

Ryan Kane & Ricky Orndorff

Software Design Document

Table of Contents

1. Introduction	3
1.1. Document Objectives	3
1.2. Intended Audiences	3
1.3. References	3
1.4. Database Overview.....	4
5.1. References	4
2. Database-wide Design Decisions.....	5
2.1. Interfaces.....	5
2.1.1. Log-in Page	5
2.1.2. Home Page.....	5
2.2. Appearance / Naming	5
2.3. DBMS Platform	5
2.4. Qualities.....	6
2.4.1. Availability.....	6
2.4.2. Security.....	6
2.4.3. Privacy.....	6
2.5. Distribution.....	6
2.6. Operations	6
2.7. Maintenance	6
3. Detailed Database Design	7
3.1. Schema	7
3.2. System Context Diagram	13
3.3. Use Case.....	14
4. Technical Design	15
4.1. System Requirements.....	15
4.1.1. Programming Tools.....	15
4.1.2. Running Environment	15
4.1.3. Database Server	15
4.1.4. Server.....	15
4.1.5. Client	15

1. Introduction

The section introduces the Design Document (DD) for the Secure Electronic Medical Records System (SEMRS) to its readers.

1.1. Document Objectives

This DD for the SEMRS software has the following objectives:

- (a) Describe the design of a database, that is, a collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via a database management system (DBMS). It can also describe the software units used to access or manipulate the data.
- (b) To serve as the basis for implementing the database and related software units. It provides the acquirer visibility into the design and provides information needed for software support.
- (c) All sections should remain in this document. If a section is to be tailored out, the section shall remain and contain the words "Tailored out".

1.2. Intended Audiences

This DD is intended for the following audiences:

- (a) Technical reviewers, who must evaluate the quality of this document.
- (b) SEMRS developers including:
 - (i) Architects, whose overall architecture must meet the requirements specified in this document.
 - (ii) Designers, whose design must meet the requirements specified in this document.
 - (iii) Programmers, whose software must implement the requirements specified in this document.
 - (iv) Testers, whose test cases must validate the requirements specified in this document.

1.3. References

This DD refers to the following references:

- (a) SEMRS Software Requirements Document
- (b) SEMRS Software Specification Document

1.4. Database Overview

This database fulfills the following purposes:

- (a) The general nature of this database is to electronically store and encrypt patient records securely. This will be accomplished by creating a mobile patient record database look up to be used as a secure web application that enables authorized users i.e. doctors and nurses to view patient information. The software will have an web based front-end and a MySQL back-end database.
- (b) The business context of this database can be shown in section 3.3 of this design document.
- (c) This development of this database began in February of 2012, and is expected to be finished development in May 2012.
- (d) This database is intended to be used by medical personnel to store patient's medical files securely.
- (e) Maintenance of this database will be implemented through a "database maintenance plan". The details of this maintenance plan are included in section 2.8 of this software design document will regularly reorganize data and index pages, update statistics, as well as remove unused space from the database files by doing physical database fragmentation. This maintenance will need to be done because as data and index pages fill, updating requires more time, therefore reorganizing data and index pages will improve performance (Robidoux).

5.1. References

Robidoux, Greg. "Performing MaINTenance Tasks in SQL Server." *SQL Server Tips, Techniques and Articles*. Mssqltips.com, 07 July 2006. Web. 05 Mar. 2012. <<http://www.mssqltips.com/sqlservertip/1013/performing-maINTenance-tasks-in-sql-server/>>.

2. Database-wide Design Decisions

This section documents decisions about how the database will behave, from a user's point of view, in meeting its requirements, as well as other decisions affecting further design of the database.

2.1. Interfaces

2.1.1. Log-in Page

All users will have to log on to the system through the same log-in page. The user will be required to enter their user name and password. After they have successfully authenticated, they will be redirected to the appropriate interface that corresponds to their user type. This type of strict access control will ensure that the correct personnel have access to only what they are permissible to view.

2.1.2. Home Page

The home page will allow all users to have the option to search for patients.

2.1.2.1. *Administration*

Administrators will have the authority to enable/disable accounts and add/modify users. They will not be allowed to search for patients or view patient data.

2.1.2.2. *Physician*

Physicians will be able to add/update patient records and send messages to other physicians concerning patients.

2.1.2.3. *Pharmacy*

Pharmacists will be able to view patient information and fill prescriptions.

2.1.2.4. *Office*

Receptionists will be able to add/modify patient information and add visits.

2.2. Appearance / Naming

Patient's records will be named by the last name of the patient, thus making it simple to see which patient records belong to what patient, just by looking at the file name.

2.3. DBMS Platform

The database management system platform to be used is MySQL version 5.5.21, released in February of 2012.

2.4. Qualities

Design decisions on the levels and types of availability, security, and privacy of operations to be offered by the database.

2.4.1. Availability

The database is expected to be available 99.99% of the time. The database is expected to be available at all times, except for when maintenance needs to be done, or system updates need to be conducted.

2.4.2. Security

Strict security measures will be put in place in the database to ensure that patients records are secure so that no unauthorized viewers have access to view or modify patient information. This database will use MySQL's advanced encryption standard algorithm to encrypt and decrypt the sensitive information contained in the database.

2.4.3. Privacy

The strict access control mentioned in Section 2.1.1 of this design document, coupled with the security measures in section 3.6.4. of the software specifications document provide privacy of patients sensitive information.

2.5. Distribution

In terms of distribution, any sensitive information will be encrypted client-side before being pushed to the server. Maintenance, as well as master database file updates will be conducted by the database administrator.

2.6. Operations

Backup operations for the database will take place nightly, ensuring that all information entered into the database is backed up securely. Permissions for the backup will only be granted to the database administrator. This database will not require any non-standard technologies such as video and sound.

2.7. Maintenance

Maintenance will be conducted by implementing a "maintenance plan". This plan includes regularly reorganizing data and index pages, updating statistics, as well as removing unused space from the database files by conducting physical database fragmentation. This maintenance plan will need to be carried out weekly because as data and index pages fill, updating requires more time, therefore reorganizing data and index pages will keep the database running at peak performance. (Robidoux)

3. Detailed Database Design

3.1. Schema

The following schema is subject to change over the course of the development of the SEMRS software application.

```

/* Stores facility information */
CREATE TABLE `facility` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `phone` VARCHAR(255) NOT NULL,
  `fax` VARCHAR(255) NOT NULL,
  `street` VARCHAR(255) NOT NULL,
  `postal_code` VARCHAR(255) NOT NULL,
  `city` VARCHAR(255) NOT NULL,
  `state` VARCHAR(255) NOT NULL,
  `country` VARCHAR(255) NOT NULL,
  `notes` TEXT NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Country reference */
CREATE TABLE `geo_country_reference` (
  `countries_id` INT(5) NOT NULL AUTO_INCREMENT,
  `countries_name` VARCHAR(64) DEFAULT NULL,
  `countries_iso_code_2` char(2) NOT NULL DEFAULT '',
  `countries_iso_code_3` char(3) NOT NULL DEFAULT '',
  PRIMARY KEY (`countries_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Patient history data */
CREATE TABLE `history_data` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `coffee` LONGTEXT,
  `tobacco` LONGTEXT,
  `alcohol` LONGTEXT,
  `sleep_patterns` LONGTEXT,
  `exercise_patterns` LONGTEXT,
  `seatbelt_use` LONGTEXT,
  `counseling` LONGTEXT,
  `hazardous_activities` LONGTEXT,
  `recreational_drugs` LONGTEXT,
  `last_breast_exam` VARCHAR(255) DEFAULT NULL,
  `last_mammogram` VARCHAR(255) DEFAULT NULL,
  `last_gynecological_exam` VARCHAR(255) DEFAULT NULL,
  `last_rectal_exam` VARCHAR(255) DEFAULT NULL,
  `last_prostate_exam` VARCHAR(255) DEFAULT NULL,
  `last_physical_exam` VARCHAR(255) DEFAULT NULL,

```

```

`last_sigmoidoscopy_colonoscopy` VARCHAR(255) DEFAULT NULL,
`last_ecg` VARCHAR(255) DEFAULT NULL,
`last_cardiac_echo` VARCHAR(255) DEFAULT NULL,
`last_retinal` VARCHAR(255) DEFAULT NULL,
`last_fluvax` VARCHAR(255) DEFAULT NULL,
`last_pneuvax` VARCHAR(255) DEFAULT NULL,
`last_ldl` VARCHAR(255) DEFAULT NULL,
`last_hemoglobin` VARCHAR(255) DEFAULT NULL,
`last_psa` VARCHAR(255) DEFAULT NULL,
`last_exam_results` VARCHAR(255) DEFAULT NULL,
`history_mother` LONGTEXT,
`history_father` LONGTEXT,
`history_siblings` LONGTEXT,
`history_offspring` LONGTEXT,
`history_spouse` LONGTEXT,
`relatives_cancer` LONGTEXT,
`relatives_tuberculosis` LONGTEXT,
`relatives_diabetes` LONGTEXT,
`relatives_high_blood_pressure` LONGTEXT,
`relatives_heart_problems` LONGTEXT,
`relatives_stroke` LONGTEXT,
`relatives_epilepsy` LONGTEXT,
`relatives_mental_illness` LONGTEXT,
`relatives_suicide` LONGTEXT,
`cataract_surgery` DATETIME DEFAULT NULL,
`tonsillectomy` DATETIME DEFAULT NULL,
`cholecystectomy` DATETIME DEFAULT NULL,
`heart_surgery` DATETIME DEFAULT NULL,
`hysterectomy` DATETIME DEFAULT NULL,
`hernia_repair` DATETIME DEFAULT NULL,
`hip_replacement` DATETIME DEFAULT NULL,
`knee_replacement` DATETIME DEFAULT NULL,
`appendectomy` DATETIME DEFAULT NULL,
`DATE` DATETIME DEFAULT NULL,
`patient_id` BIGINT(20) NOT NULL DEFAULT '0',
`additional_history` TEXT,
`exams` TEXT NOT NULL,
PRIMARY KEY (`id`),
KEY `patient_id` (`patient_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Immunization records */
CREATE TABLE `immunizations` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `patient_id` BIGINT(20) DEFAULT NULL,
  `administered_DATE` DATE DEFAULT NULL,
  `immunization_id` INT(11) DEFAULT NULL,
  `cvx_code` INT(11) DEFAULT NULL,
  `manufacturer` VARCHAR(100) DEFAULT NULL,
  `lot_number` VARCHAR(50) DEFAULT NULL,
  `administered_by` BIGINT(20) DEFAULT NULL,
  `education_DATE` DATE DEFAULT NULL,
  `vis_DATE` DATE DEFAULT NULL COMMENT 'DATE of VIS Statement',

```



```

`note` TEXT,
`create_DATE` DATETIME DEFAULT NULL,
`upDATE_DATE` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
`created_by` BIGINT(20) DEFAULT NULL,
`upDATED_by` BIGINT(20) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `patient_id` (`patient_id`),
KEY `administered_by` (`administered_by`),
KEY `created_by` (`created_by`),
KEY `upDATED_by` (`upDATED_by`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Insurance Company Information */
CREATE TABLE `insurance_companies` (
`id` INT(11) NOT NULL DEFAULT '0',
`name` VARCHAR(255) DEFAULT NULL,
`attn` VARCHAR(255) DEFAULT NULL,
`cms_id` VARCHAR(15) DEFAULT NULL,
`freeb_type` tinyINT(2) DEFAULT NULL,
`x12_receiver_id` VARCHAR(25) DEFAULT NULL,
`x12_default_partner_id` INT(11) DEFAULT NULL,
`alt_cms_id` VARCHAR(15) NOT NULL DEFAULT '',
PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Patient insurance data */
CREATE TABLE `insurance_data` (
`id` BIGINT(20) NOT NULL AUTO_INCREMENT,
`type` ENUM('primary','secondary','tertiary') DEFAULT NULL,
`provider` VARCHAR(255) DEFAULT NULL,
`plan_name` VARCHAR(255) DEFAULT NULL,
`policy_number` VARCHAR(255) DEFAULT NULL,
`group_number` VARCHAR(255) DEFAULT NULL,
`subscriber_lname` VARCHAR(255) DEFAULT NULL,
`subscriber_mname` VARCHAR(255) DEFAULT NULL,
`subscriber_fname` VARCHAR(255) DEFAULT NULL,
`subscriber_relationship` VARCHAR(255) DEFAULT NULL,
`subscriber_ss` VARCHAR(255) DEFAULT NULL,
`subscriber_DOB` DATE DEFAULT NULL,
`subscriber_street` VARCHAR(255) DEFAULT NULL,
`subscriber_postal_code` VARCHAR(255) DEFAULT NULL,
`subscriber_city` VARCHAR(255) DEFAULT NULL,
`subscriber_state` VARCHAR(255) DEFAULT NULL,
`subscriber_country` VARCHAR(255) DEFAULT NULL,
`subscriber_phone` VARCHAR(255) DEFAULT NULL,
`subscriber_employer` VARCHAR(255) DEFAULT NULL,
`subscriber_employer_street` VARCHAR(255) DEFAULT NULL,
`subscriber_employer_postal_code` VARCHAR(255) DEFAULT NULL,
`subscriber_employer_state` VARCHAR(255) DEFAULT NULL,
`subscriber_employer_country` VARCHAR(255) DEFAULT NULL,
`subscriber_employer_city` VARCHAR(255) DEFAULT NULL,
`copay` VARCHAR(255) DEFAULT NULL,

```

```

`DATE` DATE NOT NULL DEFAULT '0000-00-00',
`patient_id` BIGINT(20) NOT NULL,
`subscriber_sex` VARCHAR(25) DEFAULT NULL,
`accept_assignment` VARCHAR(5) NOT NULL DEFAULT 'TRUE',
PRIMARY KEY (`id`),
UNIQUE KEY `pid_type_DATE` (`patient_id`,`type`,`DATE`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Insurance provider numbers */
CREATE TABLE `insurance_numbers` (
  `id` INT(11) NOT NULL DEFAULT '0',
  `provider_id` INT(11) NOT NULL DEFAULT '0',
  `insurance_company_id` INT(11) DEFAULT NULL,
  `provider_number` VARCHAR(20) DEFAULT NULL,
  `rendering_provider_number` VARCHAR(20) DEFAULT NULL,
  `group_number` VARCHAR(20) DEFAULT NULL,
  `provider_number_type` VARCHAR(4) DEFAULT NULL,
  `rendering_provider_number_type` VARCHAR(4) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* User level for authorization */
CREATE TABLE `level` (
  `id` INT(1) NOT NULL,
  `type` VARCHAR(20) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Patient information */
CREATE TABLE `patient_data` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `title` VARCHAR(255) NOT NULL,
  `language` VARCHAR(255) NOT NULL,
  `fname` VARCHAR(255) NOT NULL,
  `mname` VARCHAR(255) NOT NULL,
  `lname` VARCHAR(255) NOT NULL,
  `DOB` DATE NOT NULL,
  `street` VARCHAR(255) NOT NULL,
  `postal_code` VARCHAR(255) NOT NULL,
  `city` VARCHAR(255) NOT NULL,
  `state` VARCHAR(255) NOT NULL,
  `country` VARCHAR(255) NOT NULL,
  `drivers_license` VARCHAR(255) NOT NULL,
  `national_id` VARCHAR(255) NOT NULL,
  `occupation` VARCHAR(255) NOT NULL,
  `phone_home` VARCHAR(255) NOT NULL,
  `phone_cell` VARCHAR(255) NOT NULL,
  `pharmacy_id` BIGINT(20) NOT NULL,
  `status` tinyINT(1) NOT NULL DEFAULT '1',
  `sex` ENUM('m','f') NOT NULL,
  `ethnicity` VARCHAR(255) NOT NULL,
  `race` VARCHAR(255) NOT NULL,
  `homeless` VARCHAR(255) NOT NULL,

```

```

    `mothers_name` VARCHAR(255) NOT NULL,
    `guardians_name` VARCHAR(255) NOT NULL,
    `deceased_DATE` DATETIME DEFAULT NULL,
    `deceased_reason` VARCHAR(255) NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Pharmacy information */
CREATE TABLE `pharmacy` (
    `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
    `name` VARCHAR(255) NOT NULL,
    `phone` VARCHAR(255) NOT NULL,
    `fax` VARCHAR(255) NOT NULL,
    `street` VARCHAR(255) NOT NULL,
    `postal_code` VARCHAR(255) NOT NULL,
    `city` VARCHAR(255) NOT NULL,
    `state` VARCHAR(255) NOT NULL,
    `country` VARCHAR(255) NOT NULL,
    `notes` TEXT NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Patient photo storage */
CREATE TABLE `photo` (
    `patient_id` BIGINT(20) NOT NULL,
    `data` VARCHAR(255) NOT NULL, /* We are storing filepath */
    `TIMESTAMP` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    PRIMARY KEY (`patient_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Prescription information */
CREATE TABLE `prescriptions` (
    `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
    `drug_name` VARCHAR(255) DEFAULT NULL,
    `dosage` VARCHAR(100) DEFAULT NULL,
    `count` INT(11) unsigned DEFAULT NULL,
    `refill` INT(11) unsigned DEFAULT NULL,
    `refill_limit` INT(11) unsigned DEFAULT NULL,
    `filled_DATE` DATE DEFAULT NULL,
    `notes` TEXT,
    `DATE_added` DATE DEFAULT NULL,
    `DATE_modified` DATE DEFAULT NULL,
    `patient_id` BIGINT(20) DEFAULT NULL,
    `physician_id` BIGINT(20) DEFAULT NULL,
    `pharmacist_id` BIGINT(20) DEFAULT NULL,
    `pharmacy_id` BIGINT(20) DEFAULT NULL,
    PRIMARY KEY (`id`),
    KEY `patient_id` (`patient_id`),
    KEY `physician_id` (`physician_id`),
    KEY `pharmacist_id` (`pharmacist_id`),
    KEY `pharmacy_id` (`pharmacy_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

/* Authorized user data */
CREATE TABLE `users` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `useremail` VARCHAR(50) NOT NULL,
  `password` VARCHAR(50) NOT NULL,
  `userlevel` INT(1) NOT NULL,
  `fname` VARCHAR(20) DEFAULT NULL,
  `mname` VARCHAR(20) DEFAULT NULL,
  `lname` VARCHAR(20) NOT NULL,
  `facility_id` BIGINT(20) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `facility_id` (`facility_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/* Add foreign keys to tables */
ALTER TABLE `history_data`
  ADD CONSTRAINT `history_data_ibfk_1` FOREIGN KEY (`patient_id`)
REFERENCES `patient_data` (`id`) ON DELETE CASCADE;

ALTER TABLE `immunizations`
  ADD CONSTRAINT `immunizations_ibfk_1` FOREIGN KEY (`patient_id`)
REFERENCES `patient_data` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `immunizations_ibfk_2` FOREIGN KEY (`administered_by`)
REFERENCES `users` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `immunizations_ibfk_3` FOREIGN KEY (`created_by`)
REFERENCES `users` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `immunizations_ibfk_4` FOREIGN KEY (`upDATED_by`)
REFERENCES `users` (`id`) ON DELETE CASCADE;

ALTER TABLE `insurance_data`
  ADD CONSTRAINT `insurance_data_ibfk_1` FOREIGN KEY (`patient_id`)
REFERENCES `patient_data` (`id`) ON DELETE CASCADE;

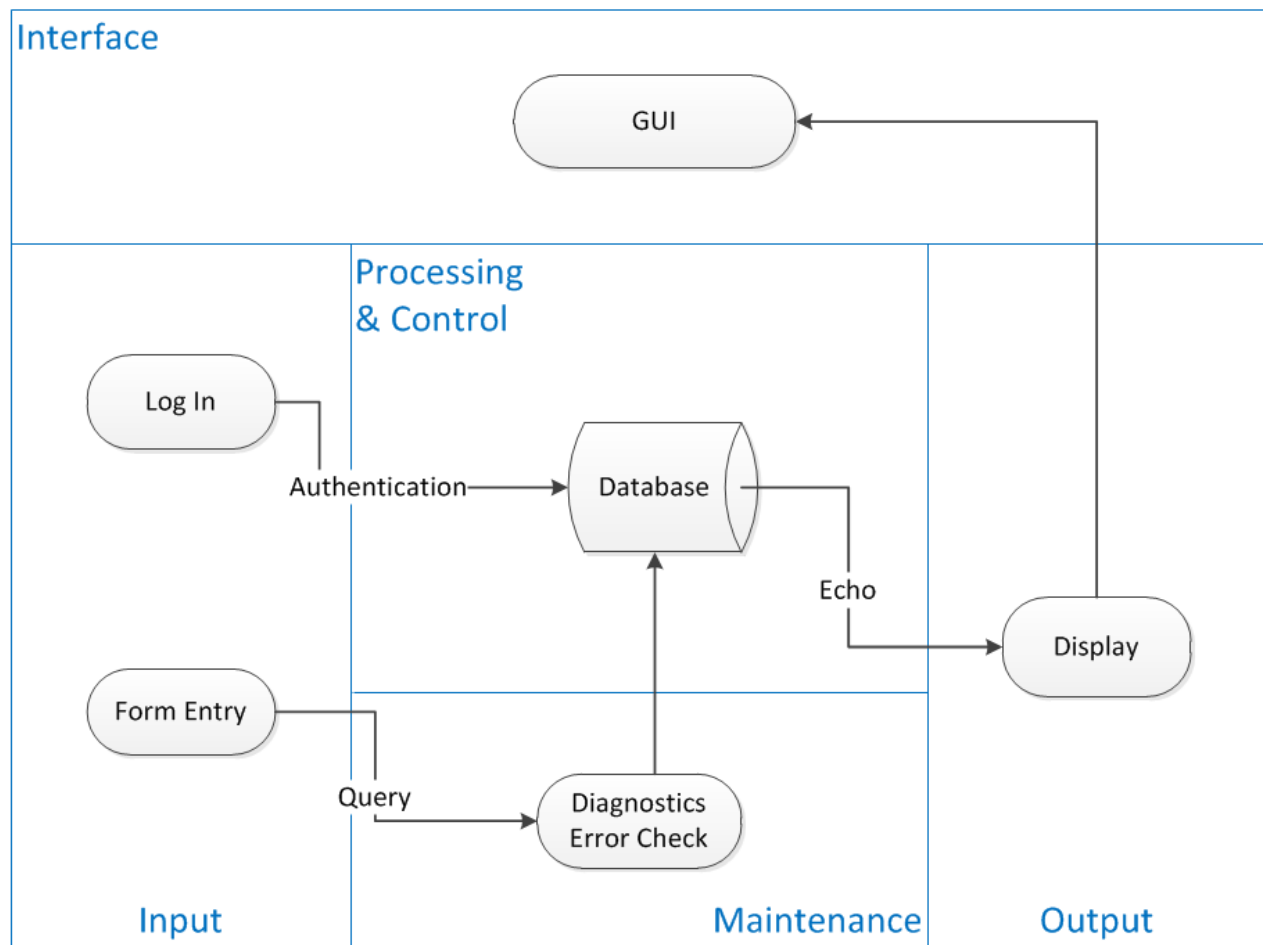
ALTER TABLE `photo`
  ADD CONSTRAINT `photo_ibfk_1` FOREIGN KEY (`patient_id`)
REFERENCES `patient_data` (`id`) ON DELETE CASCADE;

ALTER TABLE `prescriptions`
  ADD CONSTRAINT `prescriptions_ibfk_1` FOREIGN KEY (`patient_id`)
REFERENCES `patient_data` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `prescriptions_ibfk_2` FOREIGN KEY (`physician_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `prescriptions_ibfk_3` FOREIGN KEY (`pharmacist_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `prescriptions_ibfk_4` FOREIGN KEY (`pharmacy_id`)
REFERENCES `pharmacy` (`id`) ON DELETE CASCADE;

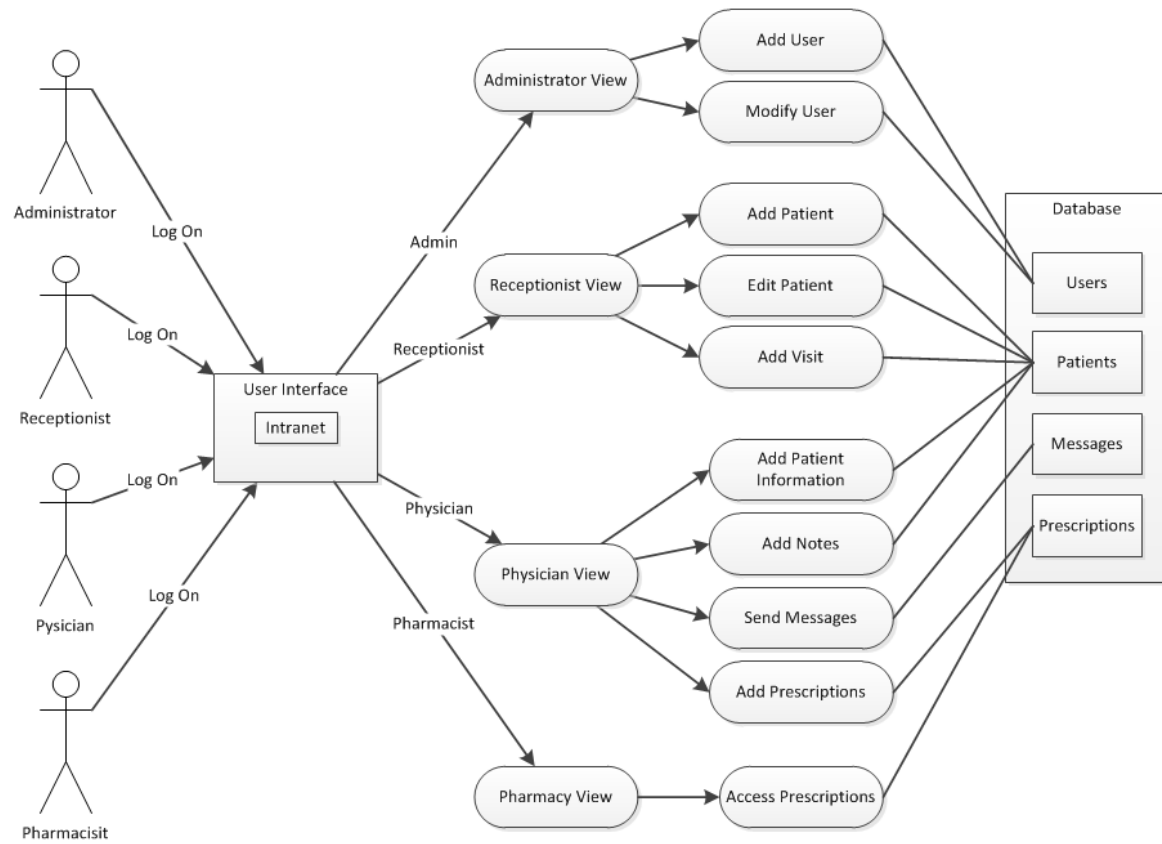
ALTER TABLE `users`
  ADD CONSTRAINT `users_ibfk_1` FOREIGN KEY (`userlevel`)
REFERENCES `level` (`id`) ON DELETE CASCADE,
  ADD CONSTRAINT `users_ibfk_2` FOREIGN KEY (`facility_id`)
REFERENCES `facility` (`id`) ON DELETE CASCADE;

```

3.2. System Context Diagram



3.3. Use Case



4. Technical Design

4.1. System Requirements

4.1.1. Programming Tools

PHP - Used to present server-side information to the client.

Javascript - The application will be using the jQuery 1.7.1 library.

4.1.2. Running Environment

Since this is a web application, the operating system or its hardware should not disrupt the Integrity of the application.

4.1.3. Database Server

The DBMS that our application will be using is MySQL.

4.1.4. Server

The database will be hosted on <http://pluto.hood.edu> once development is finished.

4.1.5. Client

A client is any authorized user specified in section 2.3 of the Software Specifications document.