Chris Creech & Nathaniel Murphy

CSC 340 Software Engineering and litercy

FINAL REPORT

Doctor-Patient Request System

3/22/19

Table of Contents

[I. Introduction 3](#_Toc531901701)

[A. Problem Statement 3](#_Toc531901702)

[B. Proposal 3](#_Toc531901703)

[II. System Description 3](#_Toc531901704)

[III. System Requirements 5](#_Toc531901705)

[A. Functional Requirements 5](#_Toc531901706)

[B. Non-functional Requirements 9](#_Toc531901707)

[IV. Use Case Diagram 10](#_Toc531901708)

[V. Class Diagram 11](#_Toc531901709)

[VI. Sequence Diagram(s) 12](#_Toc531901710)

[A. Patient 12](#_Toc531901711)

[B. Doctor 17](#_Toc531901712)

[VII. State Diagram 24](#_Toc531901713)

[VIII. Activity Diagram 25](#_Toc531901714)

[IX. Database Design 26](#_Toc531901715)

[A. ER Diagram 26](#_Toc531901716)

[B. Table Schema 27](#_Toc531901717)

[X. Conclusion 28](#_Toc531901718)

[XI. Data Dictionary 28](#_Toc531901719)

Table of Figures

[Figure 1 - Use-Case Diagram for Doctor-Patient System 10](#_Toc531901720)

[Figure 2 – Class Diagram for Doctor-Patient System. 11](#_Toc531901721)

[Figure 3 - Sequence Diagram for Doctor-Patient System. 12](#_Toc531901722)

[Figure 4 - Sequence Diagram for Doctor-Patient System. 13](#_Toc531901723)

[Figure 5 - Sequence Diagram for Doctor-Patient System. 14](#_Toc531901724)

[Figure 6 - Sequence Diagram for Doctor-Patient System. 15](#_Toc531901725)

[Figure 7 - Sequence Diagram for Doctor-Patient System. 16](#_Toc531901726)

[Figure 8 – Sequence Diagram for Doctor-Patient System. 17](#_Toc531901727)

[Figure 9 – Sequence Diagram for Doctor-Patient System. 18](#_Toc531901728)

[Figure 10 – Sequence Diagram for Doctor-Patient System. 19](#_Toc531901729)

[Figure 11 – Sequence Diagram for Doctor-Patient System. 20](#_Toc531901730)

[Figure 12 – Sequence Diagram for Doctor-Patient System. 21](#_Toc531901731)

[Figure 13 – Sequence Diagram for Doctor-Patient System. 22](#_Toc531901732)

[Figure 14 – Sequence Diagram for the Doctor-Patient System. 23](#_Toc531901733)

# Introduction

## Problem Statement

Dr. Bogus and his office has asked for a system to be developed, that will improve the convenience and efficiency of their daily work schedule. With doctors’ offices becoming more technologically advanced, the implementation of a system that would allow for the prescribing of new patient prescriptions, granting prescription refill permits to patients, receiving appointment requests from patients, and receiving phone call requests from patients to become paperless, would set Dr. Bogus and his office apart from all other doctor’s offices.

## Proposal

Our team has been tasked with designing a system that will mitigate frustration between patients and doctors, through the implementation of a Doctor-Patient Service System. This system will streamline efficiency for patients, by allowing patients to: check medical records, grant/deny medical record access to a requesting doctor, request an appointment with a specific doctor, request a phone call with a specific doctor, and request a refill permit for an expired/previous prescription; all through the Patient side of the Doctor-Patient Service System. Through the dual-implementation of the Doctor side, of the Doctor-Patient Service System, doctors will have the ability to: check patient medical records, update patient medical records, request/grant access to a patient’s medical records (via patient’s family doctor), schedule patient-requested appointments, grant/deny patient-requested prescription refill permits, and write new prescriptions.

With the implementation of this system, the ability for patients and doctors to communicate in a peer-to-peer manner, would eliminate the mundane process of contacting the doctor’s receptionist and communicating with the doctor via a third-party individual. This system would allow both patient and doctor alike to reach out and personally communicate with a family-doctor, new doctor or even a new patient.

# System Description

This system allows for the peer-to-peer communication between patients and doctors. The Doctor-Patient Service System performs a total of 12 business events, which are grouped based on whether the current user of the system is logged in as a patient or logged in as a doctor. There are 5 patient business events and 7 doctor business events.

The first set of business events that are able to be executed via the Doctor-Patient Service System are the patient business events, which allow for a patient to: check medical records, grant/deny medical record access to a requesting doctor, request an appointment with a specific doctor, request a phone call with a specific doctor, and request a refill permit for an expired/previous prescription.

**In order for a patient to request an appointment with a specific doctor:** The patient shall press a button to request an appointment, the system shall then display a drop-down of doctors and a calendar (displaying available appointment dates, and drop-down list that displays times for currently selected calendar date), the patient shall select a doctor and an available date/time and then press a button to submit the appointment request, the system shall then notify the corresponding doctor.

**In order for a patient to check their own medical records:** The patient shall press a button to check their medical records, the system shall then display the patient’s medical records (includes previous prescriptions and previous appointments), the patient shall then press a button to exit their medical records.

**In order for a patient to grant/deny medical record access to a requesting doctor:** The patient shall press a button to view his/her medical records. If there is a doctor other than the patient’s family doctor requesting access to his/her records, the system shall display a notification. The patient shall then press a button to view a list of doctors that are requesting access to his/her medical record. The patient shall click on a specific doctor that is requesting access to his/her records. The system will then display two buttons for the patient to interact with (GRANT or DENY). The patient will then have the opportunity to GRANT or DENY the selected doctor access to his/her medical record. Given that the patient wishes to revoke medical record access from a specific doctor, the patient shall simply click on a doctor that has been GRANTED access. Once selected, the system shall display a new button (REVOKE), the patient shall simply select the REVOKE button in order to revoke access from that specific doctor.

**In order for a patient to request a phone call with a specific doctor:** The patient shall press a button to request a phone call from their doctor.The system shall display a list of doctors that the patient can select from.The patient shall select the specific doctor that they wish to receive a phone call from, and then press a button to submit the doctor phone call request. The system shall then send a notification to the intended doctor, notifying them of a new phone call request.

**In order for a patient to request a refill permit for an expired/previous prescription:** The patient shall press a button to request a refill permit for a previous/expired prescription.The system shall display a list of the patient’s prescription history.The patient shall select the desired prescription they want filled, along with the last time that prescription was filled, and then press a button to submit the request.The system will then send a notification to the specified doctor notifying them of a new refill request.The patient must now wait until the doctor has either granted/denied the refill request. Once the doctor has made his/her decision, the system shall notify the patient of the status of the refill request.

The second set of business events that are able to be executed via the Doctor-Patient Service System are the doctor business events, which allow for a doctor to: check patient medical records, update patient medical records, request/grant access to a patient’s medical records (via patient’s family doctor), schedule patient-requested appointments, grant/deny patient-requested prescription refill permits, and write new prescriptions.

**In order for a doctor to check patient medical records:** The doctor shall press a button to view patient medical records. The system shall display a list of that doctor’s patients or patients that doctor has been granted access to viewing.The doctor shall select a patient from the list that he/she wishes to view and press a button to view the selected patient’s medical record.The system shall display the medical record of the selected patient.

**In order for a doctor to update patient medical records:** The doctor shall press a button to view their patients’ medical records.The system shall display a list of patients that they are the family doctor of, or that they have been granted access to viewing.The doctor shall select a patient they wish to edit.The system shall display that patient’s profile.The doctor shall press a button to edit that patient’s information.The doctor shall press a button to save the changes made to the patient profile.The system shall update that information.

**In order for a doctor to request access to a patient’s medical records (via patient’s family doctor):** The doctor shall press a button to view their patients’ medical records.The system shall display a list of patients that they are the family doctor of, or that they have previously been granted access to viewing.The doctor shall press a button to add new patients.The system shall display a new window that allows for the doctor to search for a specific patient.The doctor shall enter all necessary patient search information.The system shall display the patients name.The doctor shall select the patient and press a button to request access to that patient’s medical records.

**In order for a doctor to grant/deny medical record access to a requesting doctor:** The doctor shall press a button to view his/her patients’ medical records.The system shall display a list of patients that they are the family doctor of, or that they have previously been granted access to viewing.If there are any Record Requests, the system shall display a notification.The doctor shall press the notification button to trigger the system to display a list of the Doctors who are requesting access to their patient(s) medical records.The doctor shall then select a specific request from the list and press a button to view the details of that request. The system shall then display the full details of the selected request. Once the doctor has read and acknowledged the request, that same doctor has the option to either grant or deny the requesting doctor access to a patient(s) medical record. The system shall notify the requesting doctor once their request has been GRANTED or DENIED.

**In order for a doctor to schedule patient-requested appointments:** The doctor shall press a button to view their scheduled patient appointments.The system shall display a new window with that doctor’s appointment schedule.The system shall also display a notification when a new appointment request has been submitted.The doctor shall press the notification button.The system shall display a list of new appointment requests, so the doctor is able to select appointments one at a time and approve or deny them by pressing a button.

**In order for a doctor to grant/deny patient-requested prescription refill permits:** The system shall display a notification when a refill request has been submitted by a patient.The doctor shall press the notification button in order to view a list of refill requests.The doctor shall select a specific refill request to view the request details, including:patient information and refill request reason.The doctor shall press a button to GRANT or DENY the refill request.If the request is GRANTED, the system shall notify the intended patient’s pharmacy to fulfill the specific refill request.

**In order for a doctor to write new prescriptions:** The doctor shall press a button to view a list of his/her patients.The system shall display a list of patients that they are the family doctor of.The doctor shall select a patient and press a button to create a new prescription for that patient.The system shall display a form for the new prescription.The doctor shall fill in all necessary information for the prescription, and then press a button to submit the new prescription.The system shall save the prescription to that patient’s medical record and then send a notification to that patients preferred pharmacy, notifying them of the newly submitted prescription.

# System Requirements

## Functional Requirements

**Patient Use Cases**

**R1. The system shall allow for a patient to request a doctor appointment.**

* 1. The patient shall press a button to request an appointment.
  2. The system shall display a calendar, displaying available dates, and a drop-down list that displays times for the currently selected calendar date.
  3. The patient shall select an open time and press a button to submit the appointment request.
  4. The system shall send a notification to the doctor.
     1. The doctor shall accept or deny the appointment request.

**R2. The system shall allow a patient to check their own medical records**

* 1. The patient shall press a button to check their medical records.
  2. The system shall display the medical records of the patient, including:
     1. Previous prescriptions
     2. Previous appointments
  3. The patient shall press a button to exit his/her medical records.

**R3. The system shall allow for a patient to grant or deny a doctor, other than his/her family doctor, access to their personal medical records.**

* 1. The patient shall press a button to view his/her medical records.
     1. If there is a doctor, other than the patient’s family doctor, requesting access to his/her records, the system shall display a notification.
  2. The patient shall press a button to determine who may access their medical records.
  3. The system shall display a new window with a list of individual doctors who have been:
     1. GRANTED access to that patient’s medical records

OR have…

* + 1. REQUESTED access to that patient’s medical records.
  1. The patient shall click on a specific doctor that is requesting access to his/her records.
  2. The system shall display two buttons for the patient to interact with.
  3. The patient shall press the “GRANT” or “DENY” button, in order to grant or deny a requesting doctor access to his/her medical records.
     1. The patient may also revoke the access of a granted doctor.

**R4. The system shall allow for a patient to request a phone call from their doctor.**

* 1. The patient shall press a button to request a phone call from their doctor.
  2. The system shall display a list of doctors that the patient can select from.
  3. The patient shall select the specific doctor that they wish to receive a phone call from.
  4. The patient shall press a button to submit the doctor phone call request.
     1. If the urgent box is checked when the patient submits the request, a message will be displayed prompting the patient to call 9-1-1 if there is an emergency.
  5. The system shall notify the intended doctor that a patient is in need of a phone call.

**R5. The system shall allow for a patient to request a prescription refill permit.**

* 1. The patient shall press a button to request a refill permit for a specific medication.
  2. The system shall display a list of the patient’s previous prescription history.
  3. The patient shall select the desired prescription they want filled, along with the last time that prescription was filled, and then press a button to submit the request.
  4. The system shall notify the specified doctor of the newly submitted refill permit request.
     1. The patient shall wait until the intended doctor has approved/denied their refill permit request.
  5. The system shall notify the patient of the doctor’s decision about the refill permit request.

**Doctor Use Cases**

**R1. The system shall allow for a doctor to schedule patient appointments.**

* 1. The doctor shall press a button to view their scheduled patient appointments.
  2. The system shall display a new window with that doctor’s appointment schedule.
     1. The system shall also display a notification when a new appointment request has been submitted.
  3. The doctor shall press the notification button.
  4. The system shall display a list of new appointment requests.
  5. The doctor shall select appointments, one at a time, and approve or deny them by pressing a button.

**R2. The system shall allow for a doctor to check the medical records of his/her patient(s).**

* 1. The doctor shall press a button to view patient medical records.
  2. The system shall display a list of that doctor’s patients, or patients that, that doctor has been granted access to viewing.
  3. The doctor shall select a patient from the list, that he/she wishes to view, and press a button to view the selected patient’s medical record.
  4. The system shall display the medical record of the selected patient.
  5. The doctor shall close the medical records window or press a button to close the medical records window.

**R3. The system shall allow for a doctor to request access to a patient’s medical record, via that patient’s family doctor.**

* 1. The doctor shall press a button to view their patients’ medical records.
  2. The system shall display a list of patients that they are the family doctor of, or that they have been granted access to viewing.
  3. The doctor shall press a button to add new patients.
  4. The system shall display a new window that allows for the doctor to search for a specific patient.
  5. The doctor shall enter all necessary patient information.
  6. The system shall display the patients name.
  7. The doctor shall select the patient and press a button to request access to that patient’s medical records.

**R4. The system shall allow for a Doctor to grant/deny medical record access, of a specific patient, to a requesting doctor.**

* 1. The doctor shall press a button to view their patient’s medical records.
  2. The system shall display a list of patients that they are the family doctor of, or that they have been granted access to viewing.
     1. If there are any Record Requests, the system shall display a notification.
  3. The doctor shall press the notification button.
  4. The system shall display a list of Doctor’s who are requesting access to their patient’s medical records.
  5. The Doctor shall select a specific request and press a button to view the details of the request.
  6. The system shall display the record request.
  7. The doctor shall press a button to GRANT or DENY that doctor access to a specific patient’s medical records.
     1. Given that the Doctor GRANTs the other Doctor access, the system shall send a notification to the requesting doctor.

**R5. The system shall allow for a doctor to grant/deny a patient-prescription refill permit, upon patient request.**

* 1. The system shall display a notification when a refill request has been submitted by a patient.
  2. The doctor shall press the notification button.
  3. The system shall display a list of refill requests.
  4. The doctor shall select a specific refill request to view the request details, including:
     1. Patient information
     2. Refill request reason.
  5. The system shall allow for the doctor to GRANT or DENY the refill request.
  6. The doctor shall press a button to GRANT or DENY the request.
     1. If the request is GRANTED, the system shall notify the intended patient pharmacy to fulfill the specific refill request.

**R6. The system shall allow a doctor to update patient medical records.**

* 1. The doctor shall press a button to view their patients’ medical records.
  2. The system shall display a list of patients that they are the family doctor of, or that they have been granted access to viewing.
  3. The doctor shall select a patient they wish to edit.
  4. The system shall display that patient’s profile.
  5. The doctor shall press a button to edit that patient’s information.
  6. The doctor shall press a button to save the changes made to the patient profile.
  7. The system shall update that information.
  8. The doctor shall close the medical records window.

**R7. The system shall allow for a doctor to write a new patient prescription.**

* 1. The doctor shall press a button to view a list of his/her patients.
  2. The system shall display a list of patients that they are the family doctor of.
  3. The doctor shall select a patient and press a button to create a new prescription for that patient.
  4. The system shall display a form for the new prescription.
  5. The doctor shall fill in all necessary information for the prescription, and then press a button to submit the new prescription.
  6. The system shall save the prescription to that patient’s medical record and then send a notification to that patients preferred pharmacy, notifying them of the newly submitted prescription.

## Non-functional Requirements

**NR1.** The pharmacy service system shall be able to connect to the doctor-patient service system.

# Use Case Diagram



Figure 1 - Use-Case Diagram for Doctor-Patient System

This diagram explains what the Doctor-Patient System does at a very low level. The box is essentially the Doctor-Patient System. The ovals are the twelve functional requirements, or business events, that the system must be able to perform. The stick figures are known as actors, and they are the individuals that will be interacting with the Doctor-Patient System. The red lines indicate use-association, between the actors and the system, for each of the actors. In the Doctor-Patient System, a Doctor will be permitted the use of seven system functions, while a Patient will only be permitted the use of five system functions.

# Class Diagram



Figure 2 – Class Diagram for Doctor-Patient System.

This diagram is an overview of the system’s insides and their relationships with one another. The boxes resemble classes and the red lines indicate some sort of interaction. The top-level of the class-box is the class’ name. The middle-level of the class-box stores attributes, or variables, that will be used within that class. The type of variable being stored is indicated after the attribute name. The bottom-level of the class-box stores function calls of the class. The numbers or stars that are displayed with each red line indicates the multiplicity of each interaction, between associated classes.

# Sequence Diagram(s)

## Patient



Figure 3 - Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 4 - Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 5 - Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 6 - Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 7 - Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.

## Doctor



Figure 8 – Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 9 – Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 10 – Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 11 – Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 12 – Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 13 – Sequence Diagram for Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.



Figure 14 – Sequence Diagram for the Doctor-Patient System.

This diagram shows elements, in single use case, as they interact over time and are organized according to object (horizontally) and time (vertically). The horizontal axis shows the elements that are involved in the interaction. The arrows indicate message or data flow amongst the elements. The vertical axis represents time proceedings down the page.

# State Diagram

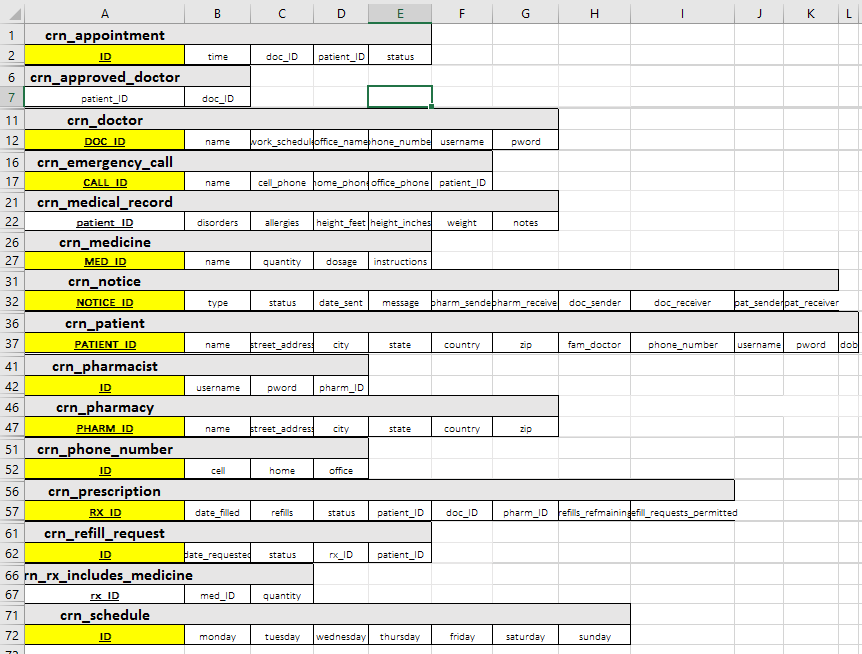


# Activity Diagram

# Database Design

## ER Diagram

## Table Schema



# Conclusion

In conclusion, the proposed Doctor-Patient Service System will help to eliminate the inefficiency of communication between doctor and patient. Through this system’s ability to integrate with the Pharmacy Service System, the elimination of handwriting, which accounts for the majority of pharmaceutical errors, will be one of many conveniences supported by the Doctor-Patient Service System. This system will completely revolutionize the way in which patients and doctors communicate and will allow for a more personal connection between patient and doctor. The Doctor-Patient Service System fulfills a total of 12 different business events: 5 patient business events and 7 doctor business events. The 5 patient business events offered by the Doctor-Patient Service System would allow for a patient to: request an appointment with a specified doctor, check personal medical records, grant/deny medical record viewing access to a requesting doctor, request a phone call from a specified doctor, and request a prescription refill permit for a previous or expired prescription. The 7 doctor business events offered by the Doctor-Patient Service System would allow for a doctor to: schedule patient appointments, check medical records of patients, grant/deny medical record access to a requesting doctor, request access to a patient’s medical record, write new prescriptions, grant/deny patient prescription refill requests, and update patient medical record information. The automation of this system will minimize the time it takes for a patient to contact or communicate with a specific doctor (vice versa) while also increasing the margins of served patients by allowing for the continuous and uninterrupted communication between patient and doctor.

# Data Dictionary

Activity Diagram – (insert definition)

Case – An instance of a particular situation, similar to a notice.

Domain Diagram – Conceptual model of the system domain, that incorporates both behavior and data flow between elements.

Dr. Bogus – The fake doctor being referenced in this report.

Functional requirements – While a function is simply described as a specification of behavior between outputs and inputs, a functional requirement defines the functionality of a system or its components.

Notice – To become aware of, notified or warned of something; especially when preparations need to be made.

Phony Pharmacy – The fake pharmacy being referenced in this report.

Prescription – An instruction written by a doctor that authorizes a patient to be provided a medicine or treatment.

Process – A series of actions or steps taken in order to achieve a particular end.

Refill – An allowance of medicine prescribed to a patient, by a doctor, and filled by a pharmacy for a doctor-allotted number of times.

Service – An act of assistance or business activity.

Sequence Diagram – an interaction diagram that details how operations are carried out -- what messages are sent and when.

State Diagram – a diagram that shows the possible states of an object and the transitions that cause a change in state.

System – A group of related hardware units or programs (or both), especially when dedicated to a single application.

Use Case Diagram – A graphic representation of the interactions amongst the elements of a system.

User – A person who uses or interacts with the system.