

# Software Requirements Specification (SRS)

## EMR Data Analysis

**Authors:** James Drallos, Jordan Clare, Joseph Korolewicz, Daniel Laboy

**Customer:** Dr. Gary Ferenchick

**Instructor:** Dr. Betty Cheng

### 1 Introduction

This document outlines the software requirements for an electronic medical record (**EMR**) data analysis clinical decision support system. It will cover the overall description of the system, specific requirements as well as modeling requirements, diagrams and a description of a prototype to be built to demonstrate the system's functionality.

#### 1.1 Purpose

This document is intended to inform those who have a vested interest in the EMR data analysis clinical support system of its purpose and design. It should be useful to the customer(s) as well as any members of a software development team who are tasked to build and/or maintain the system itself. Users of the system need not concern themselves with the information in this document unless they desire a deeper understanding of the system's architecture. However, this document does not focus on ways in which the end-user will interact with the system to perform its intended tasks.

#### 1.2 Scope

The software system of which this document concerns is an EMR data analysis-based clinical decision support system (**CDSS**). The system will be a contained software package that will be installed on and reside in the computers of a medical care facility in order to better assist medical personnel in the diagnosis and treatment of patients. It is intended to focus on patients exhibiting symptoms of Methicillin-Resistant *Staphylococcus aureus* (**MRSA**), a common virus which has become resistant to typically prescribed drugs. The system will provide access to patient data contained in an EMR such as symptoms, medical history and current prescriptions. The system will then evaluate known data about a patient and use it to suggest a clinical diagnosis and a case-specific treatment plan. The system will receive data from remote databases including

relevant clinical trials which can be prescribed and medical best practice guidelines to be followed. Treatment will then be monitored. The system will improve the quality of medical care by providing a wealth of data from various sources to medical professionals, by promoting the use of best practices and tracking treatment and by suggesting a diagnosis based on data about previous cases.

### 1.3 Definitions, acronyms, and abbreviations

**CA-MRSA** – community-acquired MRSA

**CDSS** – Clinical Decision Support System. Software which generates case-specific advice based on a medical knowledge base, patient data and an inference engine.

**EMR** – Electronic Medical Record. A digital form of a patient's past medical information.

**EMRSA** – epidemic MRSA

**HA-MRSA** – Health-care-acquired MRSA

**MRSA** – Methicillin-Resistant *Staphylococcus aureus*. The medical condition targeted by the system for evaluation.

### 1.4 Organization

This document is organized in seven sections with various subsections describing various aspects of the system. Section two gives the overall description of the product, including major functions and constraints. Section three lists specific requirements, section four contains diagrams outlining the design and use of the system, section five describes the product prototype, section six lists references and section seven gives the point of contact for questions about the product.

## 2 Overall Description

This section contains the complete description of our project. It states the perspective of the system, how the system is just one part of a complete EMR. This section also shows the constraints faced by the system, and the assumptions we made both about the systems it will be interacting with, and the people who will be using the system.

### 2.1 Product Perspective

Our system is the data analysis portion of an electronic medical record. Our system will, given a diagnosis from a doctor about a certain patient, find and create a relevant treatment plan for that patient. Our system will suggest these treatments using information regarding patient specific information, medical best practices and relevant clinical trials. As you can see in figure 2.1, our TreatmentManager will get all of this information and then be able to design a treatment.

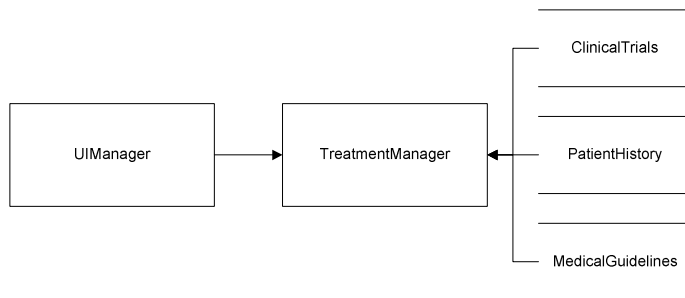


Figure 2.1. TreatmentManager

Our system is just one part of a complete EMR system. The other pieces of an EMR system are the data collection and data dissemination pieces respectively. The data collection piece of the system would be required to collect all of the input given by medical professionals, and store all of this data into an accessible location. Our project would use this information and make a treatment decision using all of this data. The other piece of a successful EMR, the data dissemination piece, would take the information, and find current medical trends. It would check whether people from a certain area were more prone to a certain disease. See figure 2.2 for an example of a complete EMR system. Notice that the data dissemination and data analysis pieces run concurrently. This is because both systems take the same data given from the data collection section, and use it in different ways.

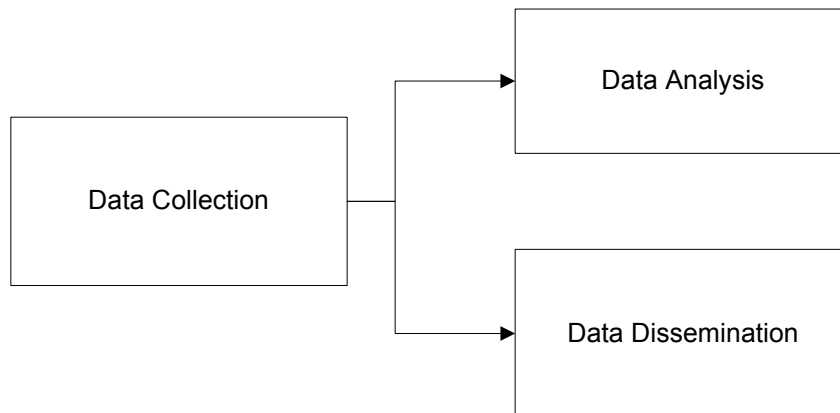


Figure 2.2. Complete EMR System

For our system to work there must be certain information present. There must be databases set up for patient history, medical best practices, and clinical trials. Our system would then check these databases, and pull out relevant information. It would then use this information to build a treatment which would be customized for the current patient and current state of the medical field.

## 2.2 Product Functions

The system shall provide customized treatment plans for patients who have been diagnosed with different diseases.

The system shall confirm a clinical diagnosis given by a medical professional by checking the symptoms present with the diagnosis.

The system shall use a patient's medical record in order to further provide customized treatment plans.

The system shall provide access to relevant clinical trials for a given patient, and give a medical professional the choice of whether or not he or she would like to try and use this particular trial for their current treatment.

The system shall provide a plan for preventive care for its patients.

The system shall show the medical best practices of a given diagnosis, and remind a physician as to what guidelines have not yet been completed.

The system shall track previously diagnosed patients, and check to see if there are any treatments which still need to be done.

The system shall be able to differentiate whether a patient has a certain kind of MRSA, and how to treat that patient.

The system shall check a current diagnosis and find out whether or not the doctor's diagnosis is accurate.

The system shall abide by HL-7 standards.

## **2.3 User Characteristics**

The user must be able to properly input the symptoms or diagnosis, as if any incorrect input is put into the system, the treatment plan provided by the system would not be valid in any way. Also, the user needs to have some kind of medical experience, and must have a certain level of certification. The user needs to be able to understand what the treatment the system provides actually is, and can be able to act upon that treatment accordingly.

## **2.4 Constraints**

The system must have access to databases which contain medical records of patients, relevant clinical trials, and medical best practices for different diseases.

The system will only work for diseases which are in the database, if the patient has a disease that our databases do not have a record of, our system will not accurately be able to provide useful information.

The input into the system must be correct, if a doctor enters a wrong diagnosis, the treatment provided will not be relevant to the current patient.

## 2.5 Assumptions and Dependencies

The system assumes that there are databases set up which hold patients medical records, ongoing clinical trials, and medical guidelines for specific diseases, and that our system has access to these pieces of information. As without this information, our system has no information to base its recommendations for treatment on.

## 2.6 Apportioning of Requirements

Our customer stated that it would be ideal for our project to contain voice recognition for a method of input, however, this just does not seem to be an option given the time frame.

# 3 Specific Requirements

### 1. Manage clinical information:

#### 1.1 Identify and prescribe relevant clinical trials / protocols:

The system will query the list of clinical trials hosted on [www.clinicaltrials.gov](http://www.clinicaltrials.gov) and determine if there are any ongoing trials that are relevant to the patient's diagnosis and if so will display information about the trial to the physician.

#### 1.2 Track orders for treatment:

The system will allow the user to view information regarding both the patients treatment history as well as any treatment that has been scheduled. Information about previously administered treatments will be available in a variety of forms depending on the type of treatment, for example X-rays, lab reports, or prescriptions. Information pertaining to treatments that have been

scheduled to take place sometime in the future will have contain the proposed date and location as well as any notes the physician has entered regarding treatment customization. Once the treatment has been completed the relevant artifact will be available for viewing.

### 1.3 Referral follow-up:

The system will allow the user to view information about all treatments the patient received including referral treatments provided the treatment took place in a hospital using the same or compatible system.

### 1.4 Prescribe preventative care:

The system will provide the user with guidelines for relevant preventative care for a given diagnosis.

## 2. Decision Support:

### 2.1 Confirm a clinical diagnosis:

The system will compare the problem list the user has acquired based on a patients symptoms and compare them against a database of symptoms for a given diagnosis. If the system finds the diagnosis to be inconsistent with symptoms, the system will display a message to that effect as a warning to the user.

### 2.2 Suggest treatment plan processes:

The system, when given a diagnosis, will compare treatments the patient has already received to a list of possible treatments for that diagnosis. The system will then suggest the relevant treatments that have not already been administered.

### 2.3 Promote the use of “best practices”:

The system, when presented with a diagnosis, will present a list of generally accepted treatments for the diagnosis. The list will be retrieved from a database maintained by the system.

#### 2.4 Customize treatment according to specific conditions:

The system will allow the user to enter any notes or special instructions when scheduling a treatment. Any special instructions that a user adds will be viewable by any other user observing the patient's medical history.

#### 2.5 Distinguish between various types of MRSA:

##### 2.5.1 Determine if MRSA is Health-Care acquired

The system, when presented with a diagnosis of MRSA, will check its database to see if the patient has been recently admitted or has been in staying in the hospital. If so, they system will display a message to the user indicating the possibility of HA-MRSA.

##### 2.5.2 Determine if MRSA is the result of an epidemic

The system, when presented with a diagnosis of MRSA, will check the MRSA Health Surveillance System to see if there has been an unusually large number of MRSA cases in the area.

### 3. Compliance with HL-7 standards

The system will conform to HL-7 standards in all data storage and message passing protocols.

## 4 Modeling Requirements

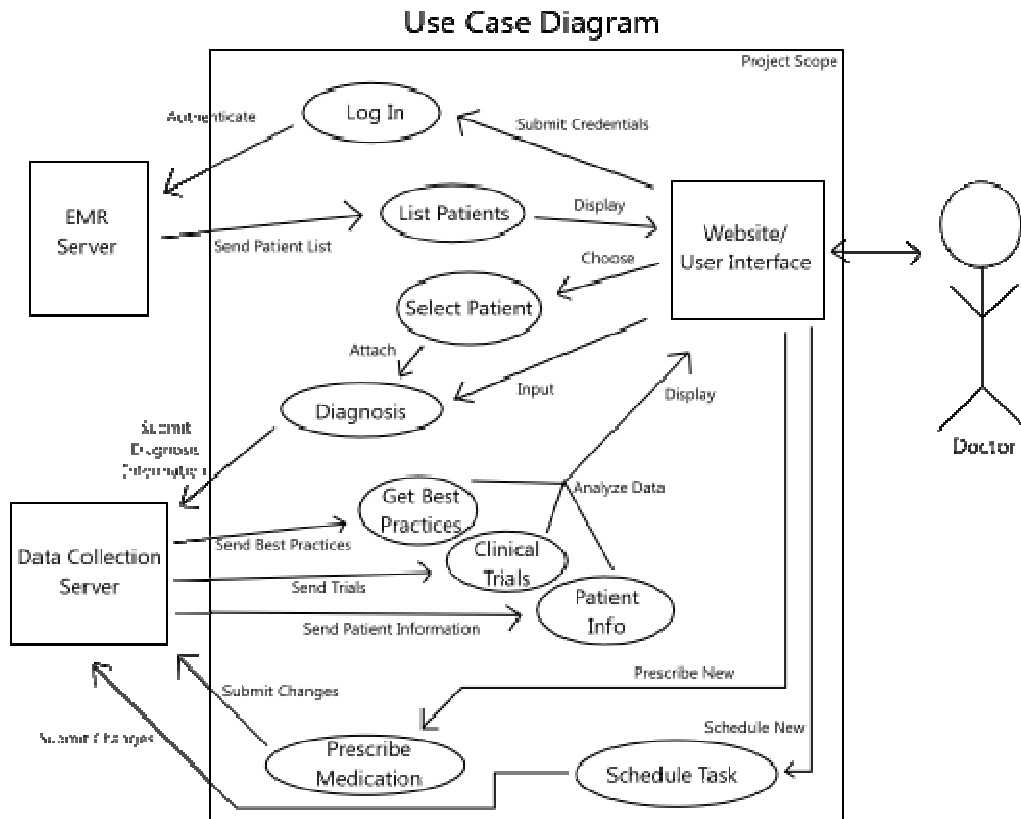


Figure 4.1. Use Case Diagram

Figure 4.1 depicts the typical sequence of events a user (in this case, a doctor) will traverse while using the software design. The sequence will flow from top to bottom, with the final five steps being repeatable at any time after their initial calls.



Figure 4.2 (below) is the class diagram for the system. The PatientInfo class is a collection of all the relevant information about a patient such as height, weight, vitals, as well as a reference to the patient’s medical history which will have all the information about previous treatments, diagnoses, as well as current symptoms. The Diagnosis class contains a reference to the BestPractices class which is a collection of widely approved methods for treating a given diagnosis. The UIManager coordinates the different subsystems and lets the user move easily between various tasks such as looking at a patient’s history and scheduling a new treatment. Each instance of the Treatment class, which stores information about a single treatment that a user can administer, also has zero or more Artifacts associated with it. The Artifact class has a digital copy of various results of a given treatment, such as an X-Ray, lab result, or copy of a prescription.

EMR – Data Analysis Class Diagram

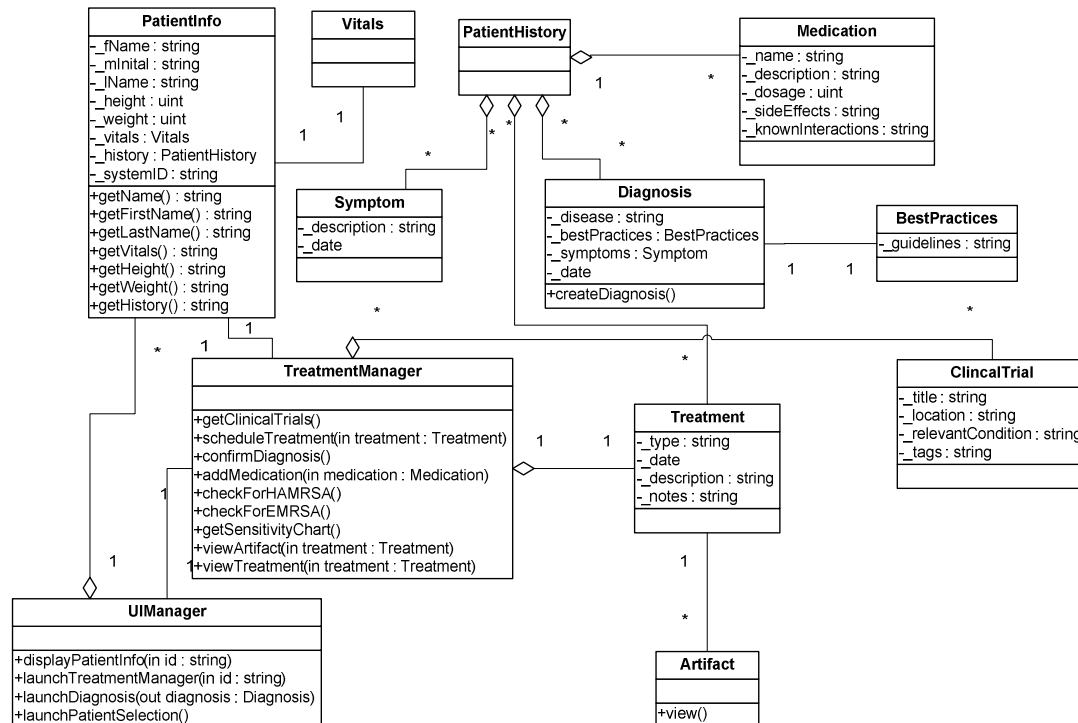


Figure 4.2. EMR – Data Analysis Class Diagram

Figure 4.3 is a sequence diagram displaying one example use of the system from the initial login until treatmentManager has finished initializing.

Sequence Diagram 1 – System login to initialization of TreatmentManager

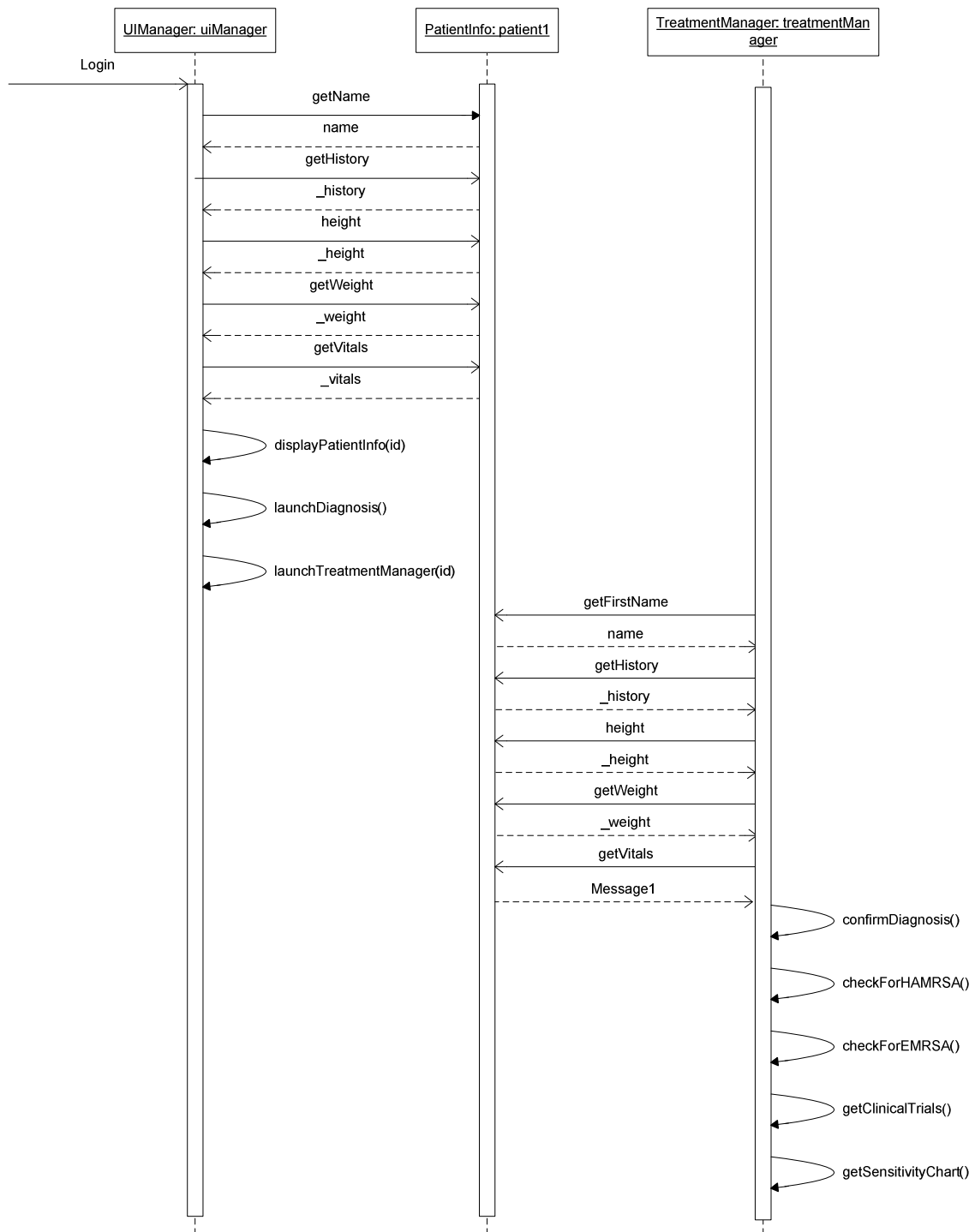


Figure 4.3 Sequence Diagram 1

Figure 4.4 is a sequence diagram illustrating a scenario in which the treatmentManager has already been launched. In this example the user views the details of a past treatment, it's resulting artifact, schedules a new treatment and prescribes a medication.

Sequence Diagram 2 – Various Treatment Operations

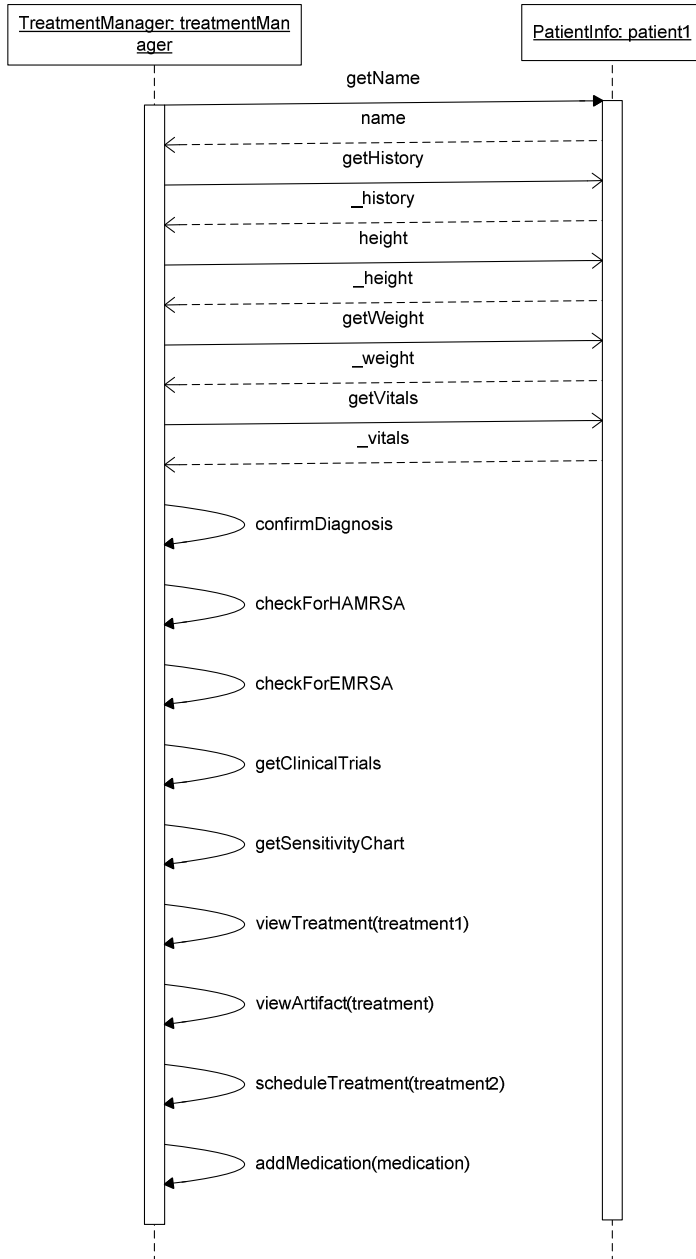


Figure 4.4 Sequence Diagram 2

Figure 4.5 (below) is a state diagram for the UIManager class. The diagram depicts the three main states of the UIManager, Patient Selection, Diagnosis, and Treatment as well as how to transition between them. When a user logs on to the system they are initially placed in the Patient Selection state. From here they have the choice of entering a diagnosis for a patient (Diagnosis state) or entering a treatment for a patient that has already been diagnosed (Treatment state). If the user chooses to enter a diagnosis, upon doing so they will then transition to the Treatment state.

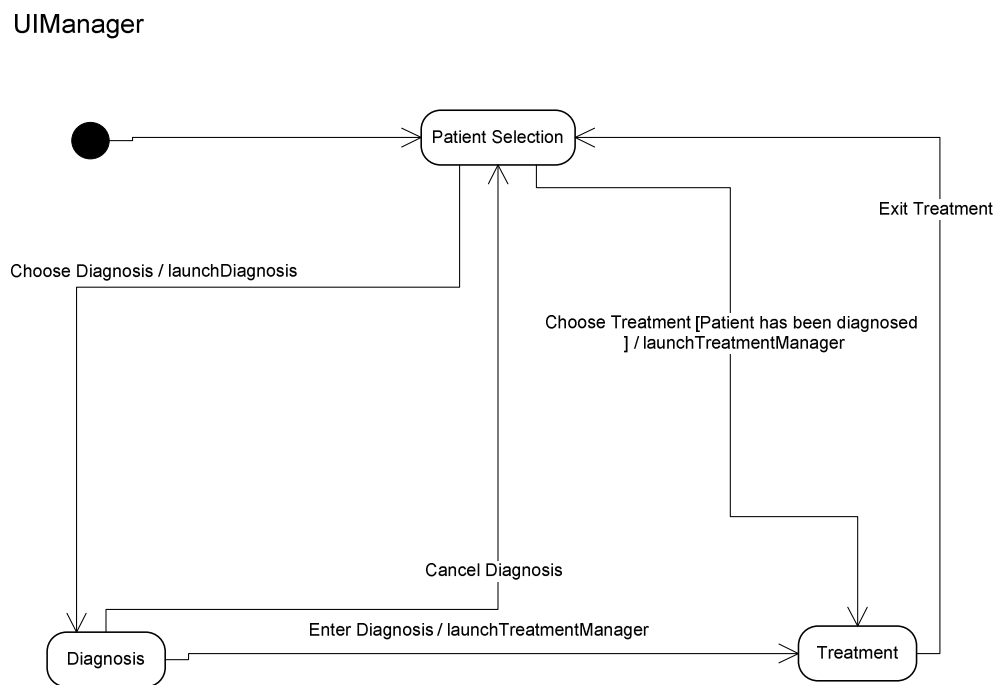


Figure 4.5. UIManager State Diagram

Figure 4.6. (below) is the state diagram for the TreatmentManager class. Initially whenever the Treatment Manager is launched it enters the Analyze state. In this state various checks and analyses are performed such as confirming the user's diagnosis and in the event of a MRSA diagnosis determining the type of MRSA the patient may have. Once these steps are complete the TreatmentManager transitions into the View Treatment Plan/Progress state. From here the user can view the progress of treatments that have already been carried out, see a list of upcoming treatments, view a list of relevant clinical trials, as well as choose to schedule a new treatment for the patient. If the user chooses to view the progress of a treatment that has already been scheduled the TreatmentManager enters the Detailed View state and from here the user can view more detailed information about the treatment as well as any artifacts if the treatment has already been completed. If the user chooses to schedule a treatment the TreatmentManager enters the Schedule Treatment state and the user can then schedule a new treatment.

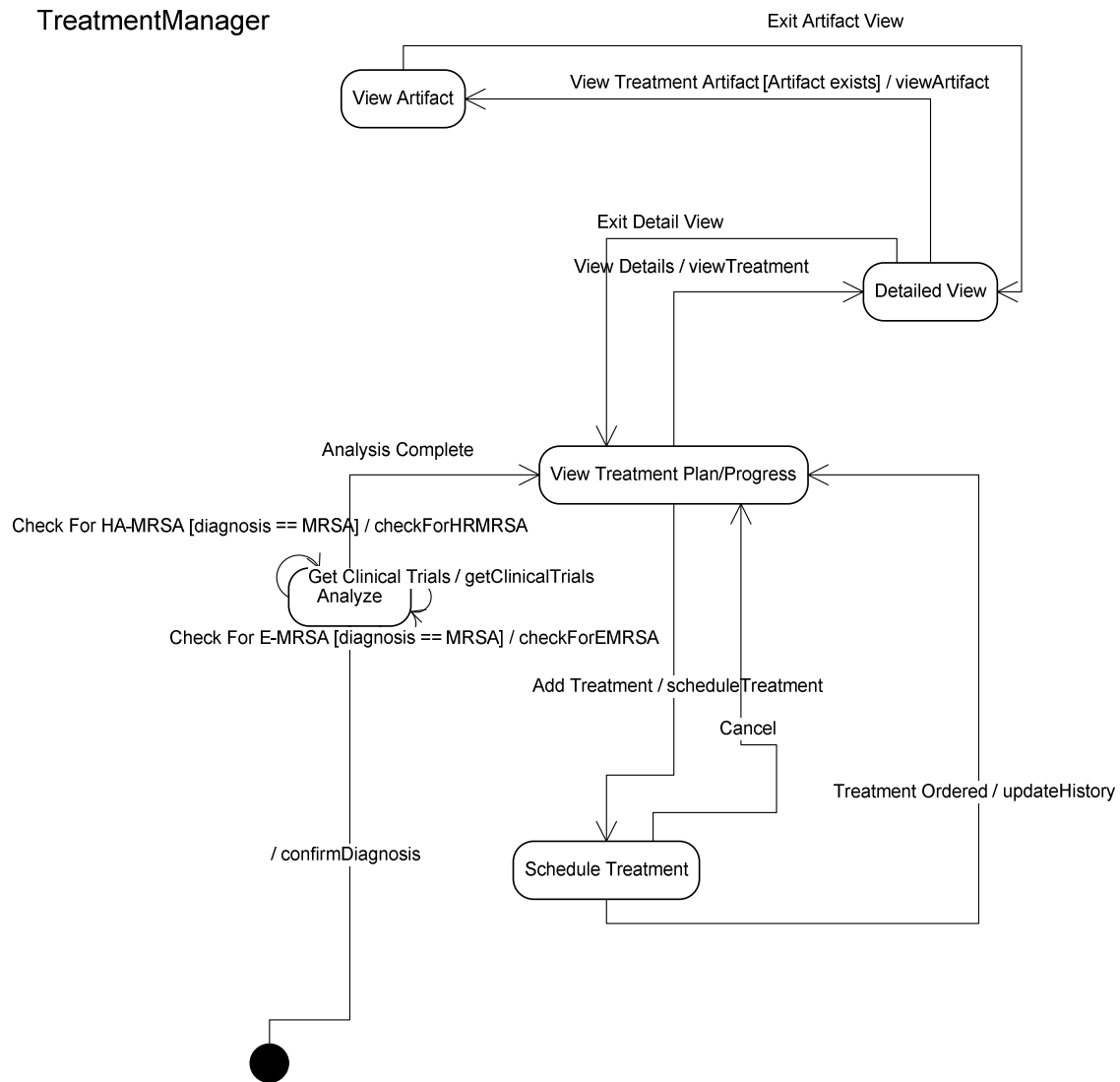


Figure 4.6. TreatmentManager State Diagram

## 5 Prototype

The prototype will illustrate the interaction between a medical professional and the designed system to diagnose a case of a type of MRSA. This process will show how the doctor will be able to use the designed system to create, customize, and maintain a treatment plan for the specific patient relevant to the diagnosis, along with decision support provided by the software.

### 5.1 How to Run Prototype

Our prototype will be implemented using Microsoft's Silverlight, version 3. Microsoft Silverlight is a cross-platform, cross-browser plug-in that is similar to the more widely used Adobe Flash. We decided to use Silverlight for this application because it allows us to easily to create and customize rich application user interfaces either directly using XAML encoding or through a separate program developed by Microsoft (Microsoft's Blend) to facilitate the design and customization of applications using Silverlight. Silverlight also allows us to implement custom interactions within the UI elements by utilizing code behind files written in the C# programming language. Another reason we have decided to use Silverlight is that the default toolkit of UI elements provided to design user interfaces is simple and has high aesthetic value.

In order to run the application, the only requirements are that the user is on a computer with an updated browser (Firefox, Safari, IE, etc.) and that they download a small (4.3 MB) plug-in when they navigate to the application for the first time. Silverlight is not platform or browser specific, so there are no hardware, networking or operating system requirements.

Our prototype can be accessed through the web via a link on our team website at <http://www.cse.msu.edu/~cse435/Projects/F09/EMR-Analysis/web/>.

### 5.2 Sample Scenarios

When a doctor, or any other medical professional, starts the prototype he/she will first encounter the log in screen shown below in Figure 5.1. The doctor will enter his/her user credentials in the appropriate fields and then click the “Log In” button, which will authenticate on the main EMR server.

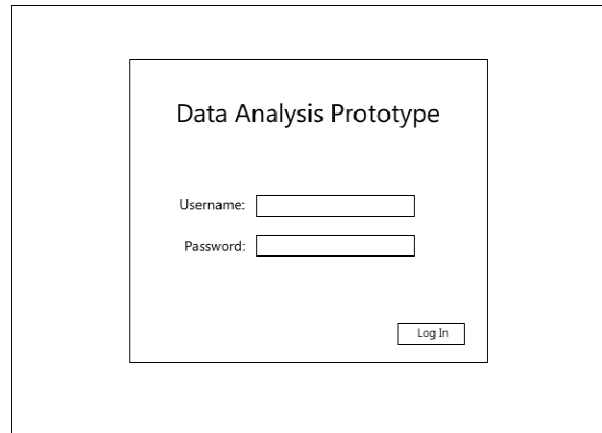


Figure 5.1

Upon logging in to our system, the user will next encounter the home screen shown in Figure 5.2 below. This home screen will have a table in the upper third of the screen that contains a list of all patients accessible to the doctor. Below the patient list will be the patient detail view area. If the doctor selects a patient from the list, the detail view area will show important information about the selected patient, including: full name, measured height and weight, any recent vitals measurements, and a brief overview of the patients medical history.

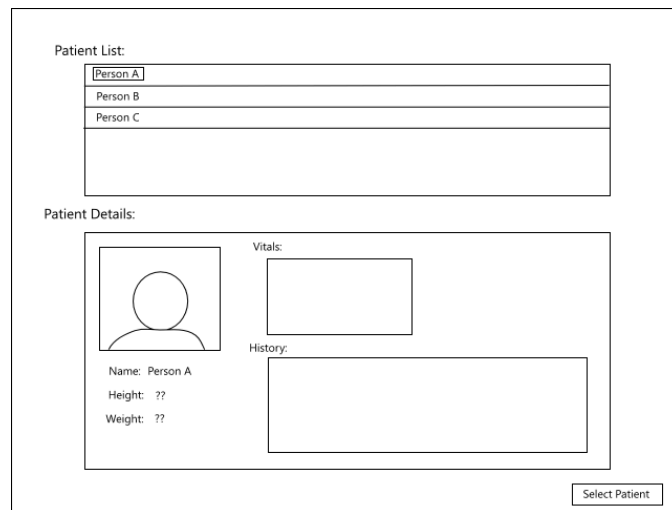


Figure 5.2

After the user has selected a patient and has completed any review of information contained in the detail view, the user can click on the button at the bottom of the screen which brings him/her to the diagnosis screen, shown in Figure 5.3. Considering the scope of our project, Analysis and Decision Support, the design of this page is very simple and consists of a single text box and a "Next" button. The user will enter his/her diagnosis (as

shown in the diagram, MRSA) in this box and click the button to continue to the next screen.

Figure 5.3

After submitting the diagnosis, the program submits the information to the data server for cataloging and queries all relevant information on the given diagnosis and patient. The following screen, labeled the Treatment screen, shown in Figure 5.4, will display the received information from the previous query for the user to review.

Best Practices:	
✓	Task 1
?	Task 2
X	Task 3

Current Medication:	
MRSA Drug 1	⚠
MRSA Drug 2	⚠

Tasks Completed:	
-Task 1	

Tasks Scheduled:	
Task 2	1/1/2010 12:00 AM

Clinical Trials:	

Figure 5.4

The information contained in the table in the top left of the screen, shown in Figure 5.5, is the “Best Practices” of the corresponding diagnosis. This table lists the most widely accepted treatment plan for the diagnosis, and indicates, with green check marks, if any of those steps have been completed for the patient. Any incomplete tasks will be marked with a “X” mark and highlighted in red. Any tasks in the “Best Practices” list that have been scheduled for the patient, but have not yet been completed are highlighted in blue with a “?” symbol.



**Best Practices:**

✓	Task 1	Completed: 12/15/2009
?	Task 2	Scheduled: 1/1/2010
X	Task 3	

Figure 5.5

In the bottom left of the screen, shown in Figure 5.6a, are two tables labeled “Tasks Completed” and “Tasks Scheduled.”

The first of these tables, Tasks Completed, lists any and all work orders that have been performed on the patient. The user can select any of the items in this list, and click the “View Details” button below the table to bring up a window that will display the results and details of the work order, shown in Figure 5.6b.

The second of these tables, Tasks Scheduled, lists all work orders that have been scheduled for the patient and are pending completion. When the user selects an item in this list, the selected item will expand and show the date and time of when the order is scheduled to take place. This is illustrated in Figure 5.6c.

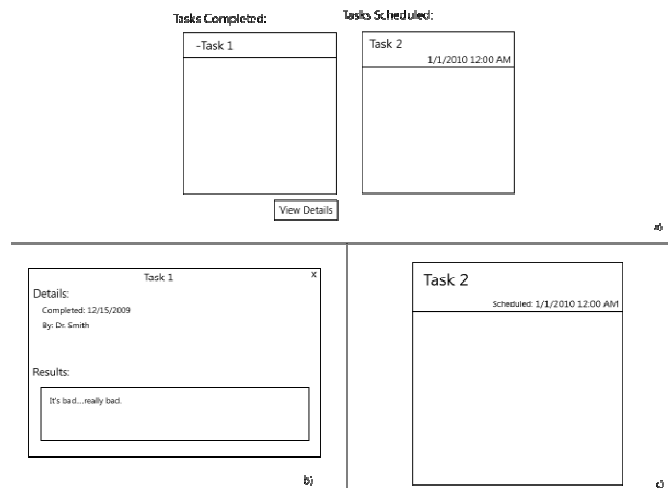


Figure 5.6a, b, & c

In the upper right corner of the screen is a table, illustrated in Figure 5.7, indicating all, if any, medications to which the patient is currently prescribed. This list alerts the user to any combination of medications to which the patient has been prescribed that have known side effects or adverse reactions when taken in conjunction with one other by highlighting the conflicting drugs in yellow with a “!” caution marker. Selecting a medication in the list will enable the “Detail View” button below the list. Clicking the button will display a window that contains all details on the medication, including common side effects, alternatives(generic brands), etc.

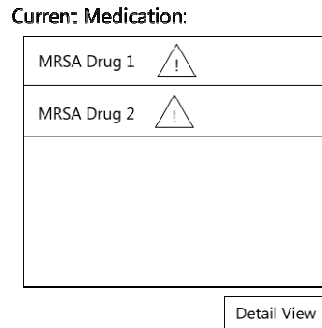


Figure 5.7

Below the medication table is an optional section for displaying any relevant clinical trials associated with the diagnosis, shown in Figure 5.8. Selecting a trial and clicking the “Open” button navigates the user to the selected trial's information. If there are no applicable trials, this section will be grayed out and disabled.

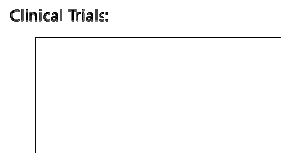


Figure 5.8

The final section of the Treatment screen in the bottom right corner of the screen, shown in Figure 5.9, consists of a grouping of “Action” buttons. These buttons allow the user to customize the treatment process displayed on the screen. The user can click “Add

New Task,” to schedule an additional task to be completed on the patient. The user can also click on the “Prescribe Medication” button to display the prescription dialog. If the user adds either a new work order or prescription, the corresponding tables will be updated to match.

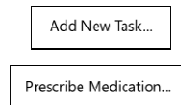


Figure 5.9

## 6 References

- [1] "EMR Systems: Separating the Wheat from the Chaff." *Future Healthcare*. 2008. Analysts In Media, Web. 20 Nov 2009. <<http://www.futurehealthcareus.com/?mc=emr-systems&page=hit-viewresearch>>.
- [2] "MRSA." *Google Health*. 2009. Google, Web. 15 Nov 2009. <<https://health.google.com/health/ref/MRSA>>.
- [3] "PubMed." *The diffusion of decision support systems in healthcare: are we there yet?*. Aug 2009. Web. 27 Nov 2009. <[http://www.ncbi.nlm.nih.gov/pubmed/11067416?itool=EntrezSystem2.PEntrez.Pubmed.Pubmed\\_ResultsPanel.Pubmed\\_RVDocSum&ordinalpos=1](http://www.ncbi.nlm.nih.gov/pubmed/11067416?itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum&ordinalpos=1)>.

## 7 Point of Contact

For further information regarding this document and project, please contact **Prof. Betty H.C. Cheng** at Michigan State University ([chengb@cse.msu.edu](mailto:chengb@cse.msu.edu)). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.