**Mental health care-patient management system**

Business Requirements Document

Bill Keys, Mallika Devireddy, Scott Sigalas

Version 1.6

|  |  |  |
| --- | --- | --- |
| Modified By | Change Description | Date |
| Bill Keys | Updated Activity Models | 12/04/10 |
| Bill Keys | Updated Sequence Models | 12/05/10 |
| Scott Sigalas | Reformatted and Updated TOC | 12/12/10 |

Table of Contents

[1. DOCUMENT OVERVIEW 2](#_Toc279964132)

[2. BUSINESS REQUIREMENTS 2](#_Toc279964133)

[2.1 Industry Overview 2](#_Toc279964134)

[2.2 Project Overview 2](#_Toc279964135)

[2.3 Project Sponsor 2](#_Toc279964136)

[2.4 Constraints 2](#_Toc279964137)

[2.5 Team Members and Roles/Responsibilities 4](#_Toc279964138)

[3. FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS 4](#_Toc279964139)

[3.1 Functional Requirements 4](#_Toc279964140)

[3.2 Non-Functional Requirements 6](#_Toc279964141)

[4. ARCHITECTURE MODELS 8](#_Toc279964142)

[4.1 System Architecture 8](#_Toc279964143)

[4.2 Application Architecture 9](#_Toc279964144)

[4.3 Entity-Relationship Diagram 10](#_Toc279964145)

[5. USE CASES 11](#_Toc279964146)

[5.1 System-Wide MHC-PMS Use Case 11](#_Toc279964147)

[5.2 Use Case #1: Patient Death 12](#_Toc279964148)

[5.3 Use Case #2: Create Prescription 13](#_Toc279964149)

[5.4 Use Case #3: Patient Requests Information Change 14](#_Toc279964150)

[6. ACTIVITY FLOW CHARTS 15](#_Toc279964151)

[6.1 Patient Death 15](#_Toc279964152)

[6.2 Create Prescription 16](#_Toc279964153)

[6.3 Patient Request Information Change 17](#_Toc279964154)

[7. SEQUENCE DIAGRAMS 18](#_Toc279964155)

[7.1 Patient Death 18](#_Toc279964156)

[7.2 Create Prescription 19](#_Toc279964157)

[7.3 Patient Request Information Change 20](#_Toc279964158)

[8. CODELINES 21](#_Toc279964159)

[8.1 Codeline A: Use Case #1- Patient Death 21](#_Toc279964160)

[8.2 Codeline B: Use Case #2 – Create Prescription 21](#_Toc279964161)

[8.3 Codeline C: Use Case #3 – Patient Requests Information Change 21](#_Toc279964162)

[8.4 Codeline D: MySQL Database 21](#_Toc279964163)

[8.5 Codeline E: User Registration Form 22](#_Toc279964164)

# 1. DOCUMENT OVERVIEW

This document defines the high level requirements of the Mental Health Care – Patient Management System (MHC-PMS) software initiative. Included are business, functional, and non-functional requirements; UML artifacts used in development of the system; and a codeline for each aspect of the completed system.

# 2. BUSINESS REQUIREMENTS

## 2.1 Industry Overview

The United States mental health services industry includes about 15,000 establishments, consisting of both single and multi-location companies. Combined annual revenue totals over $20 billion. Major players in the industry include The Betty Ford Center, CRC Health, and Psychiatric Solutions. The industry is highly fragmented – the top 50 companies account for only 20% of industry revenue. There is plenty of room for small companies to thrive without being squeezed out of the marketplace by the big box centers.

## 2.2 Project Overview

This software project intends to automate many business processes necessary to run the day-to-day operations of a single or multi-locale mental health care company. These business processes may be currently done either manually or via disparate software systems. The goal of this project is to unify all these processes into a single system without losing any current functionality, while providing new features which have been requested by the client.

## 2.3 Project Sponsor

The Mental Health Care – Patient Management System software project is sponsored by Dr. Stephen Buerle, Professor of Computer Science at Marist College. Dr. Buerle represents the interests of several health care companies seeking to use the new MHC-PMS software solution.

## 2.4 Constraints

1. Usability: The system is used by a wide range of staff with varying levels of computer experience and hence this system must be user friendly
2. Safety: Patients may cause harm to themselves or others. The provisions of the Mental Health Act must be taken into account.
3. Privacy: Patient privacy must be maintained according to the Data Protection Act and local ethical and legal guidelines
4. Operational Costs: The cost of the system must be reasonable to the client
5. Completion Deadline: The system must be ready for delivery to the customer(s) no later than December 13, 2010

## 2.5 Team Members and Roles/Responsibilities

Mallika DeviReddy – Project and Business Manager

Bill Keys – Senior Programmer

Scott Sigalas – Senior Database Architect

# 3. FUNCTIONAL & NON-FUNCTIONAL REQUIREMENTS

## 3.1 Functional Requirements

1. Users should be able to create and modify patient information records

1.1. Receptionists should be able to make appointments

1.2 Clinical Staff should be able to modify patient records at time of treatment or at a later date

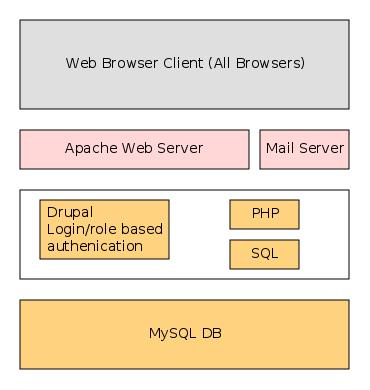
1. Users should be able to create/modify patient information records
   1. Receptionist should be able to make appointments
   2. Clinical Staff should be able to modify patient records at time of treatment or later date
      1. System should provide a method to re-use the condition & treatment from the previous appointment
2. Users should be able to retrieve data from the system
   1. Patient should be able to request their own personal.
   2. Clinical Staff should be able to look up patient information including appointment history, diagnosis history, prescriptions history,
   3. Receptionist should be able to search for records of individual patients
   4. Nurses going into the field should be able to download record on to their laptop and after upload back the modified records to the system.
3. System should generate periodic reports for the review of management
   1. System will generate reports automatically
      1. System will generate a daily list of patients who missed their appointments and email/SMS to the clinician responsible for the patients care
      2. System will generate a daily list of patients who attended an appointment but don’t have an updated record and email to the clinic where the patient attended the appointment
      3. System should generate regular reports on number of patients attended each clinic each month summary of prescriptions issued and the average wait time by the patients but omit the patient details
      4. System should generate weekly reports for each clinic the number of patients attending each day, the number of patients attended for a mental health treatment, the number of patients suffering from each condition and the total amounts of drugs provided as medication
   2. Medical Records Staff will generate health reports on requests by Mental Health Service management.
      1. System shall generate a list of patient conditions, treatments and the current care provider
4. System must generate warnings/ alerts
   1. **System will generate warnings about medication interaction but may be overridden by the prescriber**
   2. **System will generate a warning if too much medication has been prescribed**
   3. System will highlight the date of the previous consultation if it was a walk-in appointment
   4. System will generate a warning letter to clinical staff and patient relatives about the patient condition and possibility of deliberate self-harm
   5. **System should allow information about adverse reactions or side effects for any drug or medication and warning message should be generated upon prescribing a medication to which a patient is allergic.**
   6. System should highlight the patients with a high or increased risk classification level
5. **The system must have a feature for handling the death of a patient**
   1. **When a death of a patient happens the system will lock the patient’s record**
   2. Within 3 mounts of death the patient’s record will be remove from the MHCPMS system and be stored in the archive system
6. **Patients should be able to request personal information and request changes**
7. System will support printing functionality for medication prescriptions or reports etc.
8. System should allow risk assessment when the patients risk classification (low/medium/high) changes from one level to another.

## 3.2 Non-Functional Requirements

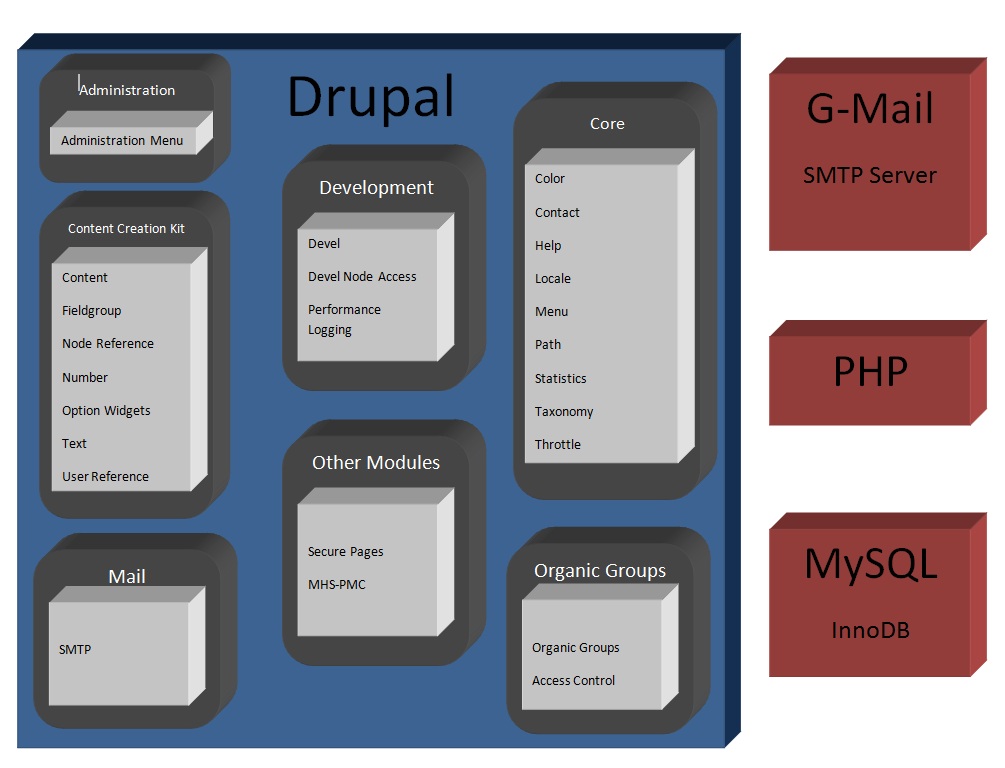
1. Security Requirements
   1. System should restrict access to authenticated users and not allow these users to access information outside their role.
   2. Patients privacy must be maintained accounting the Data Protection Act and local ethical rules
   3. System should provide role-based access to data
      1. Patients accessing the system should only have access to their own data
      2. Receptionist should have only access to the appointment details but not the diagnosis information of the patient
      3. Clinical staff should be able to view records and treatment history for patient under their own care
      4. Medical records staff should have access to all data except personally identifying information
2. Operational costs, which may include Maintenance and Running costs, must be “reasonable”.
3. System must be user friendly
4. Patients can cause harm to themselves or others so the system must take this into account
5. Reliability Requirements
   1. MHCPS system should run on the same system as Appointment diary system for easy transfer of data from Appointment system to MHCPMS system.
   2. MHCPMS data server should be maintained as a separate computer in a physically secured location.
   3. Data server should be backed up onto tape or disk each evening at 1800 and copies of backups should be saved in a separate place.

# 4. ARCHITECTURE MODELS

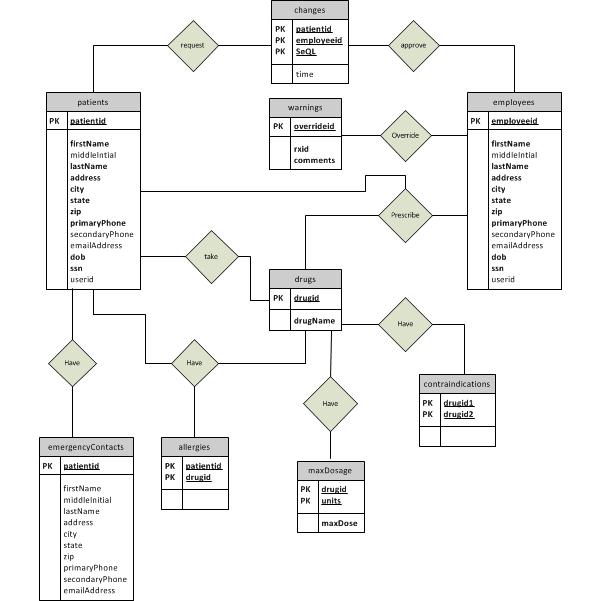
## 4.1 System Architecture



## 4.2 Application Architecture

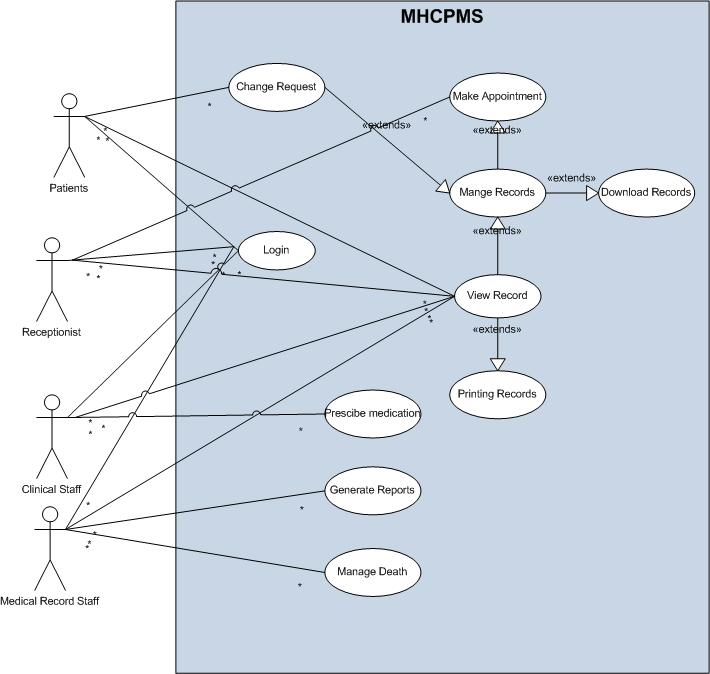


## 4.3 Entity-Relationship Diagram



# 5. USE CASES

## 5.1 System-Wide MHC-PMS Use Case



**Actors:** Patients, Receptionists, Clinical Staff, Medical Records Staff

**Pre-Condition**: Each actor must log in to interact with the MHC-PMS system.

**Description:**

**Patients** may view their own medical records. They may request changes to these records.

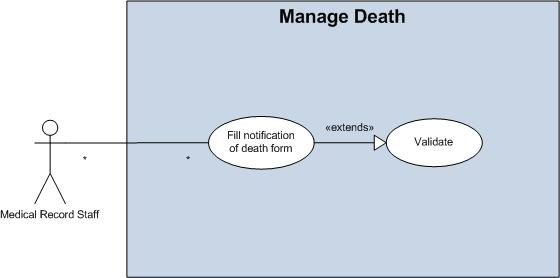
**Receptionists** create appointments and view a patient’s appointment history.

**Clinical** **staff** can create, view, and manage patient medical records. They may download these records to an authenticated laptop. They can prescribe medication.

**Medical** **Records** S**taff** can generate reports at the request of management and can also initiate the archival process associated with the death of a patient.

**All actors who can view records have the ability to print them.**

## Use Case #1: Patient Death

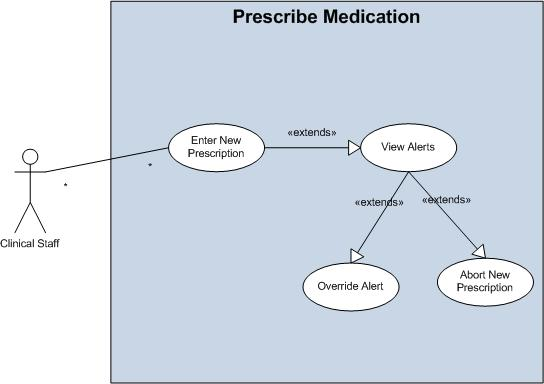


**Actors**:Medical Records Staff

**Pre**-**Condition**:Each actor must log in to interact with the MHC-PMS system.

**Description**: When a clinic receives notification that a patient has died, a member of the medical records staff will fill out a form with the patient’s ID number. The system will retrieve the name of the patient and give the user an opportunity to validate that it is the correct patient. Once the user confirms that the correct patient was entered, then the system will prevent any future modification of the patient’s appointment, prescription, and medical records.

## Use Case #2: Create Prescription

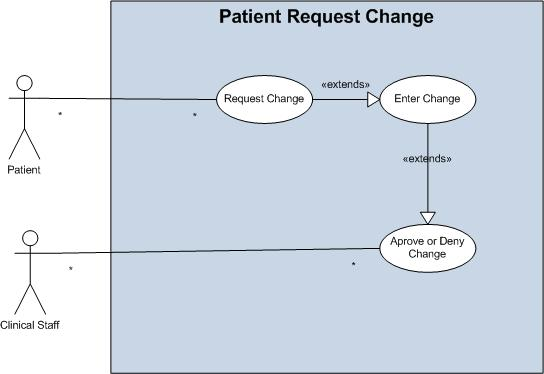


**Actor**: Clinical Staff.

**Pre-Condition**: Each actor must log in to interact with the MHC-PMS system.

**Description**: When a clinician wishes to prescribe medication, he will enter into the system the patient ID and the medication to be administered. The system will retrieve a list of all medications currently being taken by the patient from the database and cross-reference all permutations with the new medication for any known interactions. If there is a contraindication, then the clinician will receive a warning. The system will also retrieve the patient’s history with the drug to be administered. If there has been too much prescribed over the past 90 days, a warning will be issued. The system will also check to see if the patient has a known allergy to the medication. The user will receive either validation that the prescription poses no known danger to the patient, or a warning that the prescription is not advised. The clinician has the option to override the warning. If this happens, then the override will be captured and logged by the system. The doctor also has the option to abort the prescription. If the prescription is to be administered, it will be printed by the system.

## Use Case #3: Patient Requests Information Change



**Actors**:Patient,Clinical Staff

**Precondition**:Each actor must log in to interact with the MHC-PMS system.

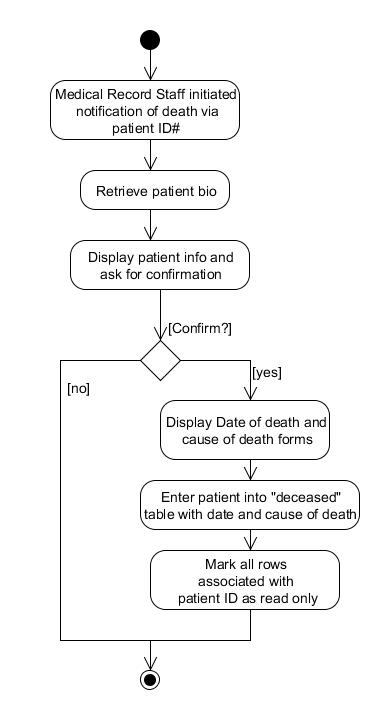
**Description**:

If a patient wishes to change a portion of his medical record, he will fill out a web-form indicating which items he wishes to change. The system will provide an input field for the new values. Once the user submits this form, a notification email will be sent to the patient’s primary clinician, asking the clinician to accept or reject these changes.

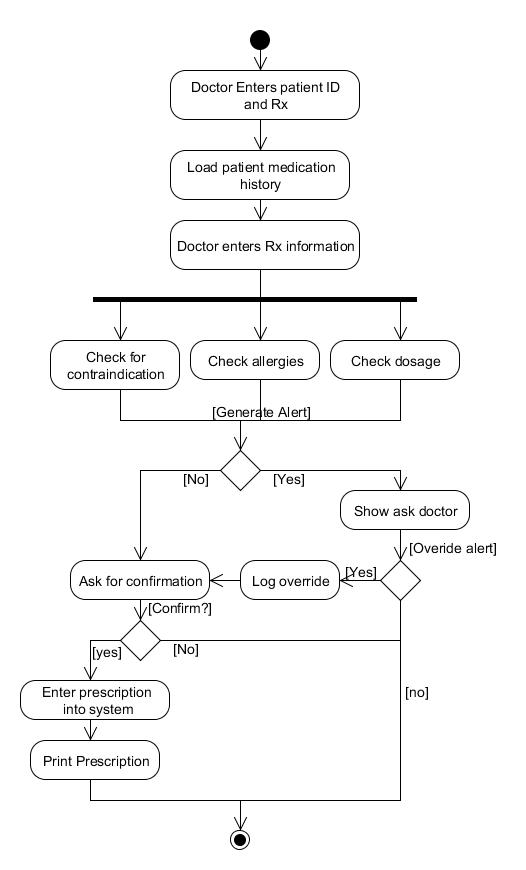
The doctor will log-in and will have the ability to accept or reject each individual modification that was requested by the patient. Depending on the response, the database may or may not be updated. The patient will be notified of the doctor’s response via email.

# 6. ACTIVITY FLOW CHARTS

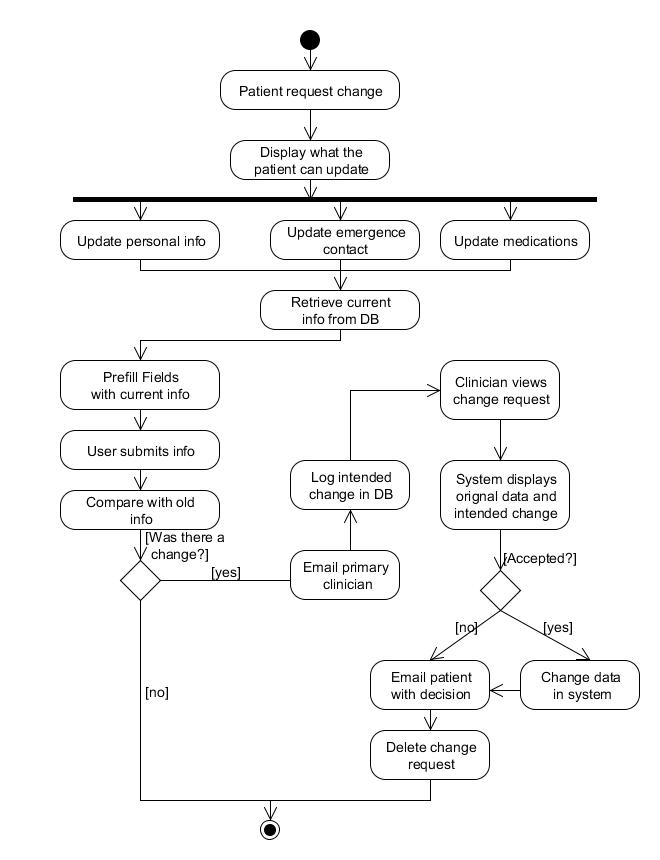
## 6.1 Patient Death



## 6.2 Create Prescription

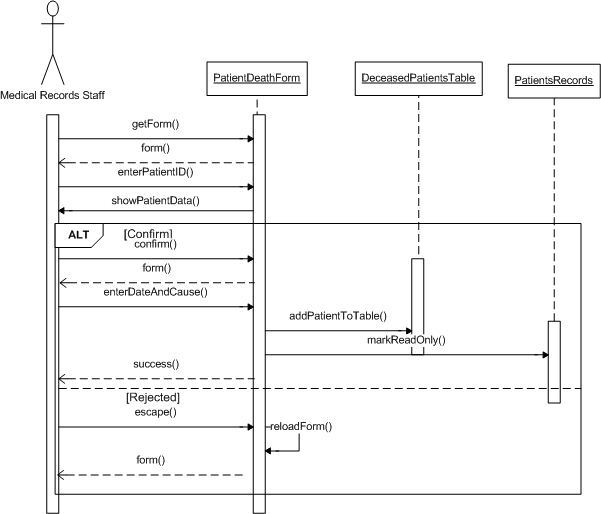


## 6.3 Patient Request Information Change

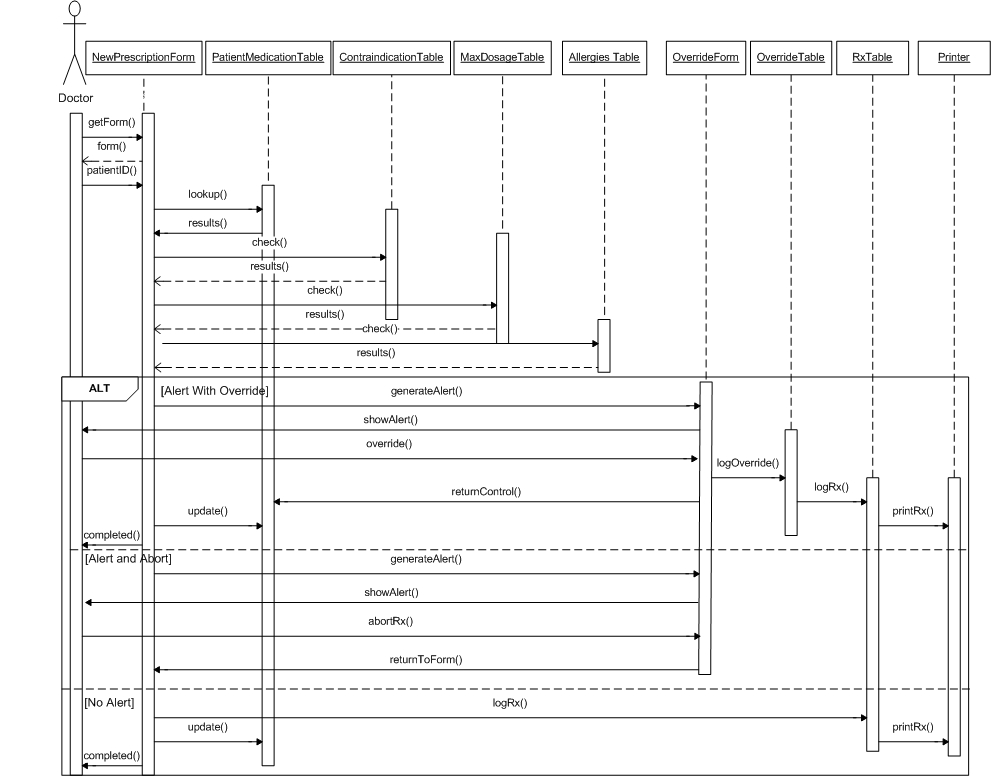


# 7. SEQUENCE DIAGRAMS

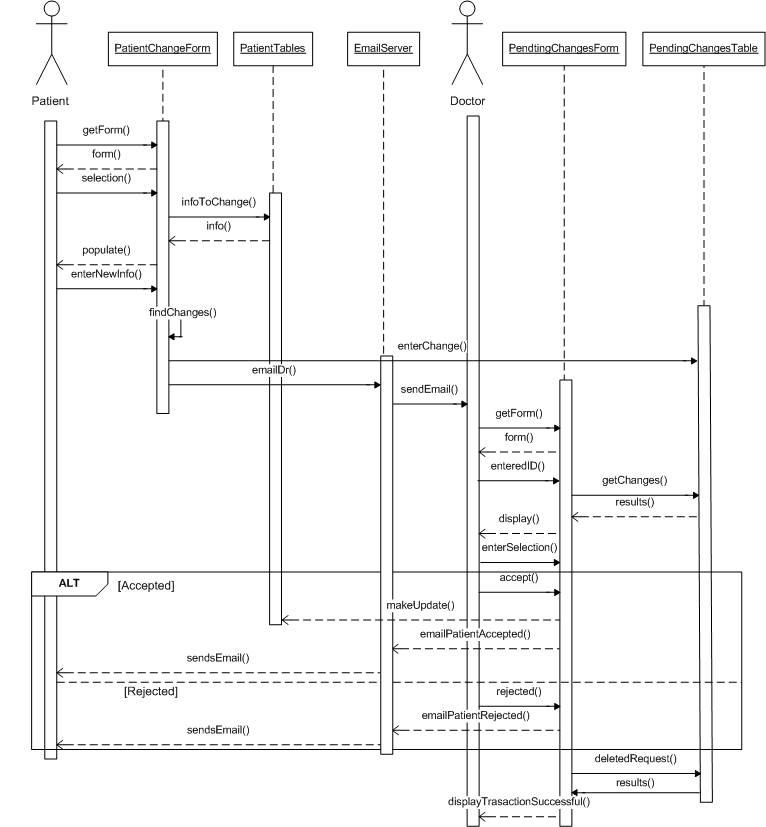
## 7.1 Patient Death



## 7.2 Create Prescription



## 7.3 Patient Request Information Change



# 8. CODELINES

## 8.1 Codeline A: Use Case #1- Patient Death

A1.0 – Initial version of form to enter patient death, confirm, and update database

A1.1 – Added function to prevent updating patient records after patient dies

## 8.2 Codeline B: Use Case #2 – Create Prescription

B1.0 – Initial version of form to enter new prescription including safety checks for contraindications, allergies, and overdose. Also includes opportunity for doctor to override safety warning

B1.1 – Fixed bug in calculation of maximum dosage

## 8.3 Codeline C: Use Case #3 – Patient Requests Information Change

C1.0 – Initial version of Patient Request Change Form features place for patients to enter changes to personal information but does not communicate with database

C1.1 – Passes changes to the database and new Pending Changes Form displays changes yet to be reviewed. Only functions for personal information

C1.2 – Added support for emergency contact and medication changes in both forms. Fixed bug with timestamp conversion and display.

## 8.4 Codeline D: MySQL Database

D1.0 – Initial version of database features only 2 tables (patients, deceasedPatients) used to support Use Case 1

D1.1 – Added tables to database to reflect requirements of ER diagram. Created foreign key constraints and added sample data.

D1.2 – Changed several date fields to timestamp for simpler entry

D1.3 – Added userid field to 2 tables (patients, employees) to track UserID in the front end

D1.4 – Added table to link a patient to his/her primary doctor

## 8.5 Codeline E: User Registration Form

E1.0 – Added form to be used by administrator to link a patient or employee to a registered user and populate the database accordingly. Allows other forms to authenticate the current user as the user they claim to be.

## 