Software Requirements Specification

for

<Hospital management system>

Version 1.0 approved

Prepared by <Mustafa Labib>

<10-25-2025>

1. Introduction

The following section provides an overview of the Software Requirements Specification (SRS) for the Hospital Management System (HMS).

1.1 Purpose

The purpose of this SRS is to outline both the functional and non-functional requirements of the Hospital Management System. This document provides an overall description of the system, including UML analysis models, to facilitate an understanding of the software's requirements.

1.2 Scope

The Hospital Management System is a software application designed to help hospital staff manage patient records, doctor schedules, appointment details, and billing information. It assists hospital personnel in efficiently managing patient data, scheduling appointments, tracking medical history, and handling billing operations.

1.3 Competitors

Our main competitor, the *Ibn Al-Haitham System* for booking clinics at Mansoura University, has some limitations that our solution aims to overcome. By addressing areas such as limited user interface, appointment flexibility and scalability, we aim to provide a more comprehensive and user-friendly experience. These strengths are further detailed in our unique selling points.

1.4 Unique Selling Points

1. **Intuitive User Interface (UI):**
   Our system offers a modern, user-friendly interface designed for smooth navigation and ease of use. Unlike the limited and outdated UI in the Ibn Al-Haitham System, our solution provides a visually appealing, streamlined experience, allowing users to access features quickly and intuitively.

2. **Flexible Appointment Scheduling:**
   We understand the need for flexibility in booking appointments. Our system allows users to make appointments on shorter notice, accommodating various schedules and last-minute needs. This feature reduces the inconvenience of booking too far in advance, providing more immediate access to clinic services.

3. **Efficient Check-in Process:**
   To eliminate the long waiting times at check-in, our system automates and accelerates the check-in process, ensuring minimal wait times upon arrival. Patients can check in quickly, allowing them to focus on their healthcare needs rather than lengthy administrative procedures.

4. **Expanded Accessibility Beyond University Boundaries:**
   While the Ibn Al-Haitham System was exclusive to Mansoura University students with a university email, our solution welcomes a broader audience. Patients from outside the university community can access the clinic's services, fostering a more inclusive and accessible healthcare environment for all.

1.5 Actors in the HMS

Admin	**Manages users, access, and system settings.**
Doctor	Handles patient records, appointments, and prescriptions.
patient	Reviews prescriptions, assigns medications.
Nurse	Assists doctors, maintains records.
Pharmacist	Reviews prescriptions, assigns medications.
Receptionist	Manages appointment scheduling, patient check-ins.
Software User	Responsible for maintaining, updating, and overseeing the system, as well as checking for errors.

---

2. Overall Description

2.1 Product Perspective

The software described in this SRS is the software for a Hospital Management System. It integrates various hardware and software elements to streamline hospital operations and improve patient care quality. The system is designed to enhance administrative tasks and facilitate efficient communication between departments.

2.2 Product Functions

The Hospital Management System provides the following key functions:

- Patient Management: System users can add, edit, and store patient data, including personal details, medical history, and contact information.
- Doctor Management: System users can manage doctor information, including specialties, schedules, and availability.

- Appointment Scheduling: The system allows users to book, reschedule, and cancel patient appointments with doctors.
- Billing and Payment Processing: The system generates bills, processes payments, and keeps track of transaction records.
- Medical Records Management: Patient medical records, diagnoses, prescriptions, and treatment histories are saved in the system for easy access by authorized personnel.

2.3 User Characteristics

- System User: Can add, edit, and delete records for patients, doctors, active substance, nurses, receptionists, pharmacists, prescription, and appointments. Also responsible for managing billing information, medical records, and hospital inventory.
- System Admin: Responsible for system maintenance, updates, and error checking. They ensure data integrity and system security and oversee user access and permissions.
- Doctor: Primarily interacts with patient data related to medical history, treatment plans, and schedules. They are given read access to patient records and limited write access for updating treatment information.
- Receptionist: Manages front-desk operations, including patient check-ins, appointment scheduling.

2.4 Constraints

- Internet Connection: The system relies on a stable internet connection to function optimally. Network outages or slow internet speeds can affect performance.
- Database Capacity: The database should have sufficient storage capacity to manage the growing volume of patient records, doctor schedules, and billing information. Regular maintenance and optimization of the database are necessary.

2.5 Assumptions and Dependencies

- The software will primarily be used on desktop computers and laptops; thus, all users should have basic computer literacy and a working knowledge of English.
- System users are expected to have medical or administrative backgrounds to understand and utilize the system's medical terminology and functions efficiently.

- The system depends on reliable hardware infrastructure within the hospital, including servers, computers, and backup solutions.

3. System Requirement Specification

3.3 Functional Requirements

This section contains the functional requirements of the Hospital Management System (HMS). Below are tables listing the functional requirements.

3.3.1 Manage Doctors [MD]

Requirement ID	Requirement Description
1-MD	The system must allow the admin to "add" doctor information.
2-MD	The system must allow the admin to "update" selected doctor information.
3-MD	The system must allow the admin to "remove" selected doctor information.

Table 1: Functional Requirements for Managing Doctors

3.3.2 Manage Patients [MP]

Requirement ID	Requirement Description
I-MP	The system must allow the admin or receptionist to "add" patient information.
2-MP	The system must allow the admin or receptionist to "update" selected patient information.
3-MP	The system must allow the admin to "remove" selected patient information.

Table 2: Functional Requirements for Managing Patients

3.3.3 Manage Appointments [MA]

Requirement ID	Requirement Description
I-MA	The system must allow the receptionist to "schedule" appointments with doctors.
2-MA	The system must allow the receptionist to "update" selected appointments.
3-MA	The system must allow the receptionist to "cancel" appointments.

Table 3: Functional Requirements for Managing Appointments

3.3.4 Manage Billing and Payments [MBP]

Requirement ID	Requirement Description
I-MBP	The system must allow the billing department to "generate" patient bills based on
2-MBP	appointments and treatments.
3-MBP	The system must allow the billing department to "process" payments from patients.

Table 4: Functional Requirements for Managing Billing and Payments

3.3.5 Manage Prescription [PA]

Requirement ID	Requirement Description
1-PA	The system must allow the doctor to "Create" Prescription.
2- PA	The system must allow the Pharmacist to "update" Prescription.

Table 5: Functional Requirements for Managing Prescriptions

3.3.6 Manage Pharmacist [MP]

Requirement ID	Requirement Description
1- MP	The system must allow the admin to "add" Pharmacist information.
2- MP	The system must allow the admin to "update" selected Pharmacist information.
3- MP	The system must allow the admin to "remove" selected Pharmacist information.

Table 6: Functional Requirements for Managing Pharmacists

3.3.7 Manage Receptionist [MR]

Requirement ID	Requirement Description
1- MR	The system must allow the admin to "add" Receptionist information.
2- MR	The system must allow the admin to "update" selected Receptionist information.
3- MR	The system must allow the admin to "remove" selected Receptionist information.

Table 7: Functional Requirements for Managing Receptionists

3.3.8 Manage Nurses [MN]

Requirement ID	Requirement Description
1- MN	The system must allow the admin to "add" **Nurse** information.
2- MN	The system must allow the admin to "update" selected **Nurse** information.
3- MN	The system must allow the admin to "remove" selected **Nurse** information.

Table 8: Functional Requirements for Managing **Nurses**

3.3.9 Manage Clinic [MC]

Requirement ID	Requirement Description
1- MC	The system must allow the admin to "add" **Clinic** information.
2- MC	**The system must allow the admin to "update" selected** Clinic **information.**
3- MC	The system must allow the admin to "remove" selected **Clinic** information.

Table 9: Functional Requirements for Managing **Clinic**

3.3.10 Manage Medication [MM]

Requirement ID	Requirement Description
1- MM	The system must allow the admin to "add" **Medication** information.
2- MM	**The system must allow the admin to "update" selected** Medication **information.**
3- MM	The system must allow the admin to "remove" selected **Medication** information.

Table 10: Functional Requirements for Managing **Medication**

3.3.11 Manage Active Substance [MAS]

Requirement ID	Requirement Description
1- MAS	The system must allow the admin to "add" **Active Substance** information.
2- MAS	**The system must allow the admin to "update" selected** Active Substance **information.**
3- MAS	The system must allow the admin to "remove" selected **Active Substance** information.

Table 11: Functional Requirements for Managing **Active Substance**

3.2 Non-Functional Requirements

Non-functional requirements specify the criteria for the operation of the HMS rather than specific functionalities.

3.2.1 Performance Requirements

Requirement ID	Requirement Description
I-PR	The system must respond to user actions within 3 seconds for all roles (admin, doctor, receptionist).
2-PR	The system should be compatible with modem web browsers and mobile platforms.
3-PR	The system should display confirmation and error messages within 2 seconds.
4-PR	The system should be stable and reliable for continuous hospital operations.

Table 12: Performance Requirements for HMS

3.2.2 Safety and Security Requirements

The HMS must not harm or damage users' systems. It should provide safe data handling to protect patient confidentiality and sensitive data.

Requirement ID	Requirement Description
I-SSR	The system must implement safe login and logout through session handling.
2-SSR	**Use encryption for secure user login and sensitive data.**
3-SSR	Protect the database from SQI injection and other attacks to prevent data leakage or loss.
4-SSR	**Implement SSL certificates to secure all data transmissions.**

Table 13: Safety and Security Requirements for HMS

3.2.3 Reliability Requirements

The HMS should be modularly designed to allow ease of maintenance and upgrades. Modularity reduces interdependence, ensuring each module performs its specific function.

Requirement ID	Requirement Description
I-SSR	The system should include exception handling to avoid system crashes or unintended results.

Table 12: Reliability Requirements for HMS

3.2.4 Other Software Quality Attributes

3.2.4.1 Usability

- ***The system should have a user-friendly and intuitive interface that hospital staff can use with minimal training.***

3.2.4.2 Availability

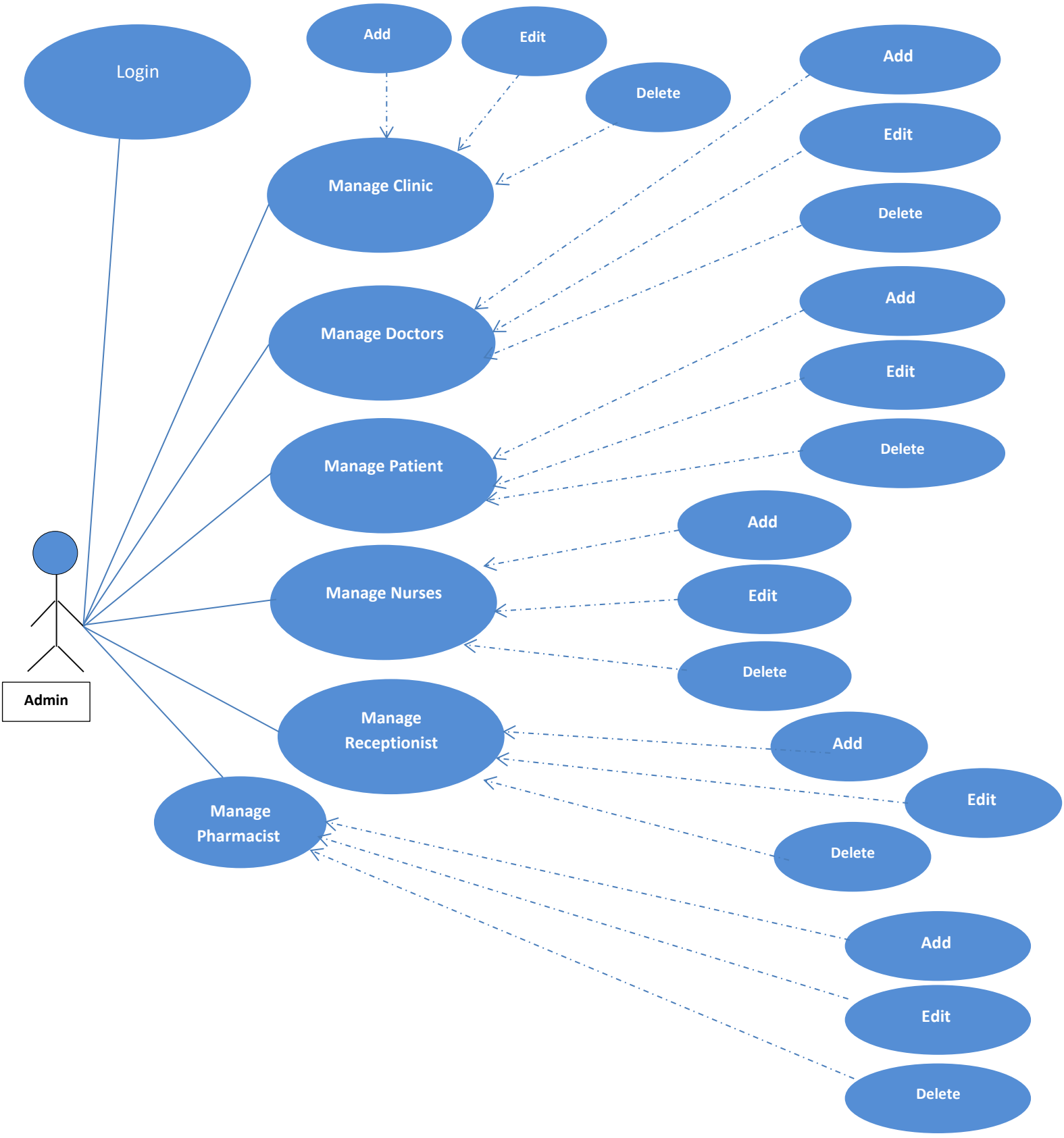
- ***The system must be available 24/7 with a maximum downtime of 20 minutes per day, as healthcare facilities operate continuously.***

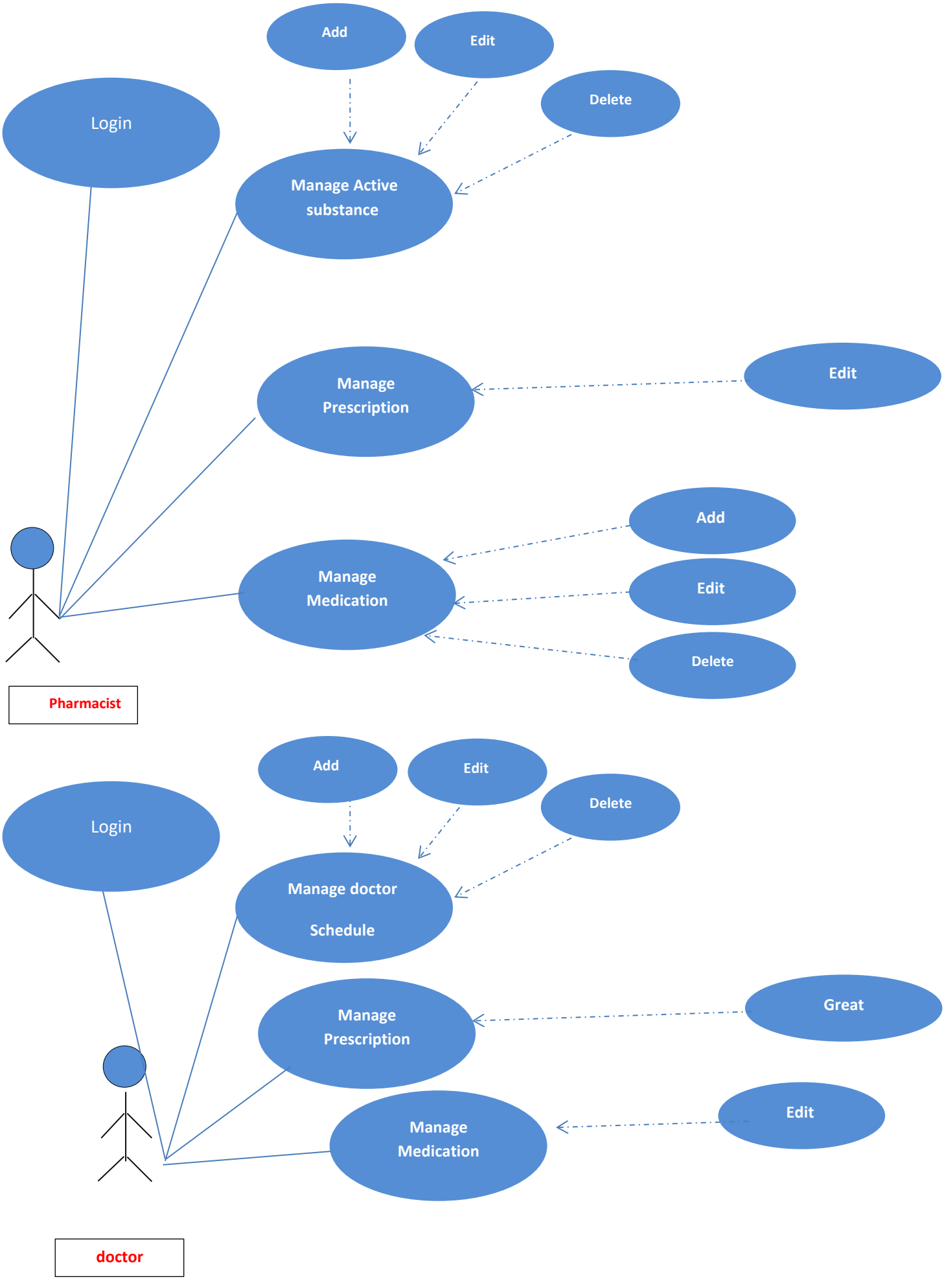
4. System Design

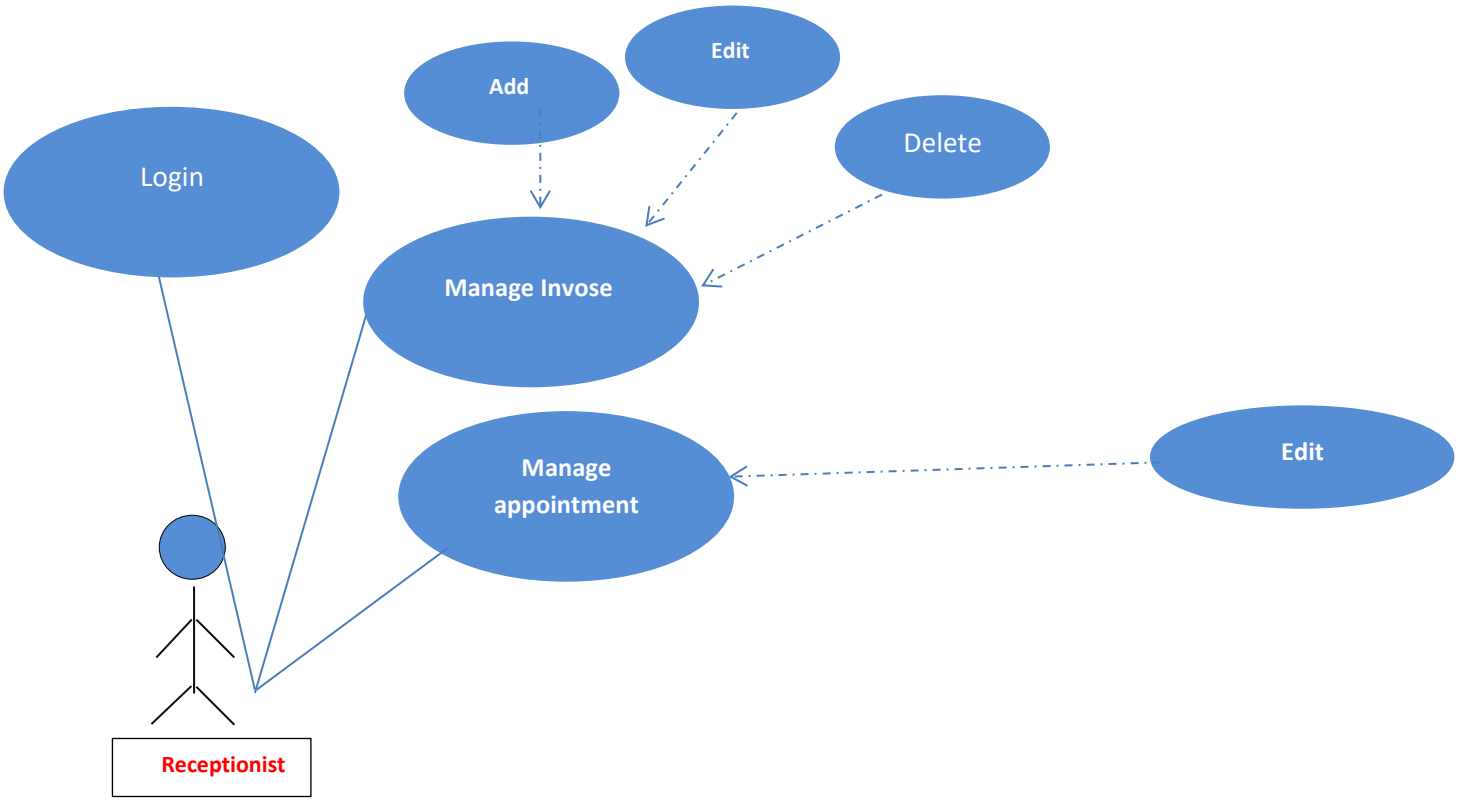
4.1 UML Diagram

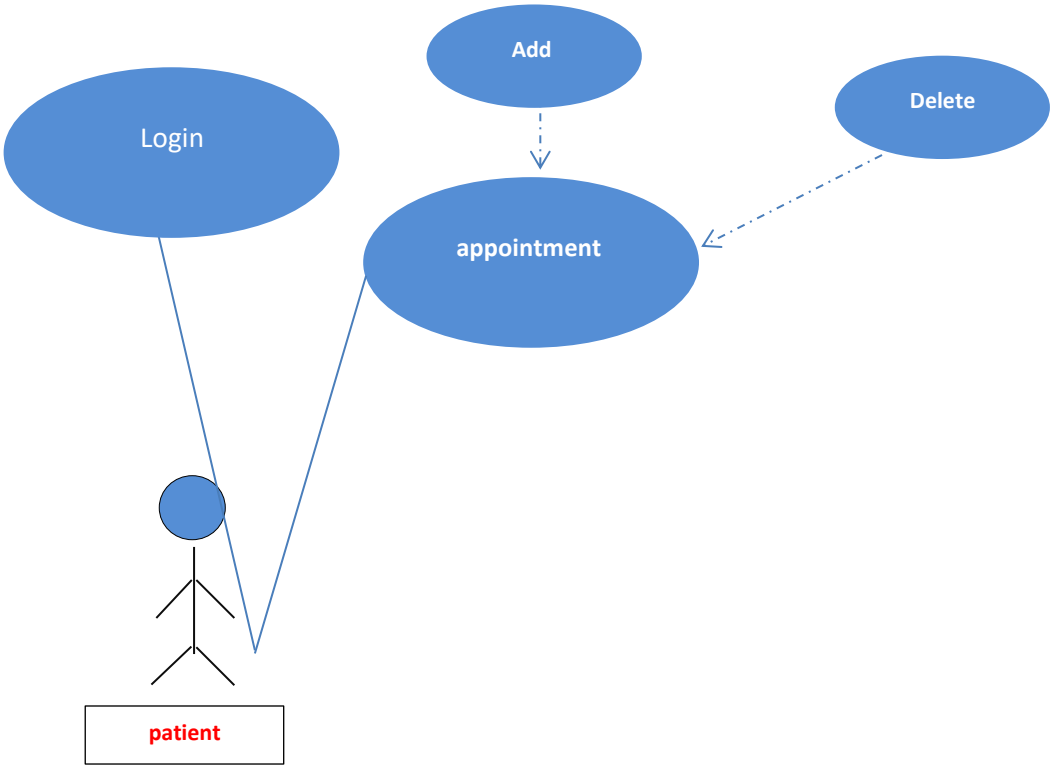
For this project only 5 diagram of UML diagram will be used. They are Use Case, Collaboration, Sequence, Activity, and Entity Relationship Diagram.

4.2 High Level Use Case Diagram of Hospital Management System



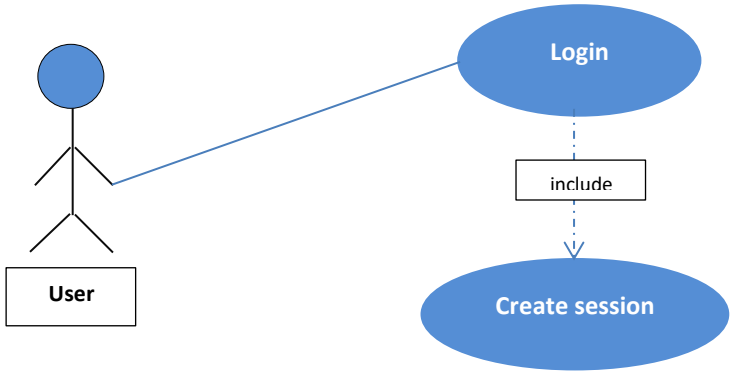

Add
Edit
Delete
Login
Manage Active substance
Manage Prescription
Edit
Add
Manage Medication
Edit
Delete
Pharmacist
Add
Edit
Delete
Login
Manage doctor Schedule
Manage Prescription
Great
Edit
Manage Medication
doctor


Edit
Add
Delete
Login
Manage Invose
Manage
appointment
Edit
Receptionist


Add
Delete
Login
appointment
patient

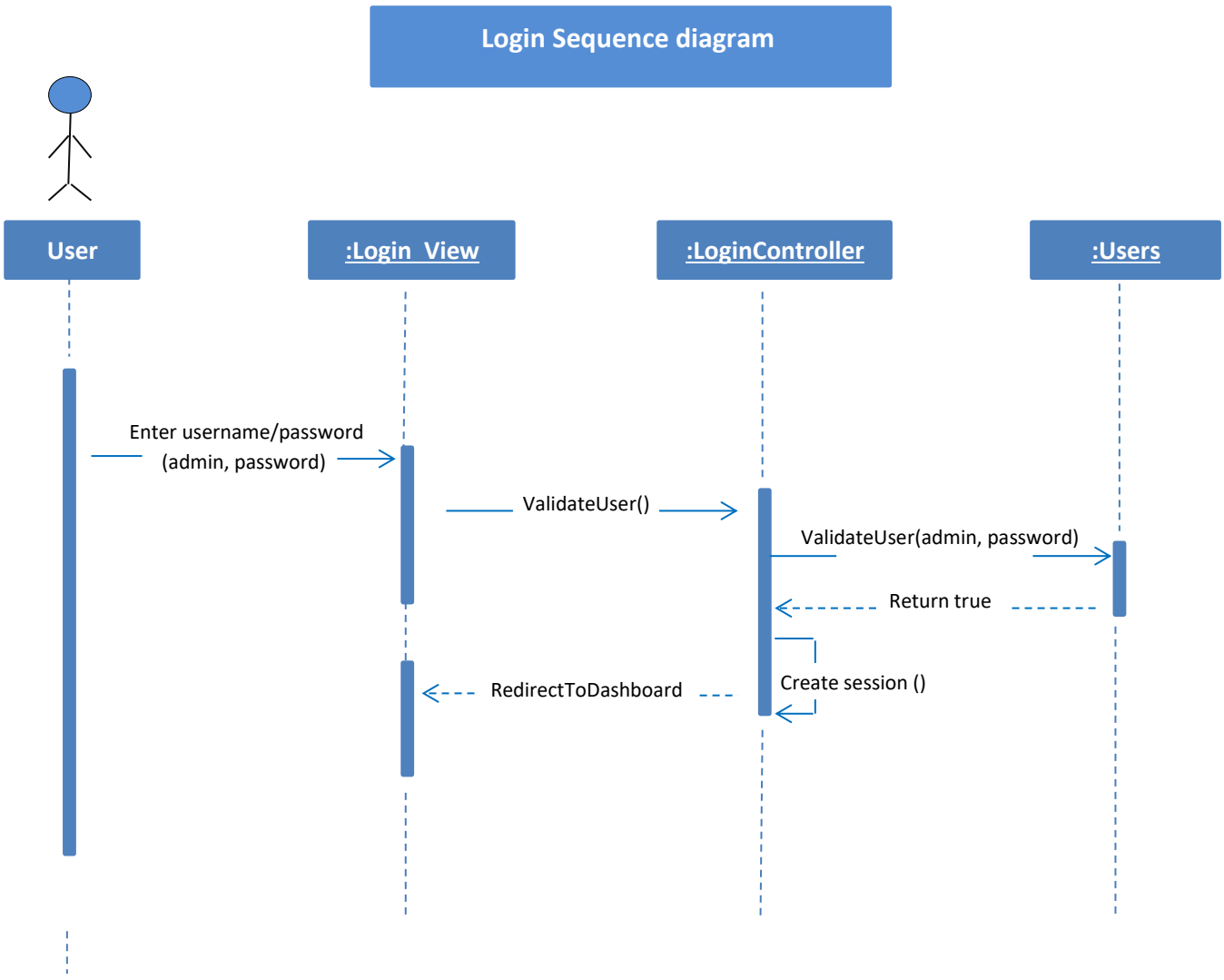
4.3.1 Use Case Diagram of Login

Figure 1 High Level Use Case Diagram of Library Management



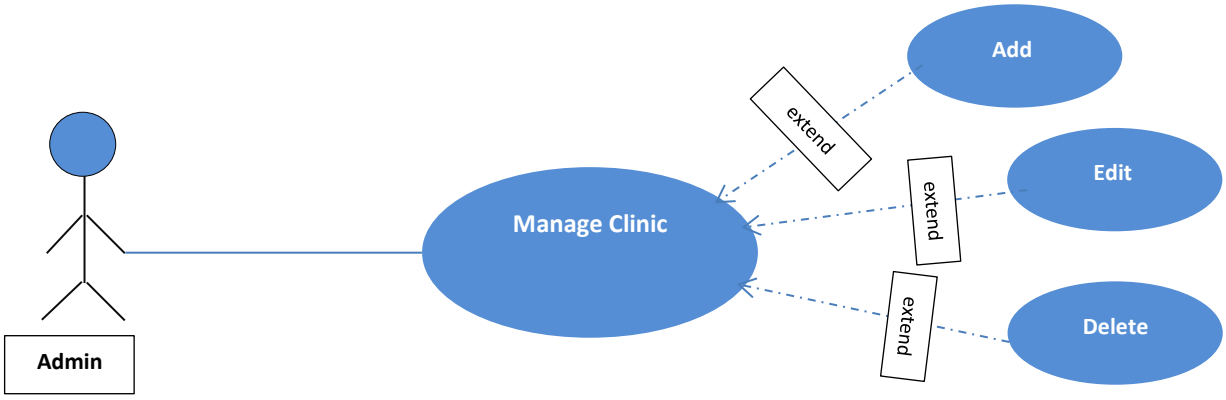
Use Case Diagram of Login Dashboard

4.3.2 Sequence Diagram of Login

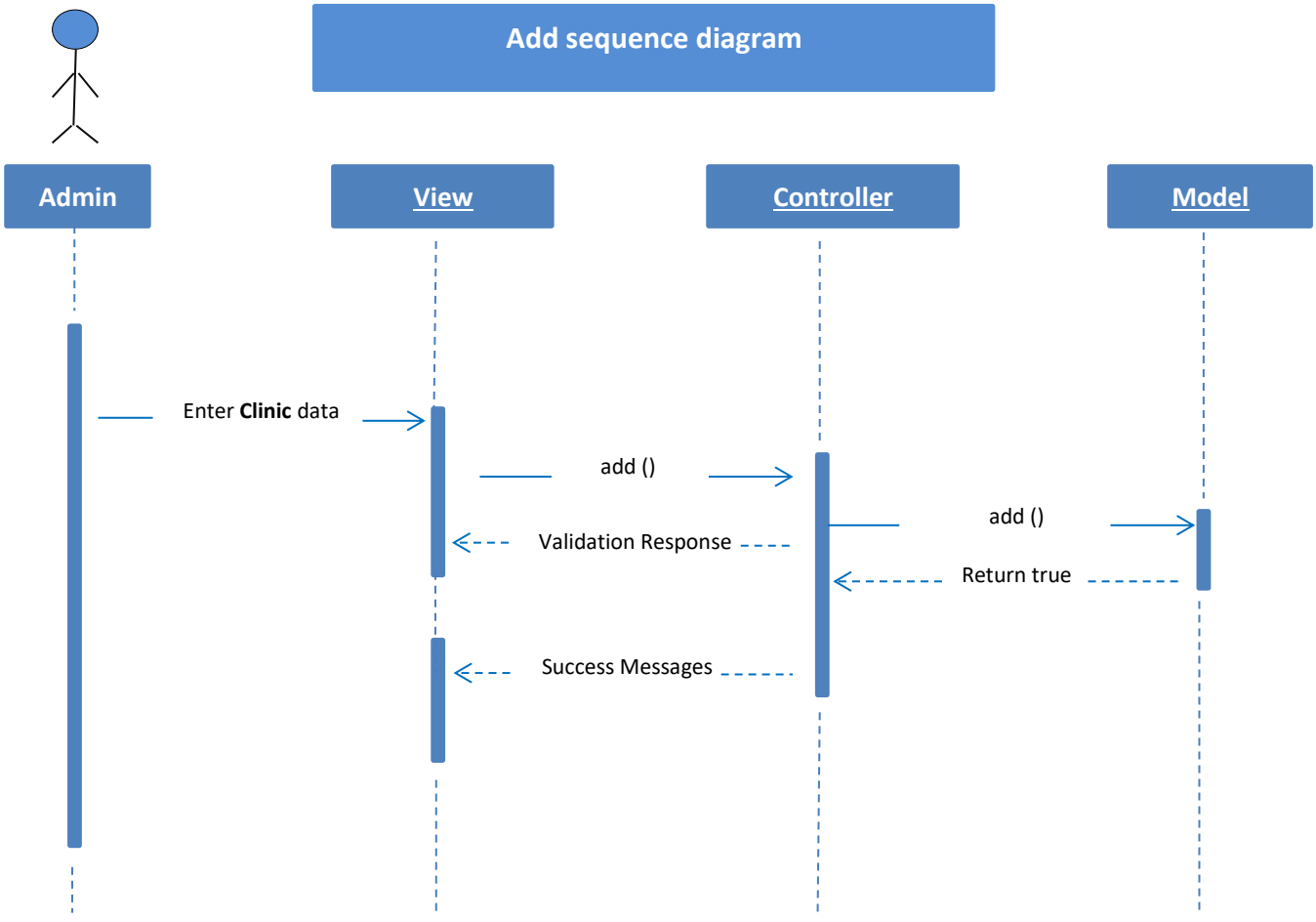


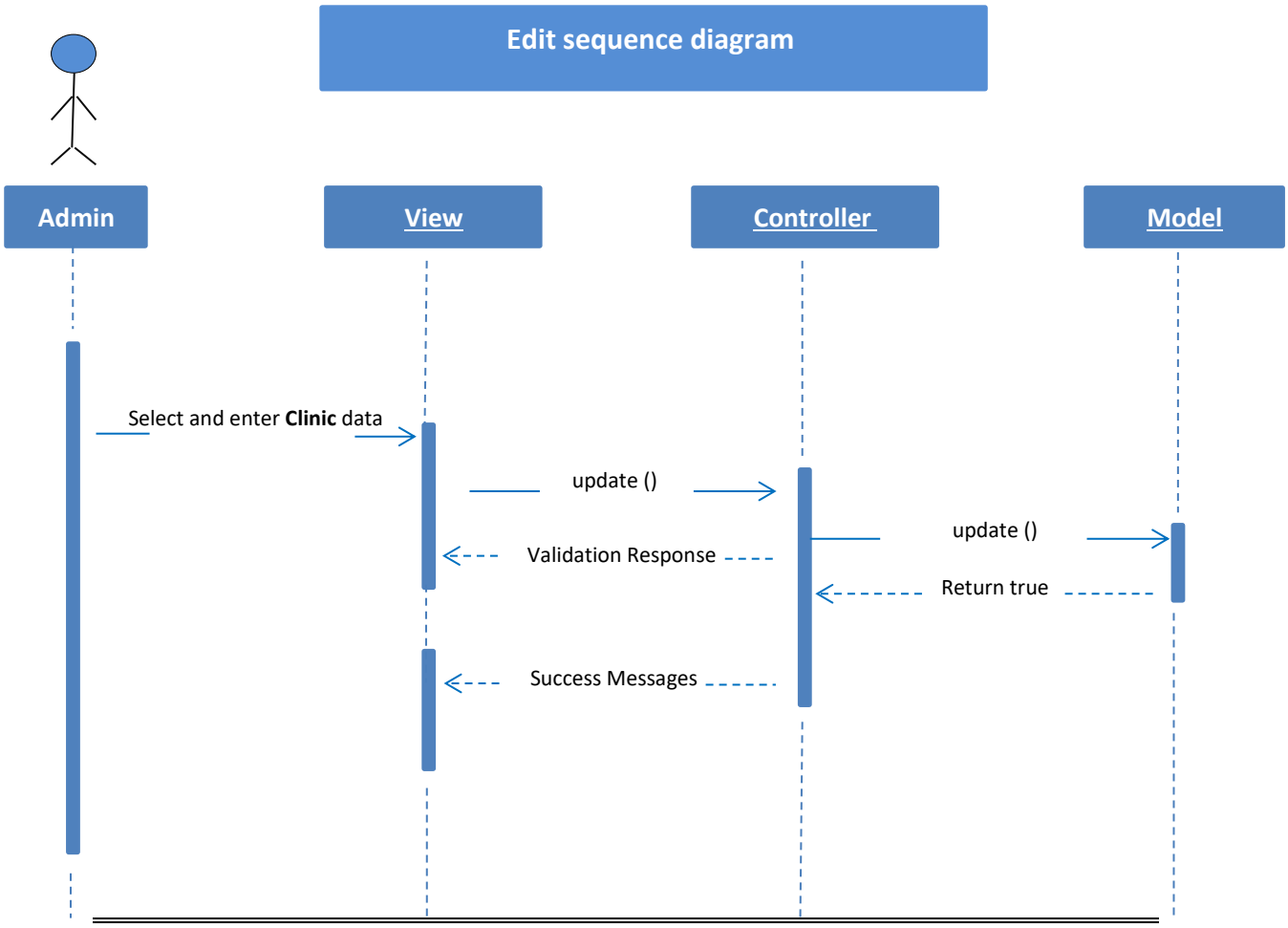
Sequence Diagram of Login

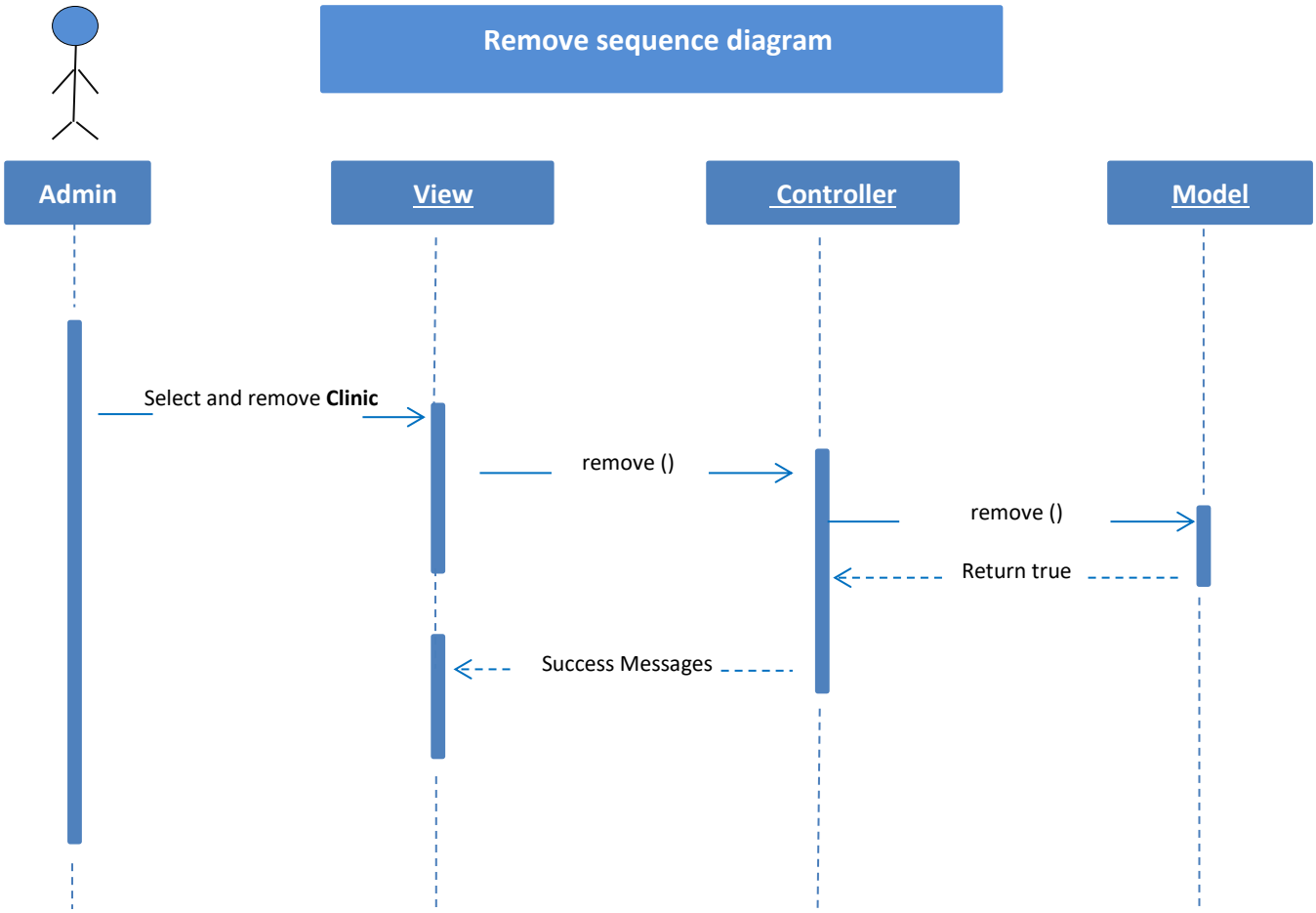
4.4.1 Use Case Diagram of Manage Clinic



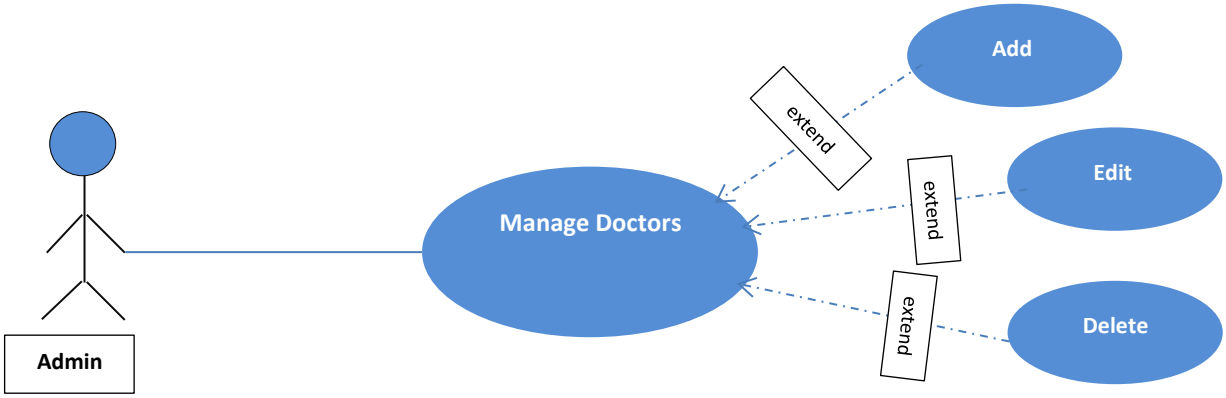
4.4.2 Sequence Diagram of Manage Clinic



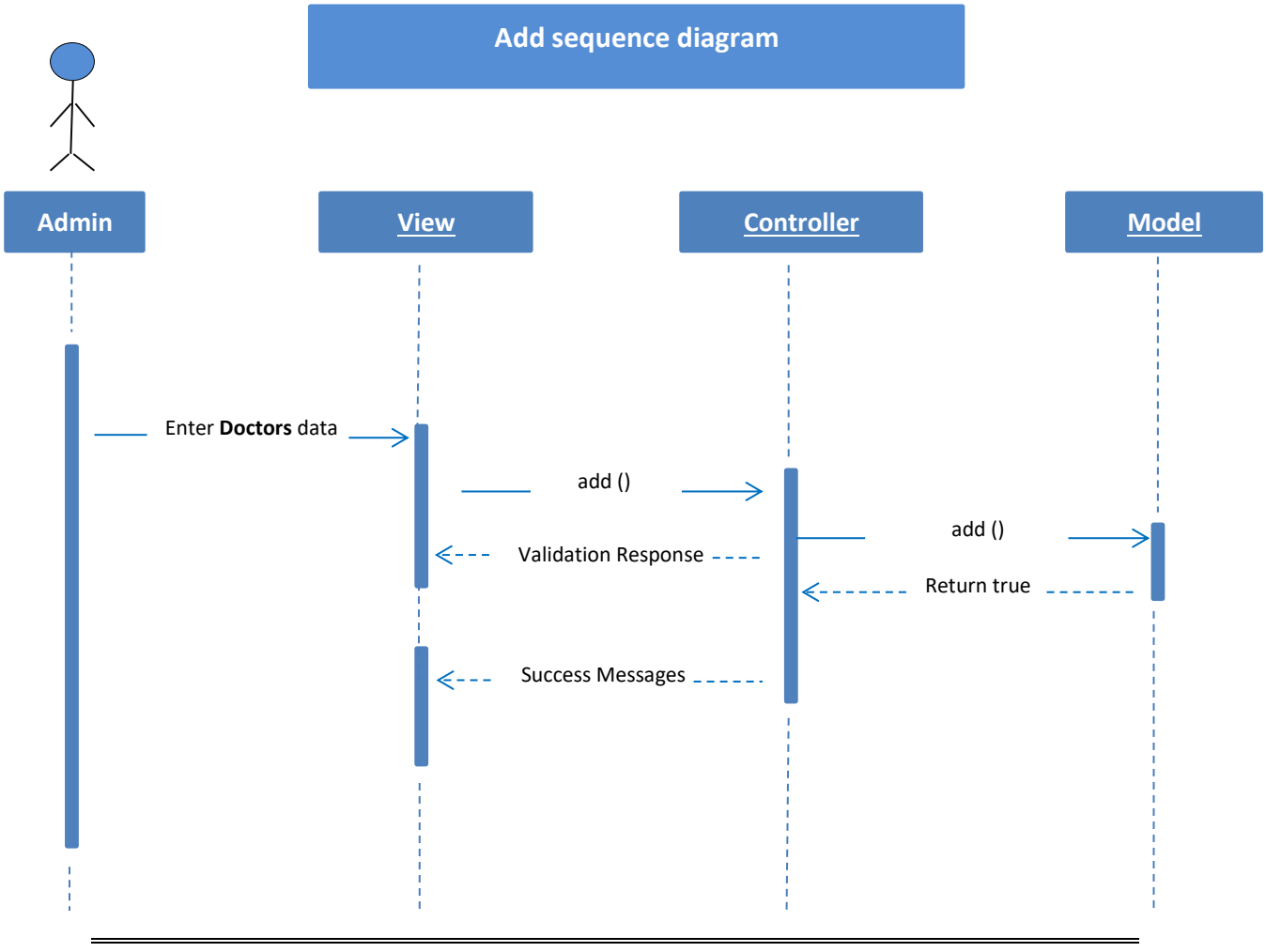

Edit sequence diagram
Admin
View
Controller
Model
Select and enter **Clinic** data
update ()
update ()
Validation Response
Return true
Success Messages

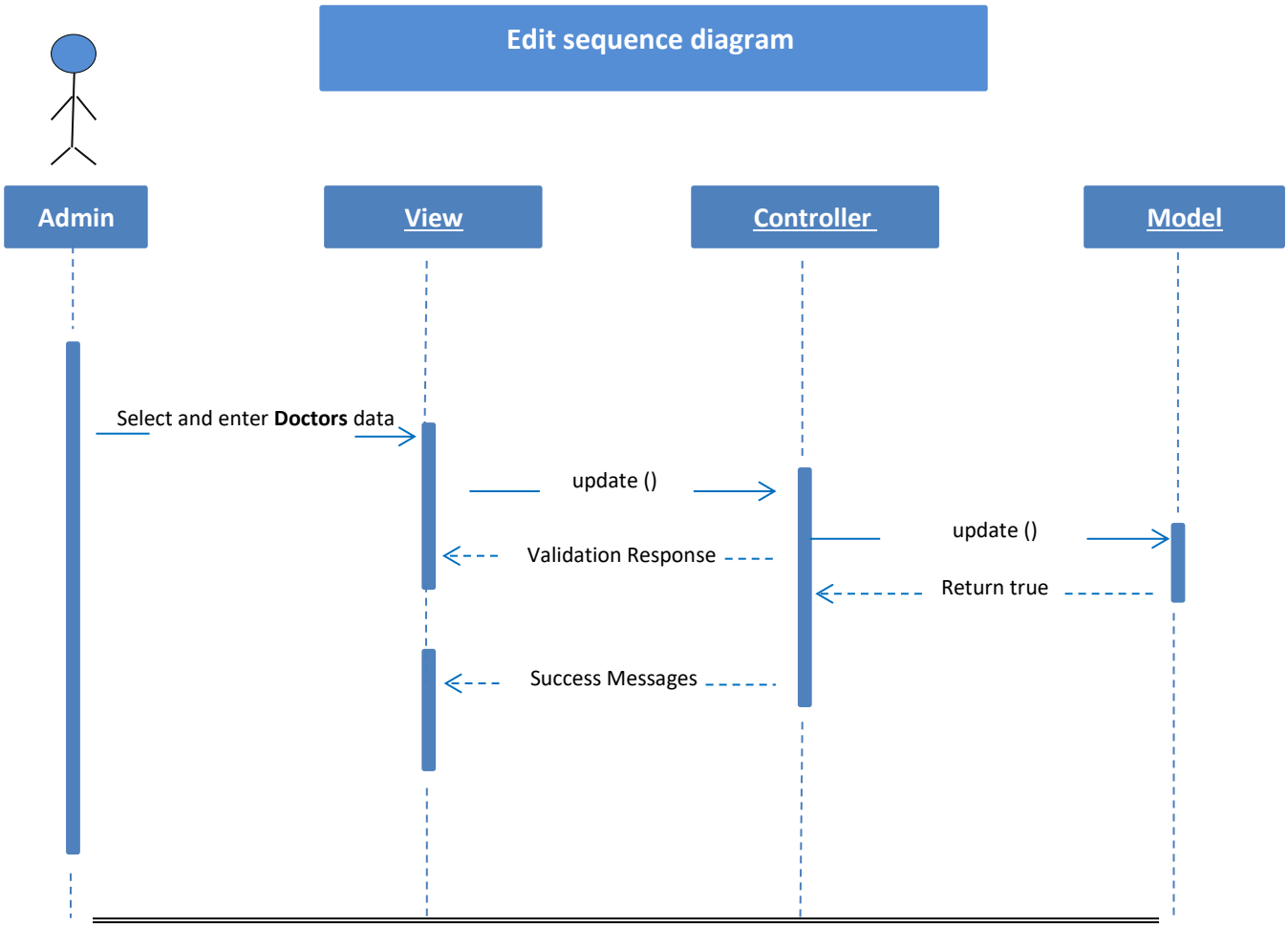

Remove sequence diagram
Admin
View
Controller
Model
Select and remove **Clinic**
remove ()
remove ()
Return true
Success Messages

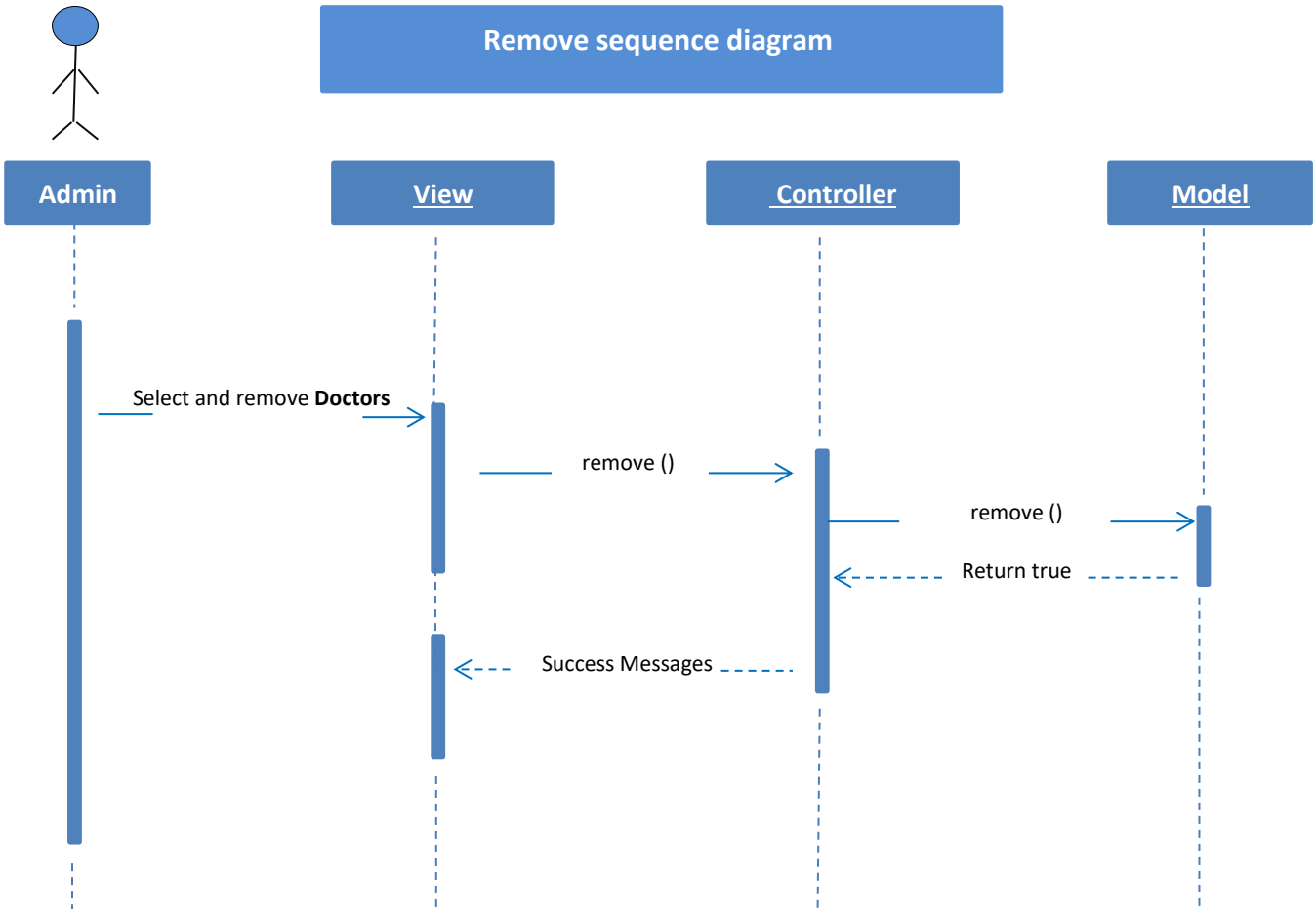
4.5.1 Use Case Diagram of Manage Doctors



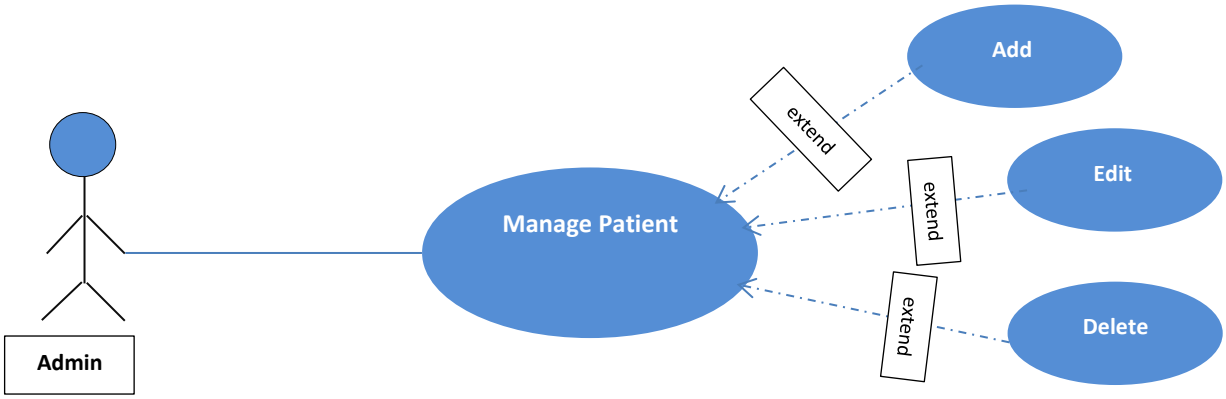
4.5.2 Sequence Diagram of Manage Doctors



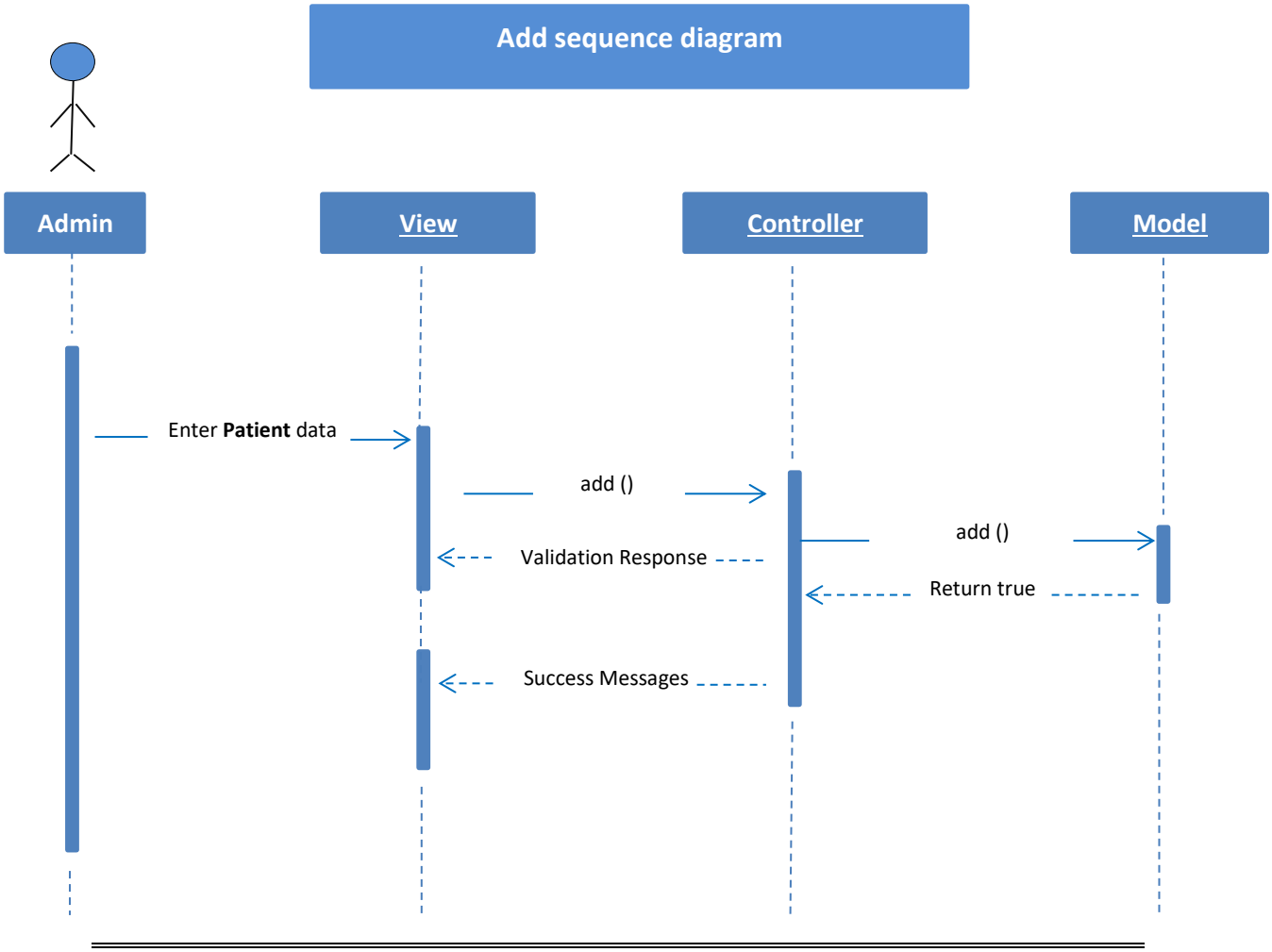

Edit sequence diagram
Admin
View
Controller
Model
Select and enter **Doctors** data
update ()
update ()
Validation Response
Return true
Success Messages

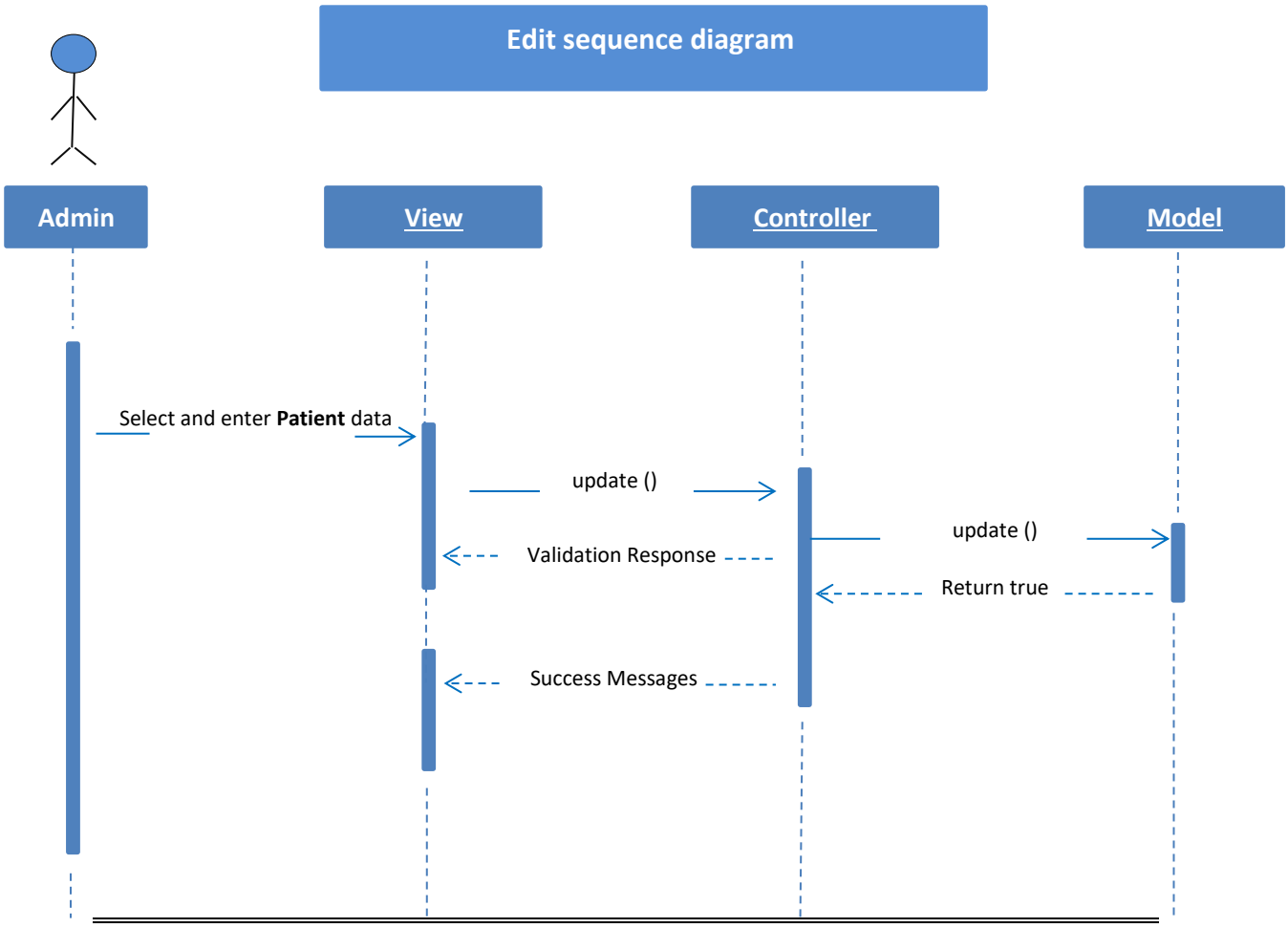

Remove sequence diagram
Admin
View
Controller
Model
Select and remove **Doctors**
remove ()
remove ()
Return true
Success Messages

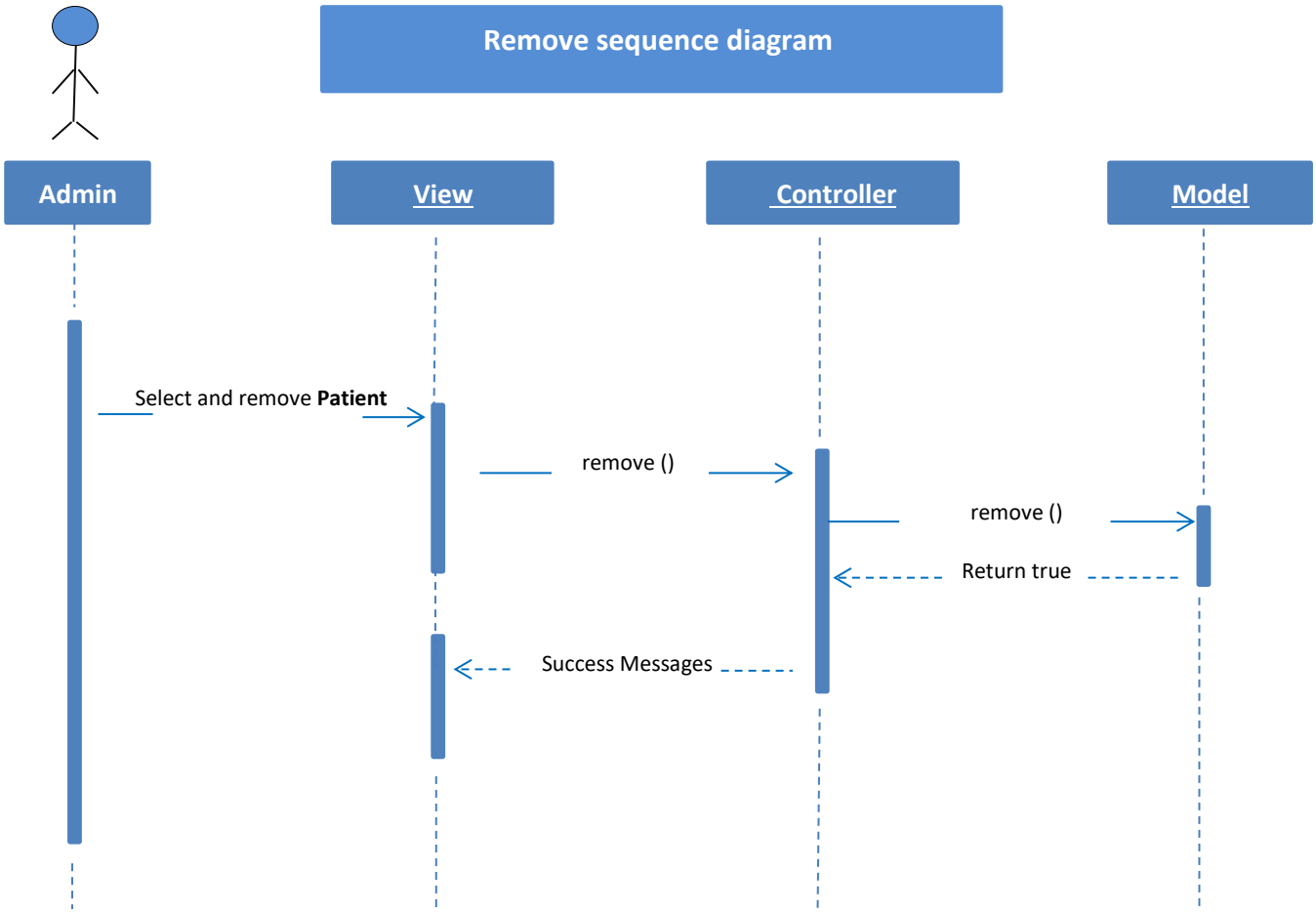
4.6.1 Use Case Diagram of Manage Patient



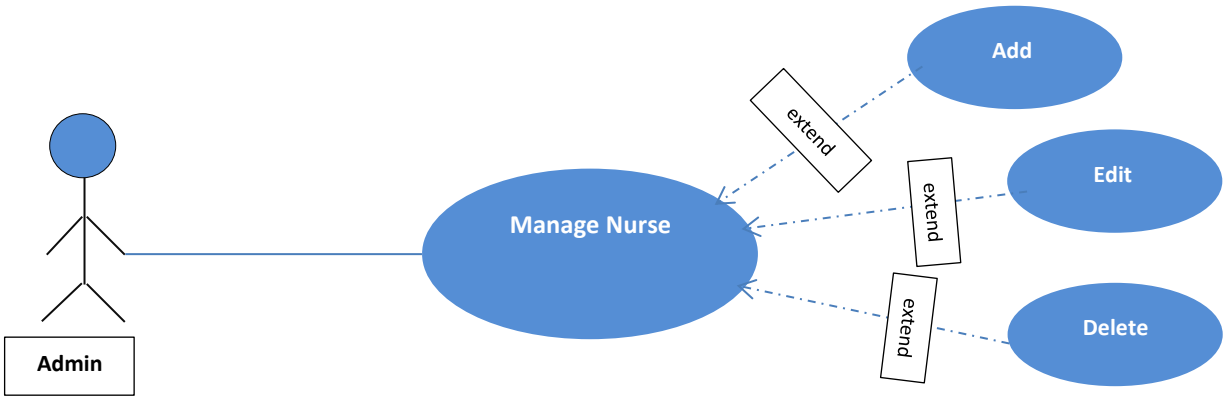
4.6.2 Sequence Diagram of Manage Patient



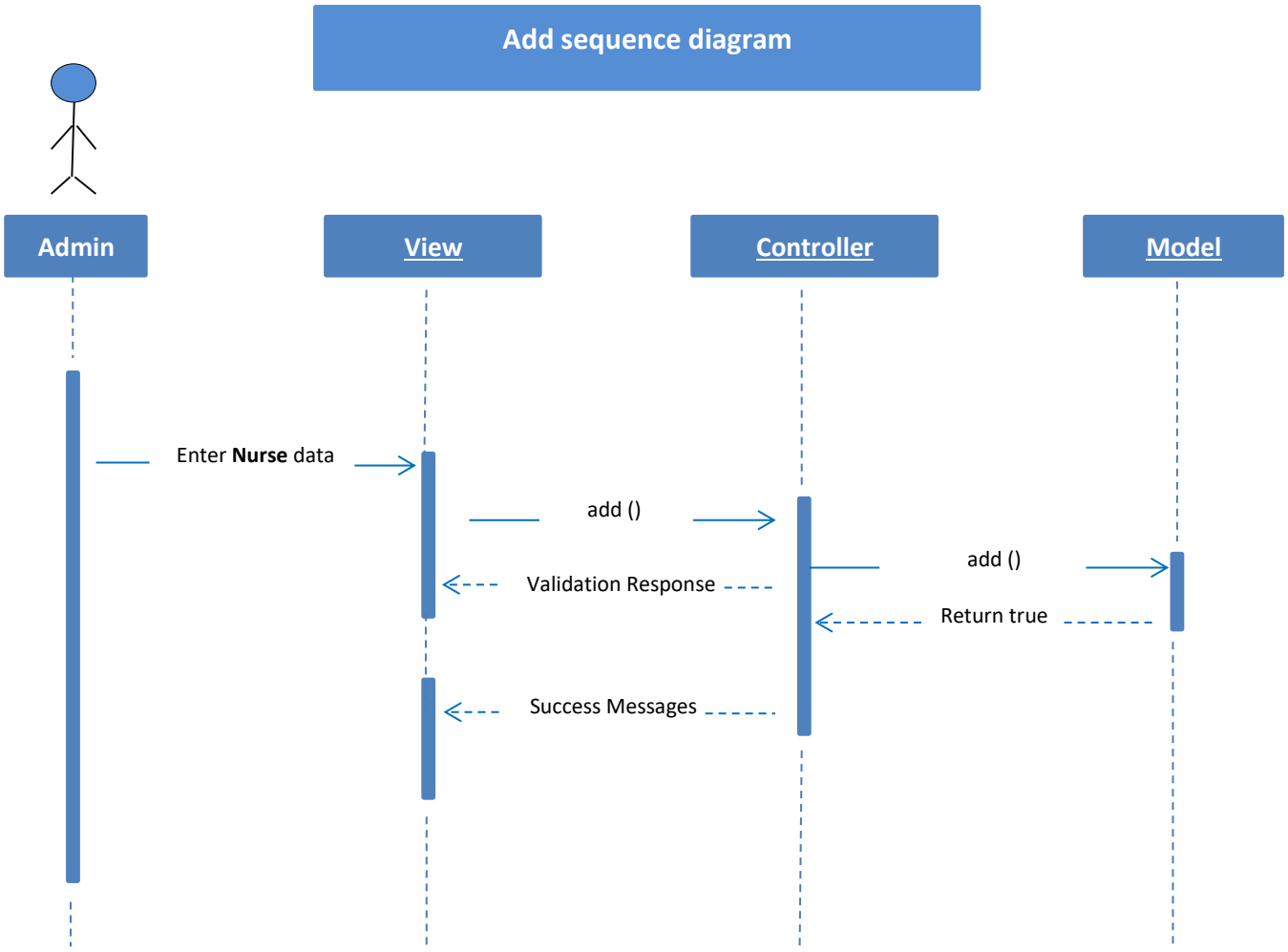

Edit sequence diagram
Admin
View
Controller
Model
Select and enter **Patient** data
update ()
update ()
Validation Response
Return true
Success Messages

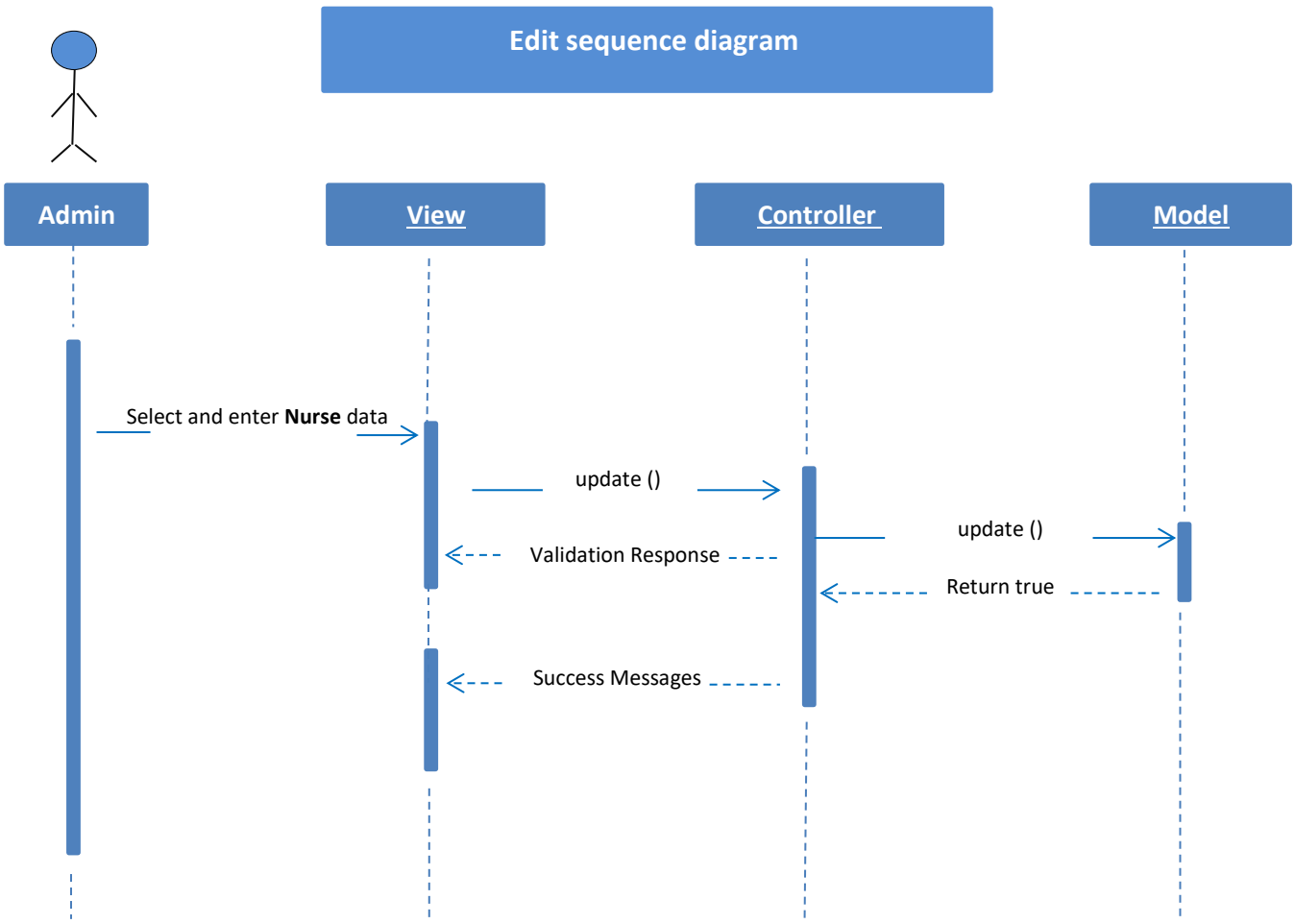

Remove sequence diagram
Admin
View
Controller
Model
Select and remove **Patient**
remove ()
remove ()
Return true
Success Messages

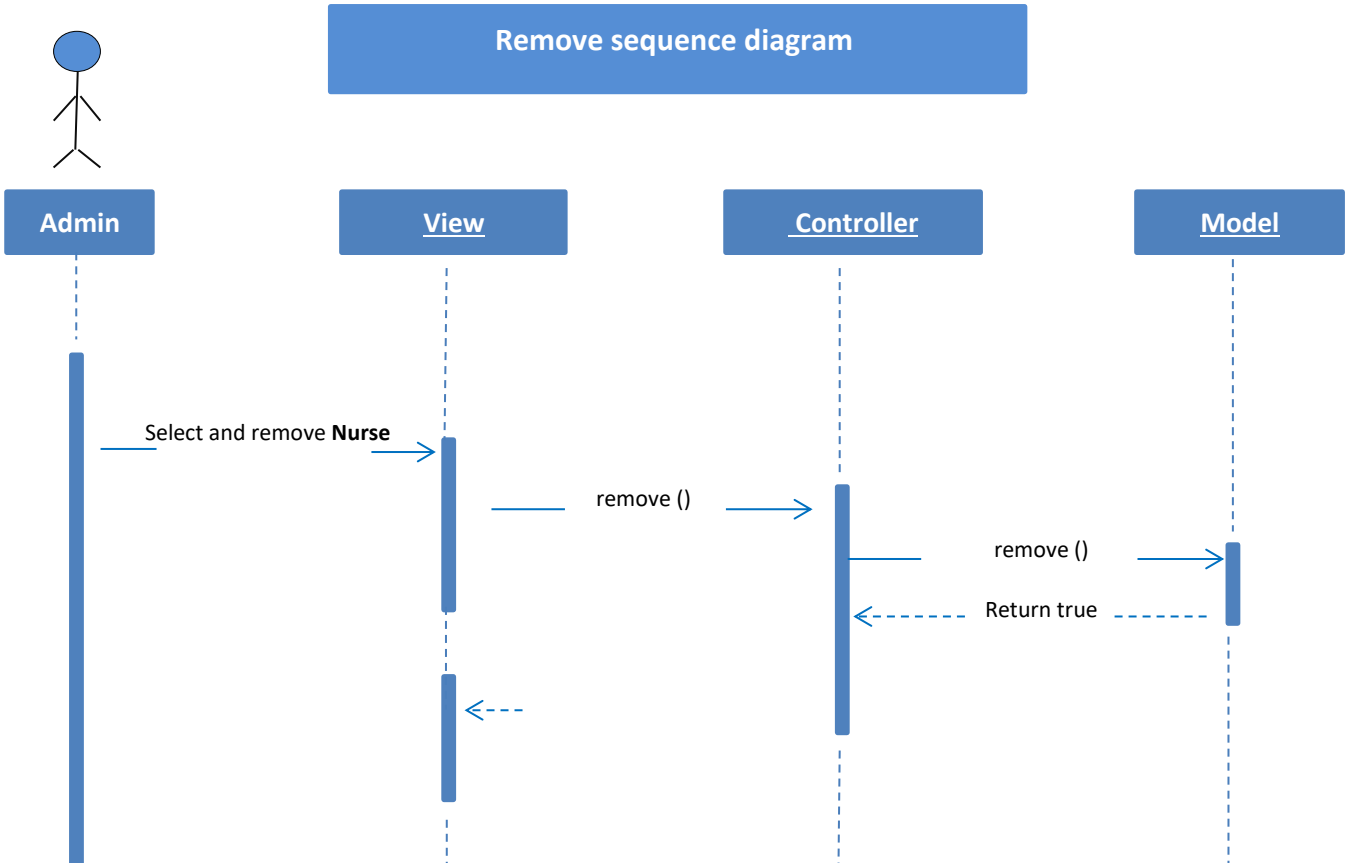
4.6.1 Use Case Diagram of Manage Nurse



4.6.2 Sequence Diagram of Manage Nurse

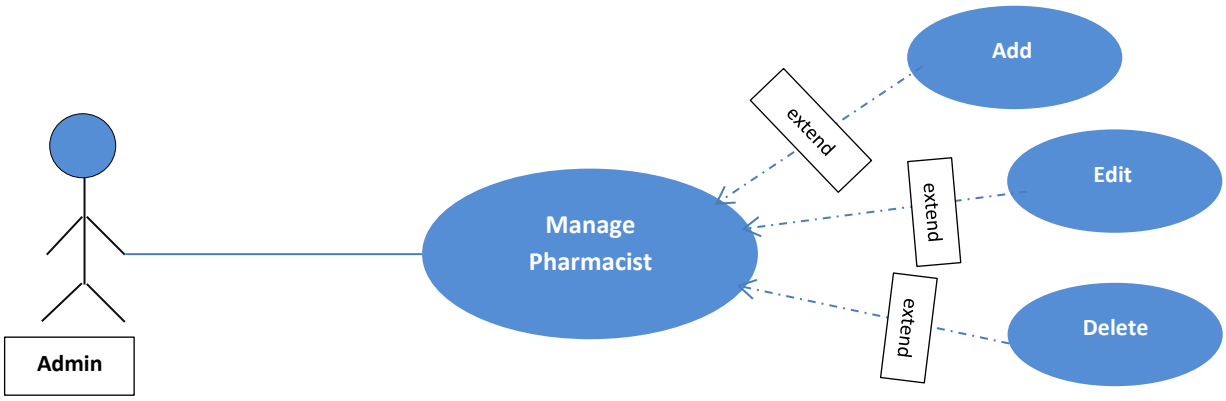



Edit sequence diagram
Admin
View
Controller
Model
Select and enter **Nurse** data
update ()
update ()
Validation Response
Return true
Success Messages

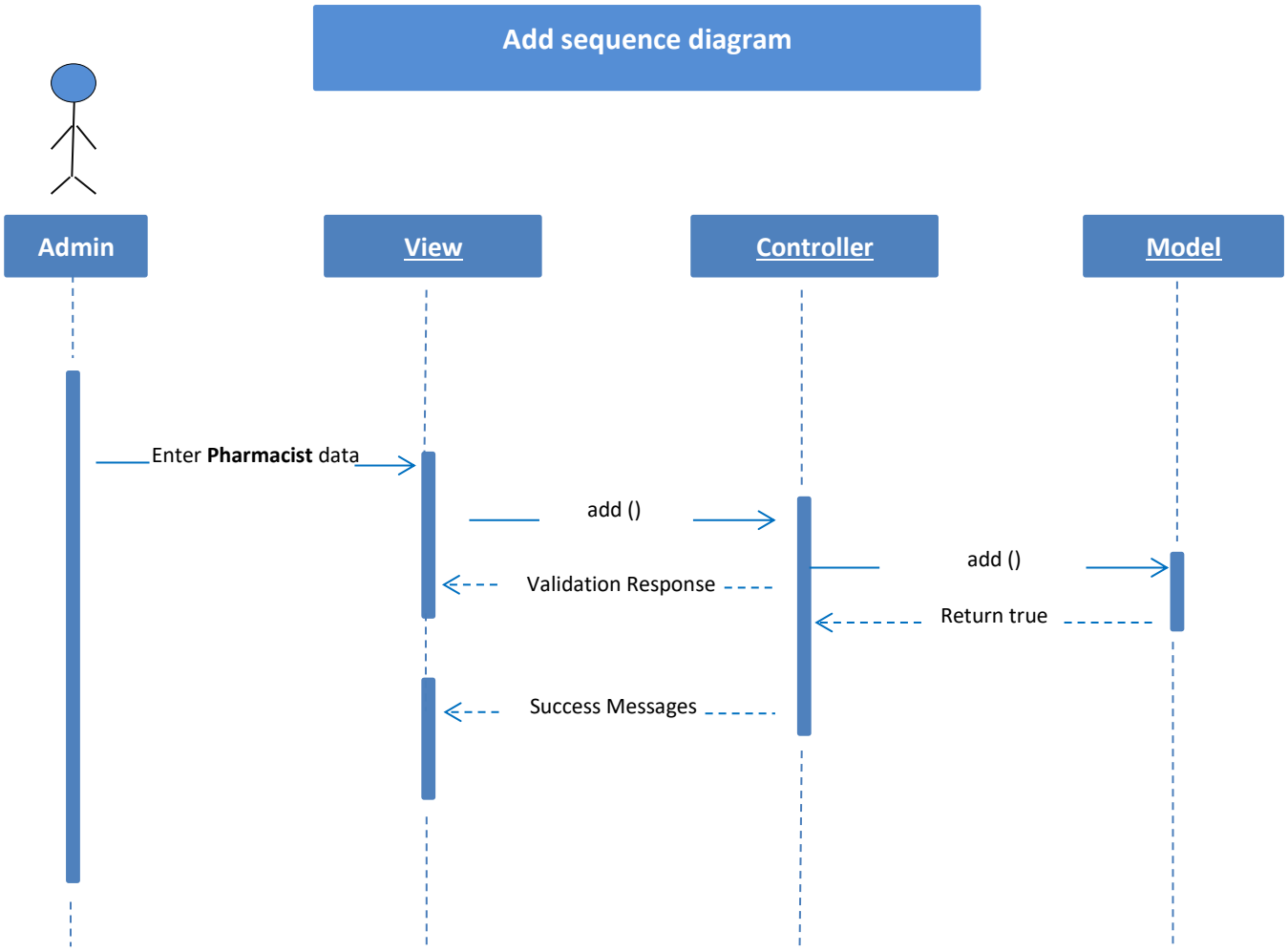

Remove sequence diagram
Admin
View
Controller
Model
Select and remove **Nurse**
remove ()
remove ()
Return true

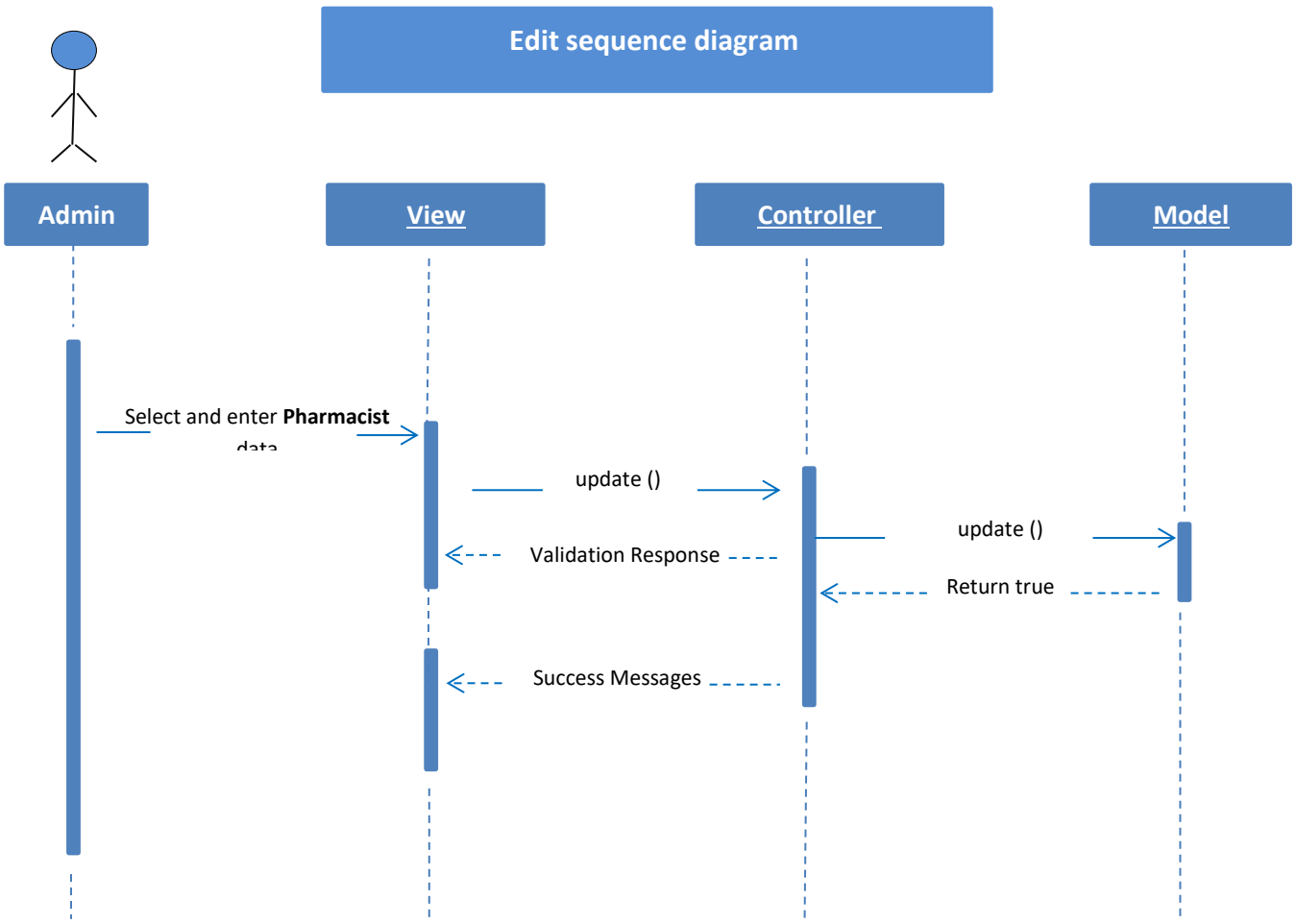
Success Messages

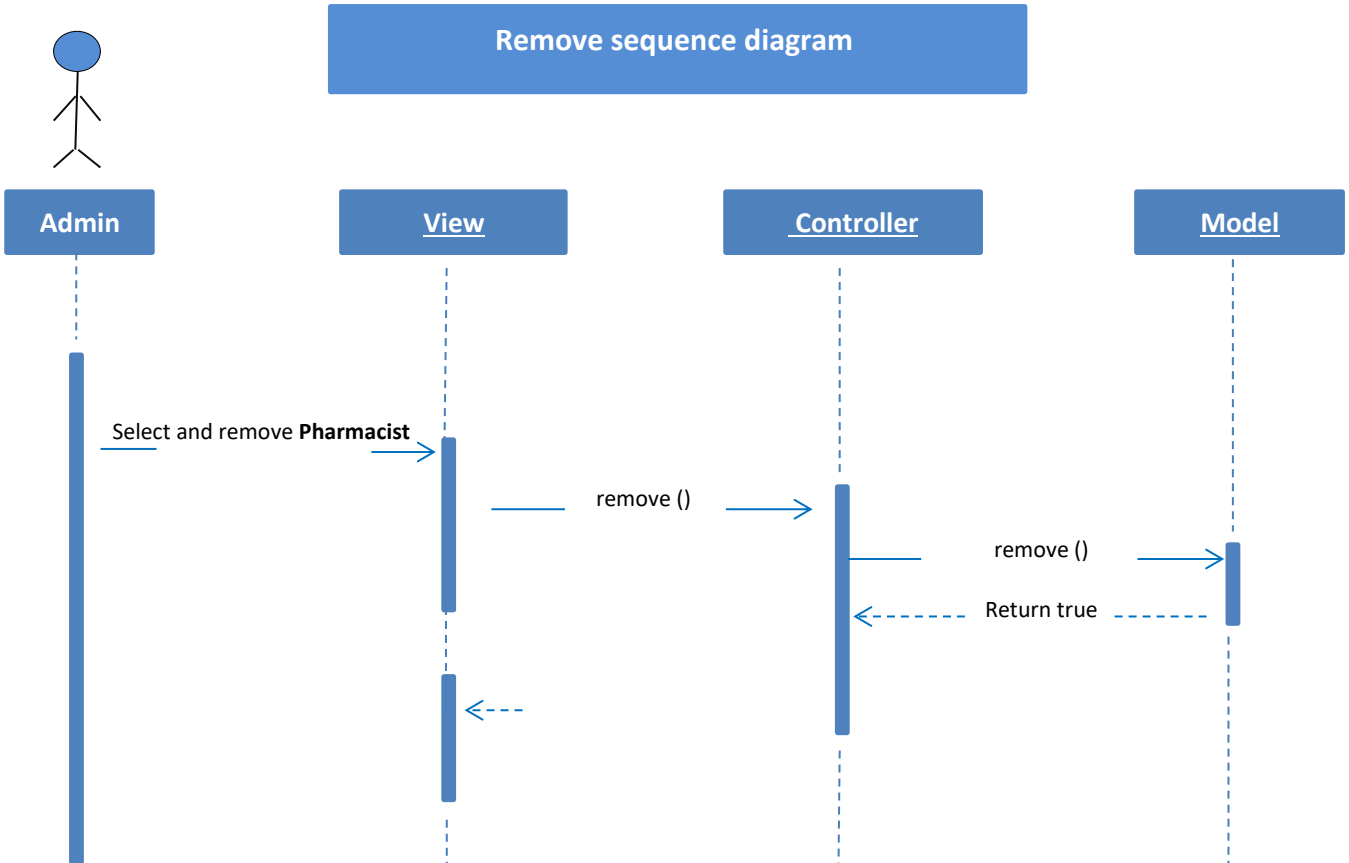
4.6.1 Use Case Diagram of Manage Pharmacist



4.6.2 Sequence Diagram of Manage Pharmacist

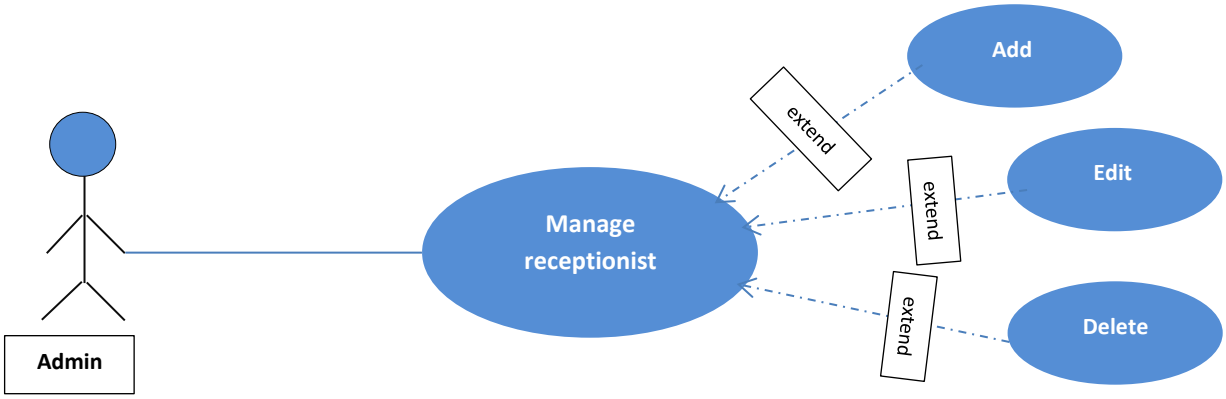



Edit sequence diagram
Admin
View
Controller
Model
Select and enter **Pharmacist** data
update ()
update ()
Validation Response
Return true
Success Messages

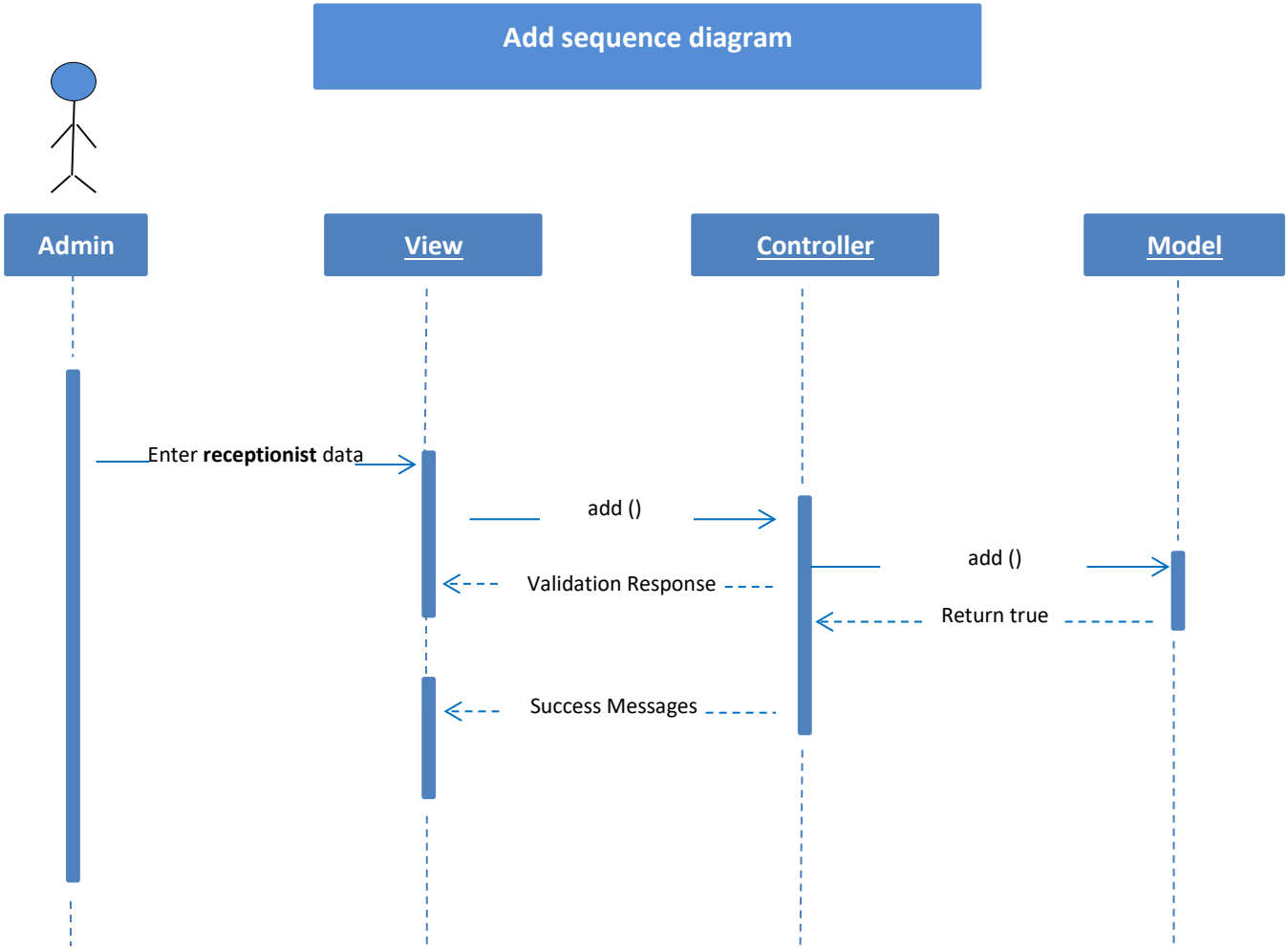

Remove sequence diagram
Admin
View
Controller
Model
Select and remove **Pharmacist**
remove ()
remove ()
Return true

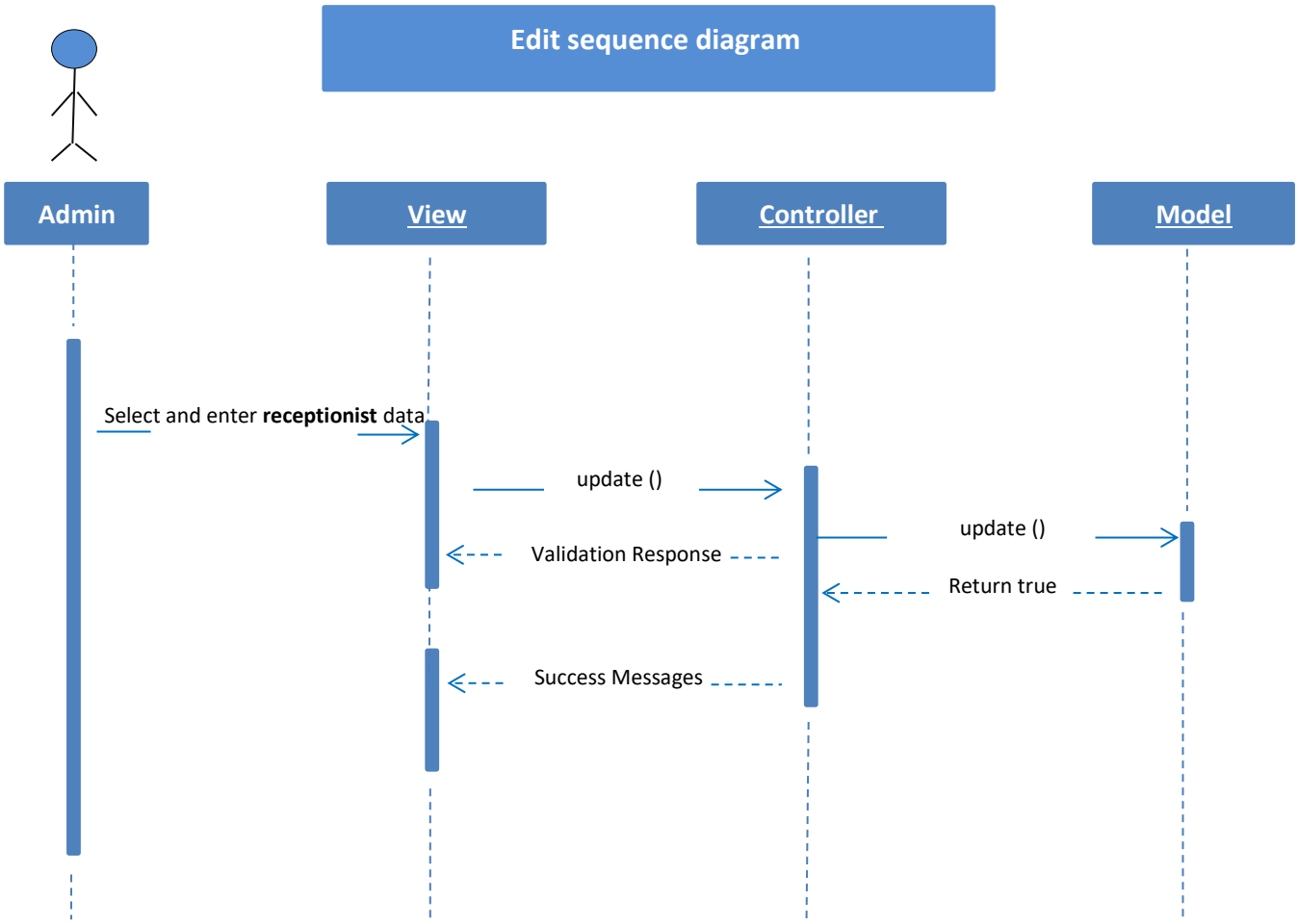
Success Messages

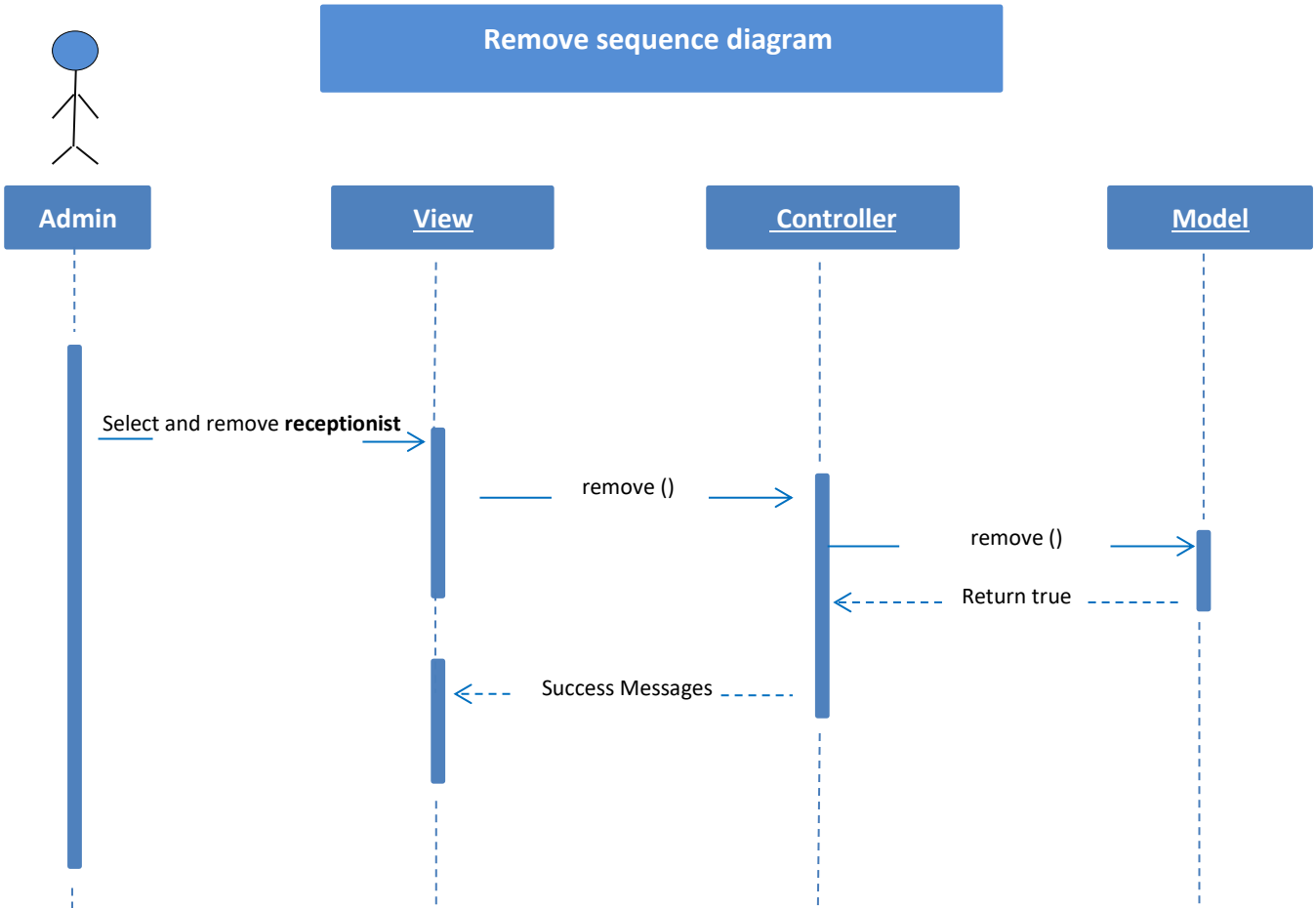
4.6.1 Use Case Diagram of Manage receptionist



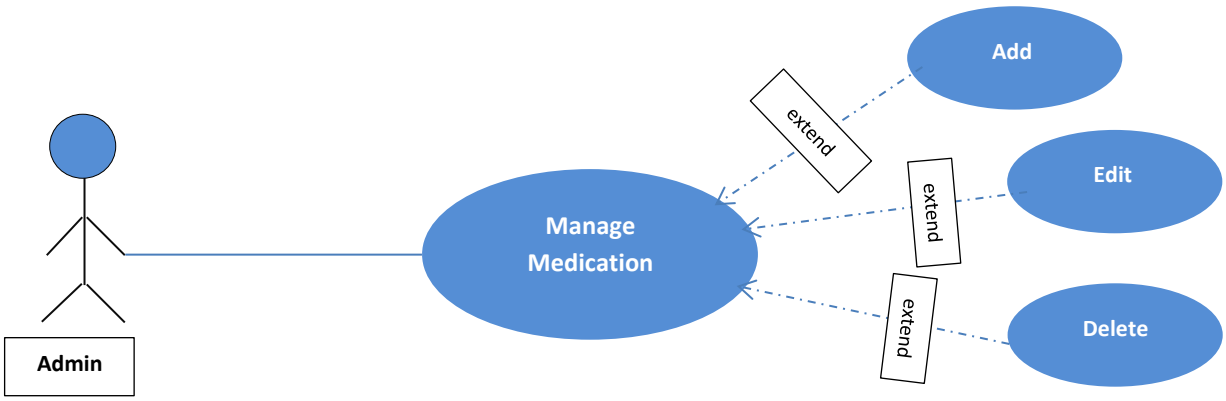
4.6.2 Sequence Diagram of Manage receptionist



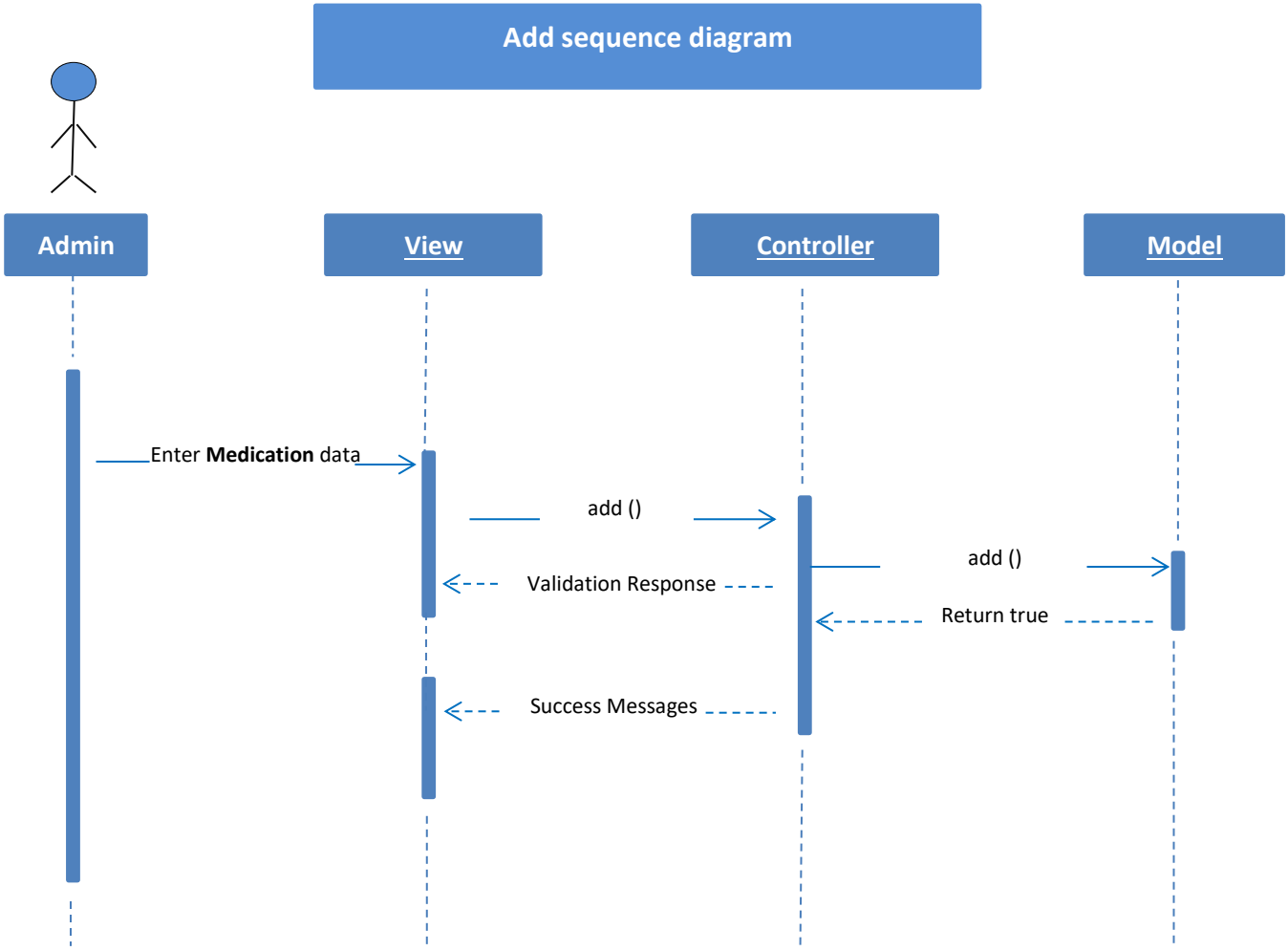

Edit sequence diagram
Admin
View
Controller
Model
Select and enter **receptionist** data
update ()
update ()
Validation Response
Return true
Success Messages

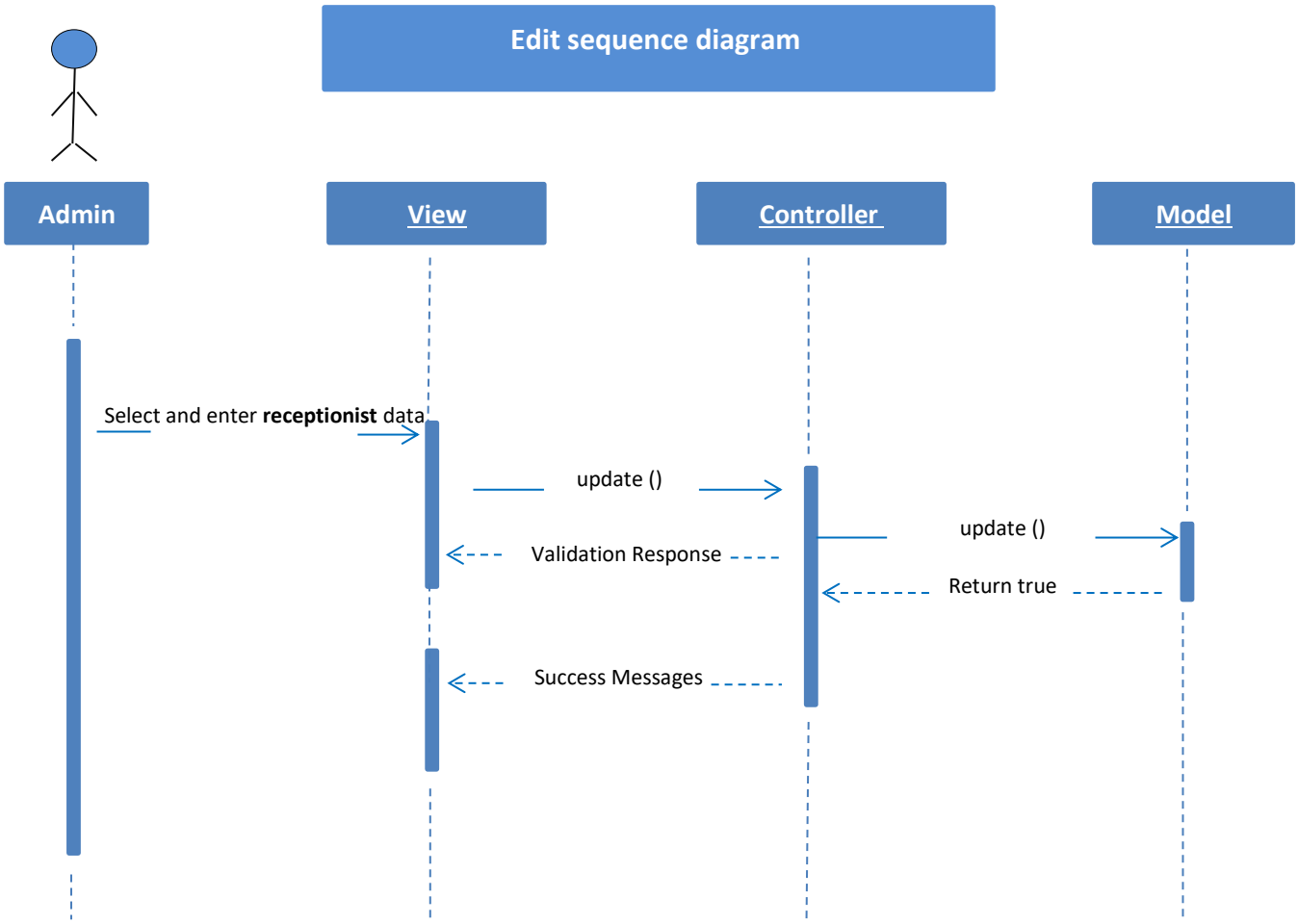

Remove sequence diagram
Admin
View
Controller
Model
Select and remove **receptionist**
remove ()
remove ()
Return true
Success Messages

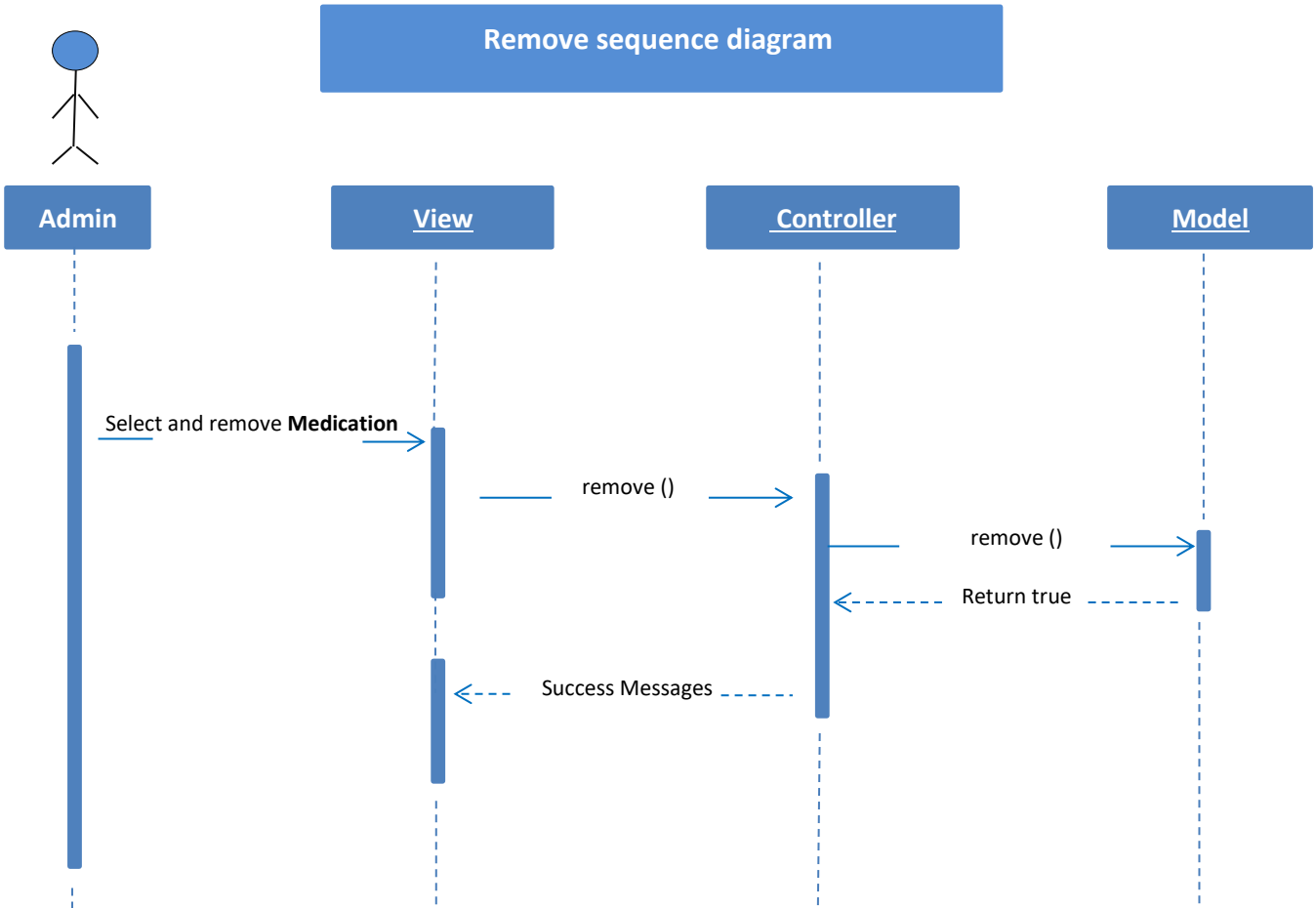
4.7.1 Use Case Diagram of Manage Medication



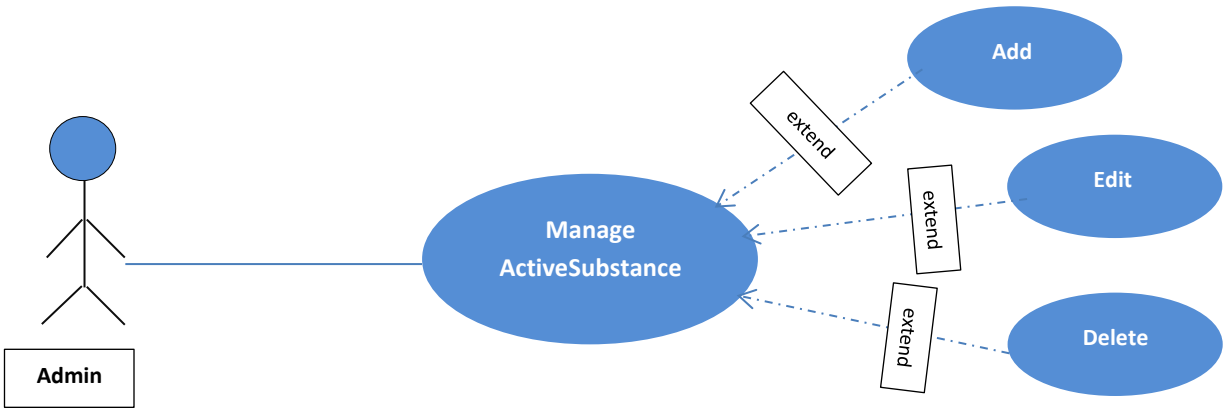
4.7.2 Sequence Diagram of Manage Medication



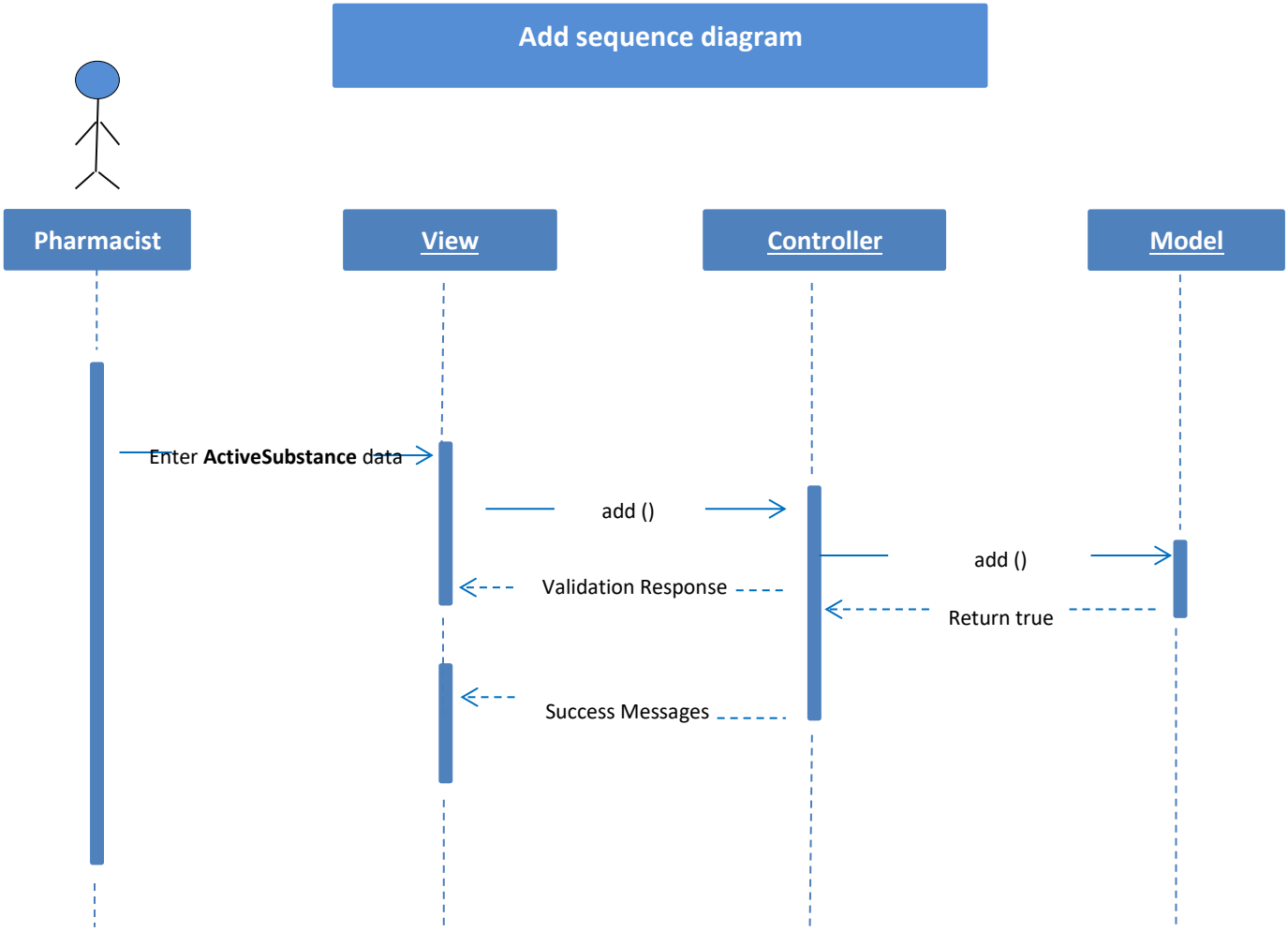

Edit sequence diagram
Admin
View
Controller
Model
Select and enter **receptionist** data
update ()
update ()
Validation Response
Return true
Success Messages

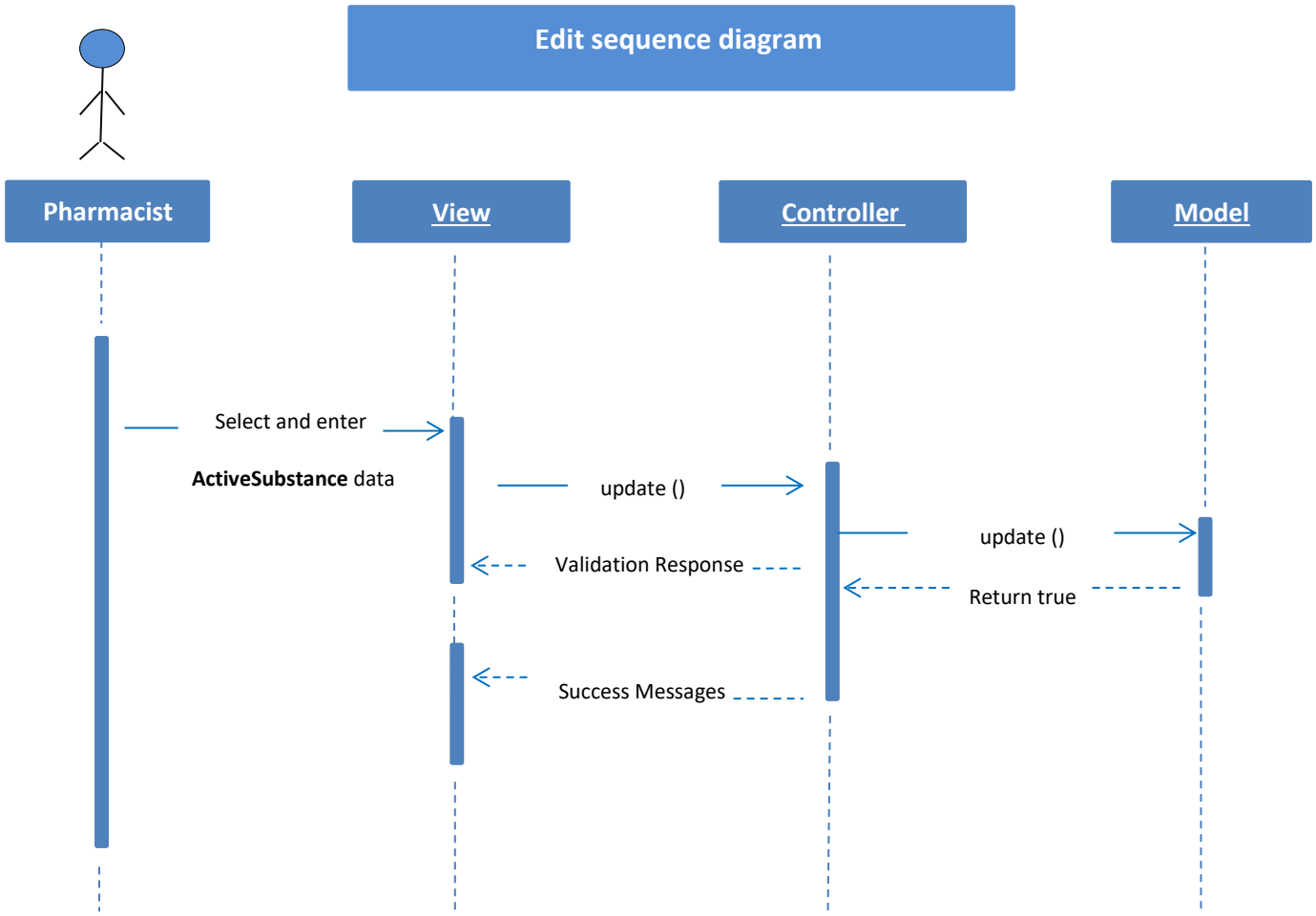

Remove sequence diagram
Admin
View
Controller
Model
Select and remove **Medication**
remove ()
remove ()
Return true
Success Messages

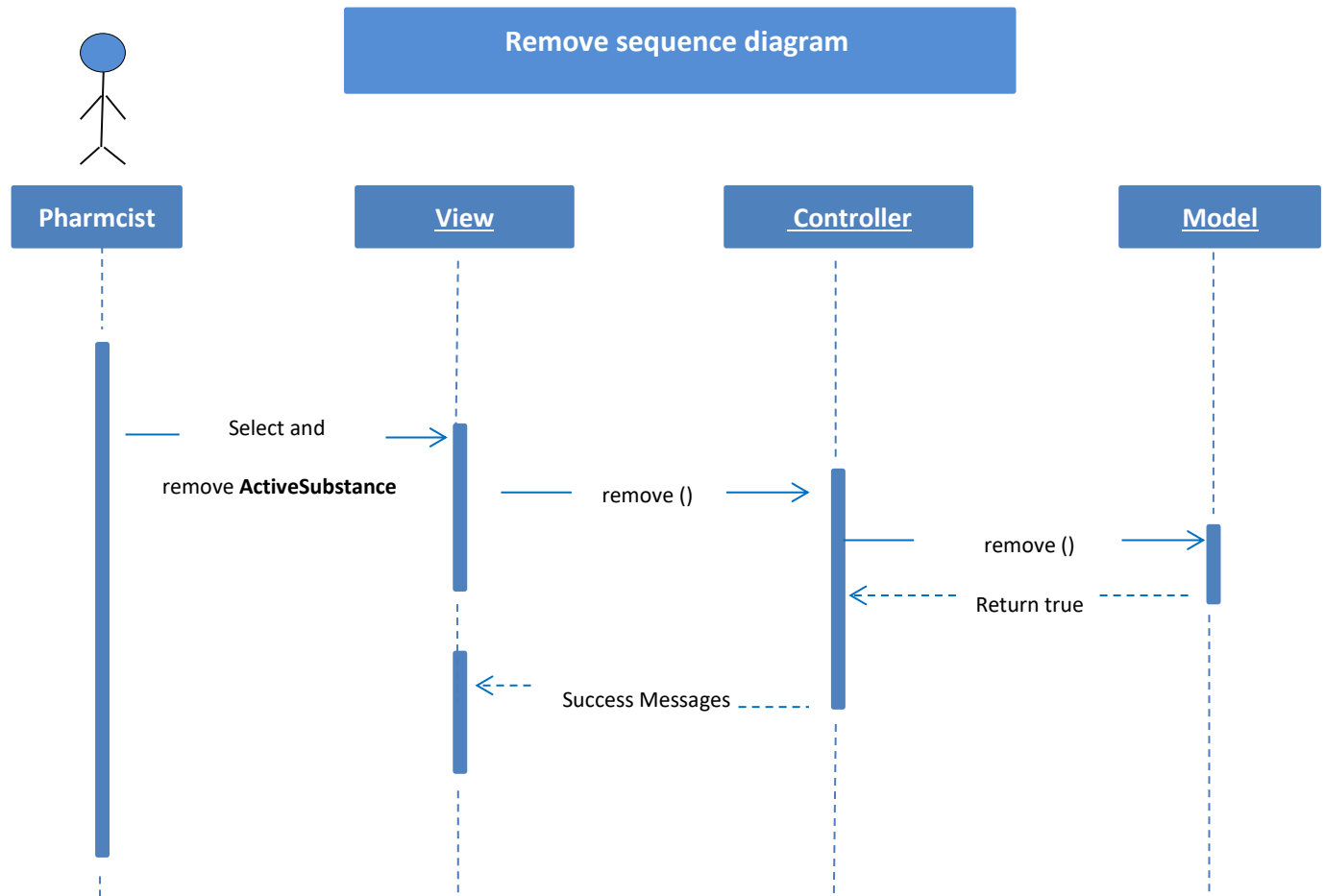
4.8.1 Use Case Diagram of Manage ActiveSubstance



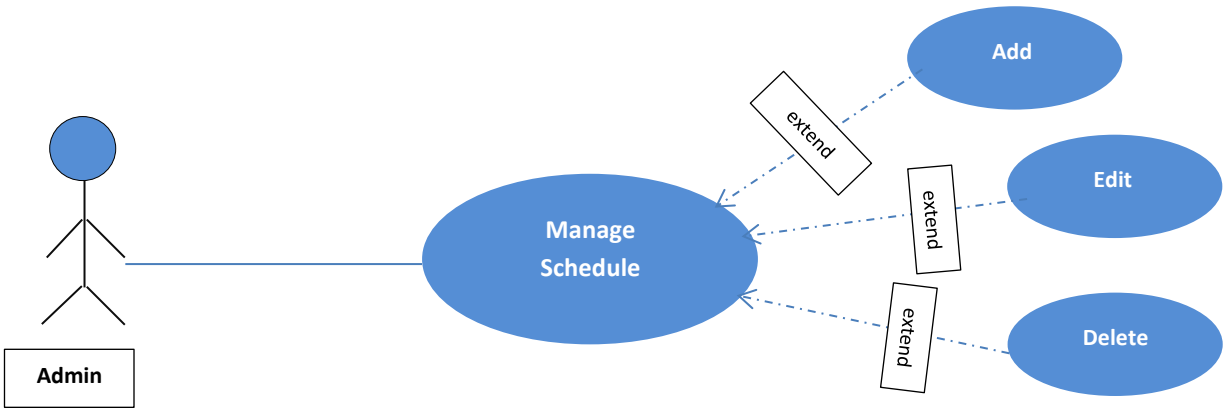
4.8.2 Sequence Diagram of Manage ActiveSubstance



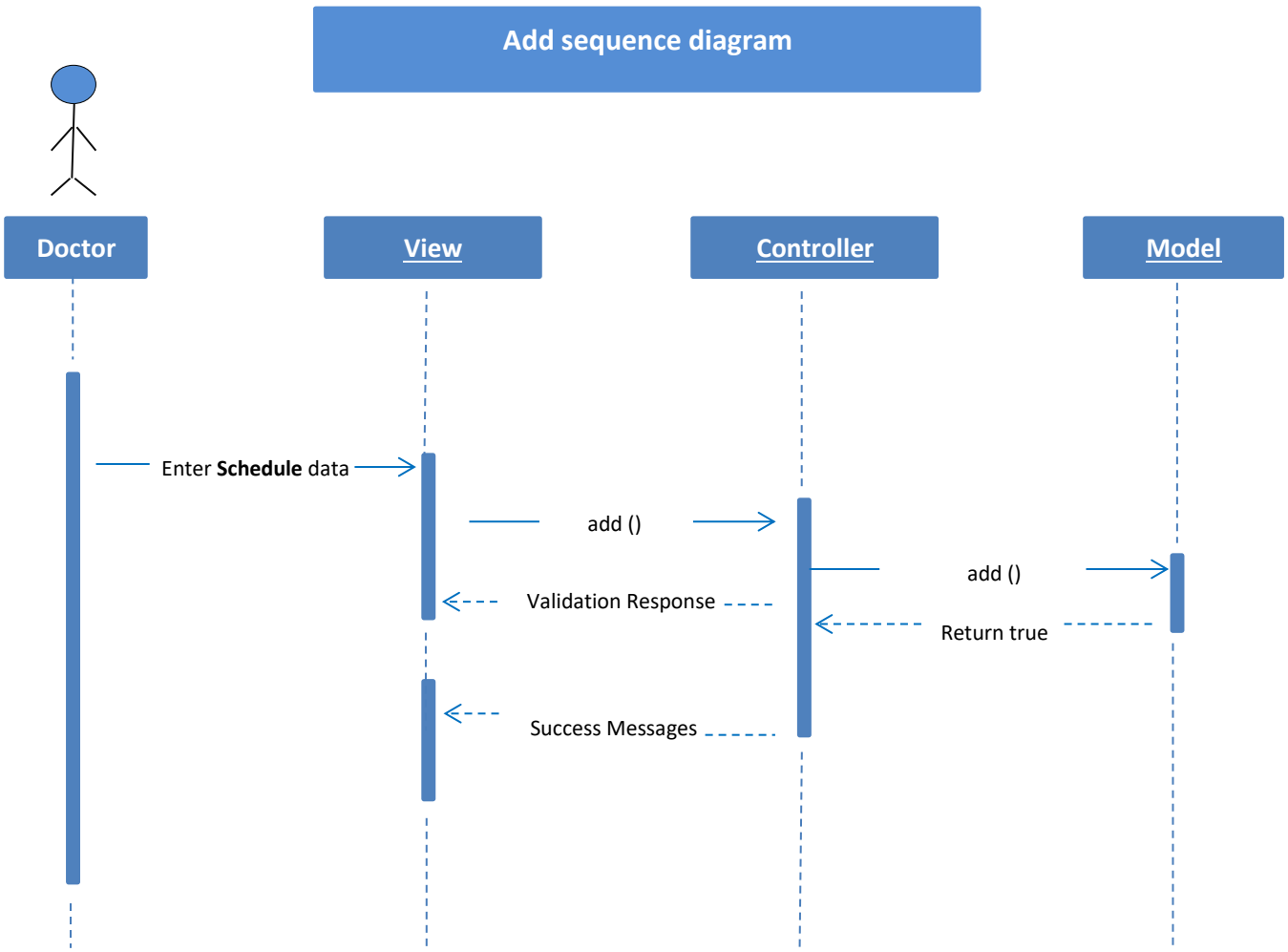

Edit sequence diagram
Pharmacist
View
Controller
Model
Select and enter
ActiveSubstance data
update ()
update ()
Validation Response
Return true
Success Messages

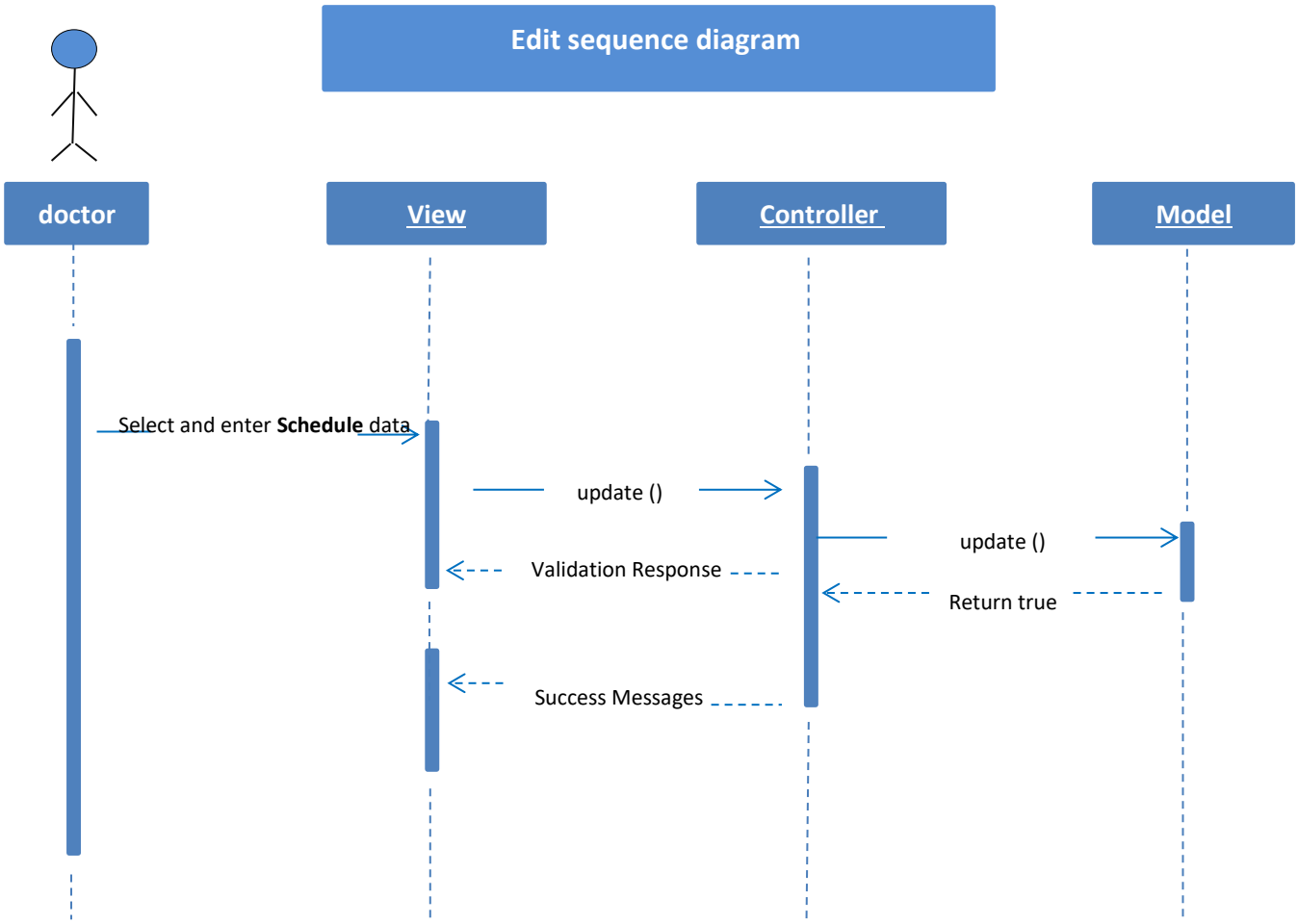

Remove sequence diagram
Pharmcist
View
Controller
Model
Select and
remove ActiveSubstance
remove ()
remove ()
Return true
Success Messages

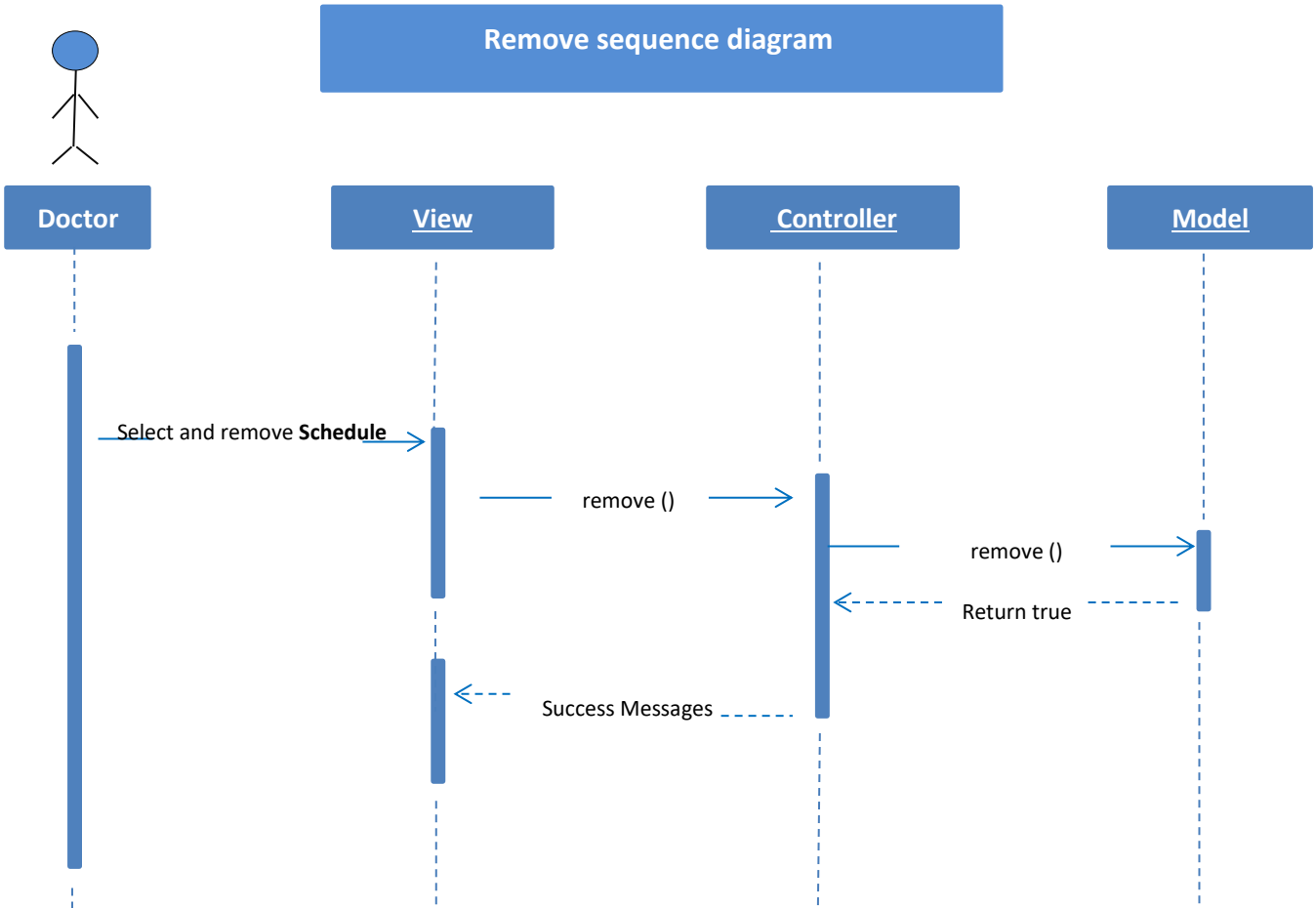
4.9.1 Use Case Diagram of Manage Schedule



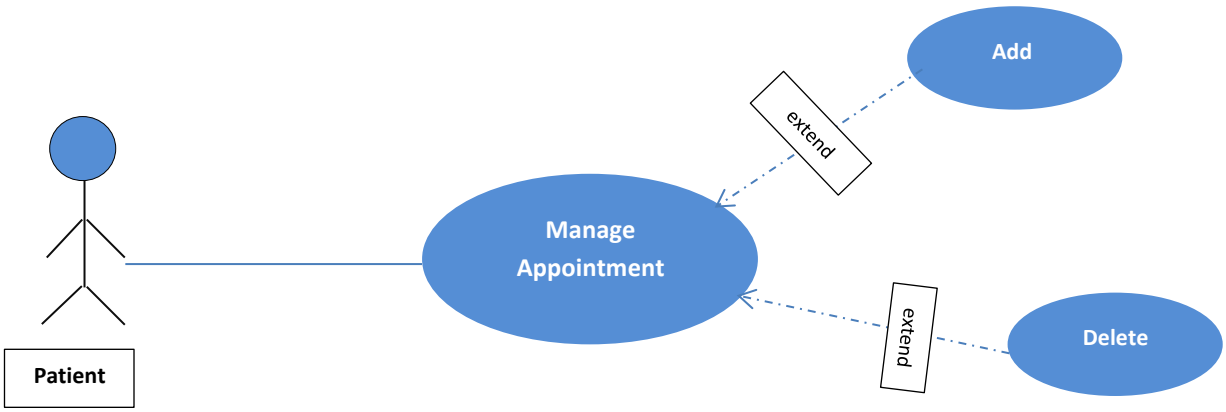
4.9.2 Sequence Diagram of Manage Schedule



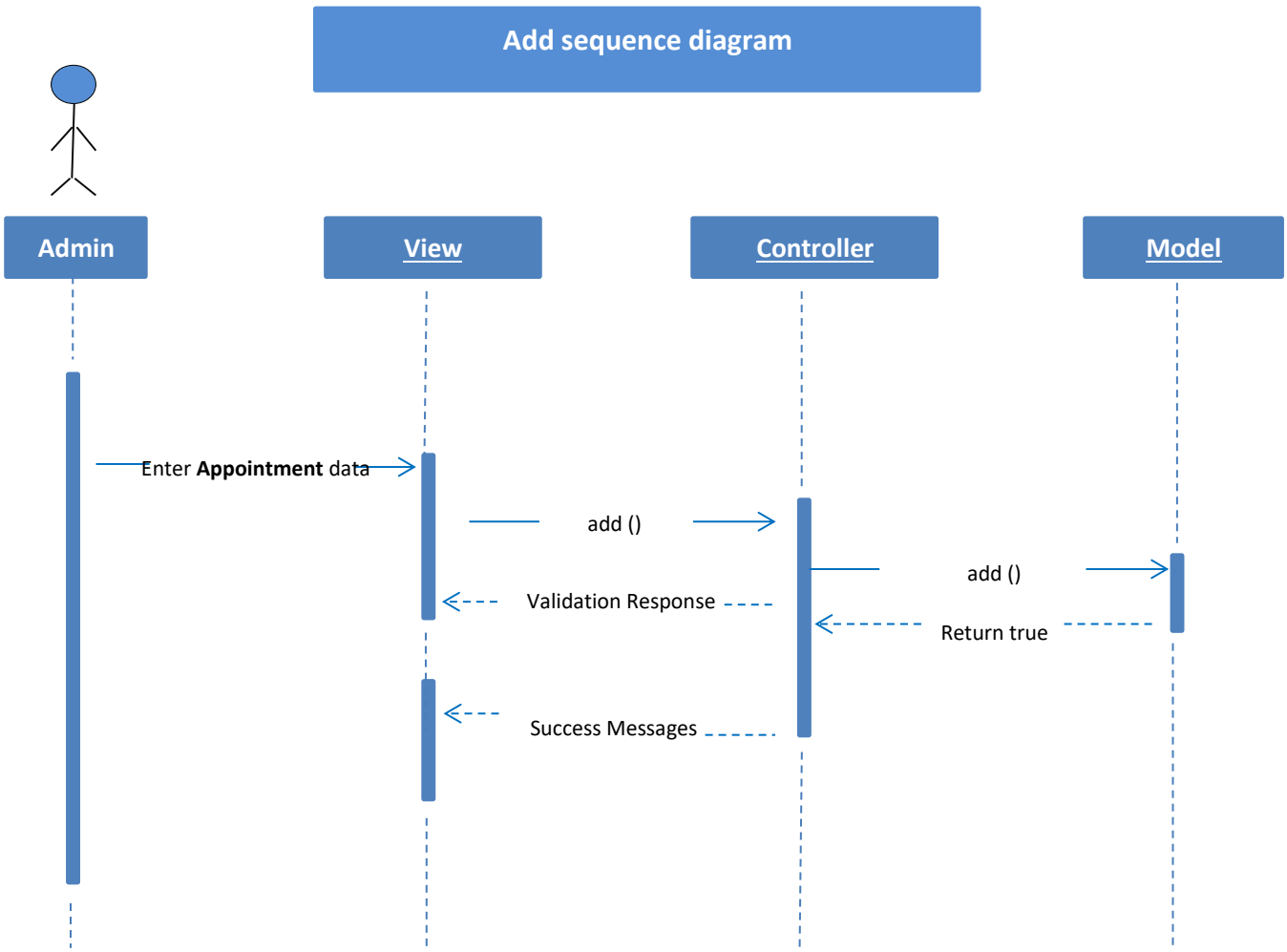

Edit sequence diagram
doctor
View
Controller
Model
Select and enter **Schedule** data
update ()
update ()
Validation Response
Return true
Success Messages

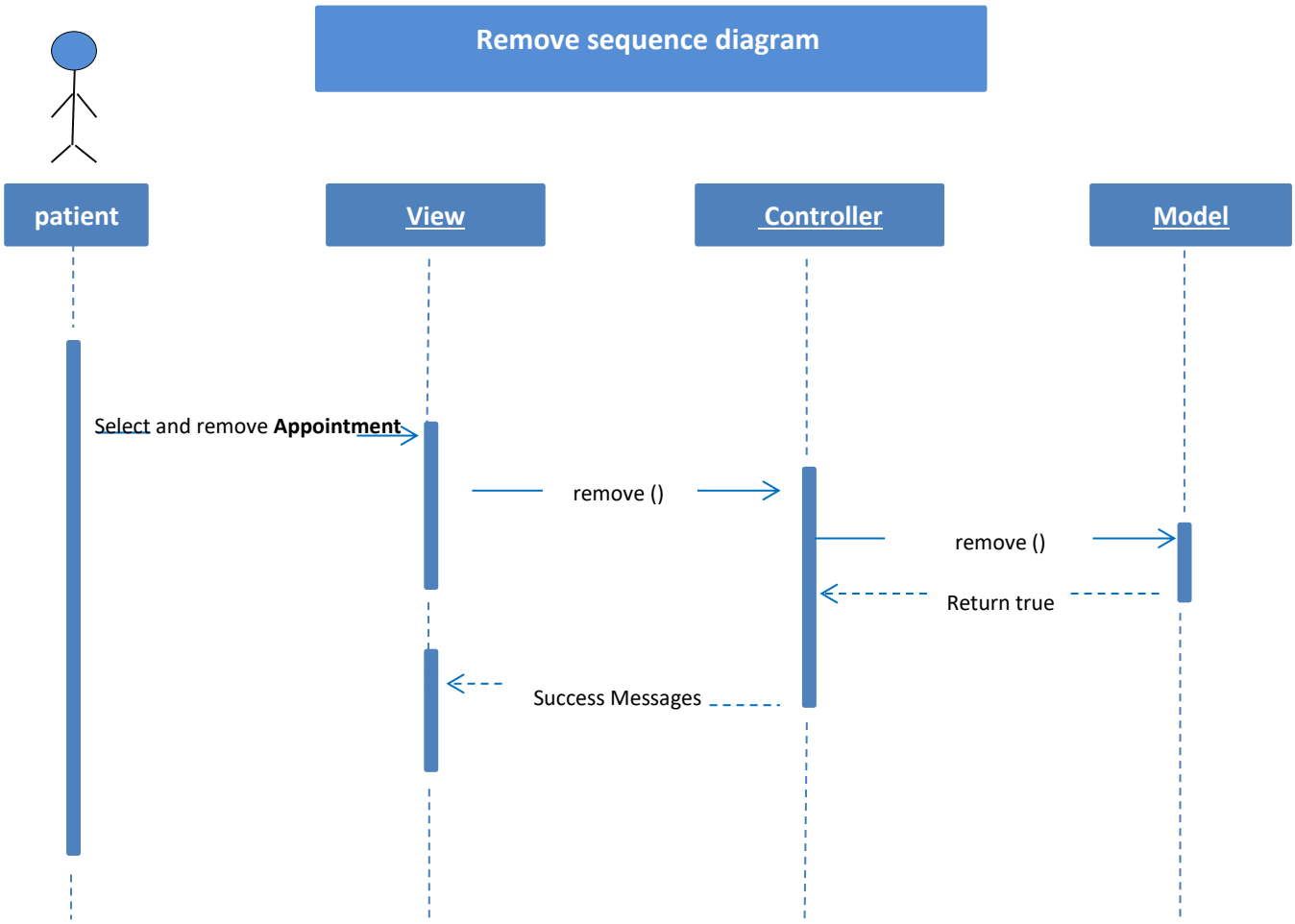

Remove sequence diagram
Doctor
View
Controller
Model
Select and remove **Schedule**
remove ()
remove ()
Return true
Success Messages

4.10.1 Use Case Diagram of Manage Appointment

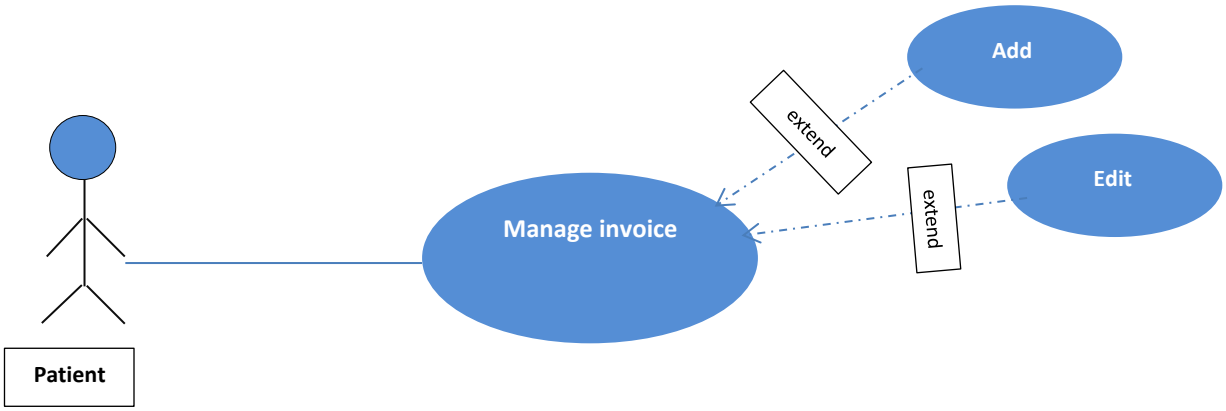


4.10.2 Sequence Diagram of Manage Appointment

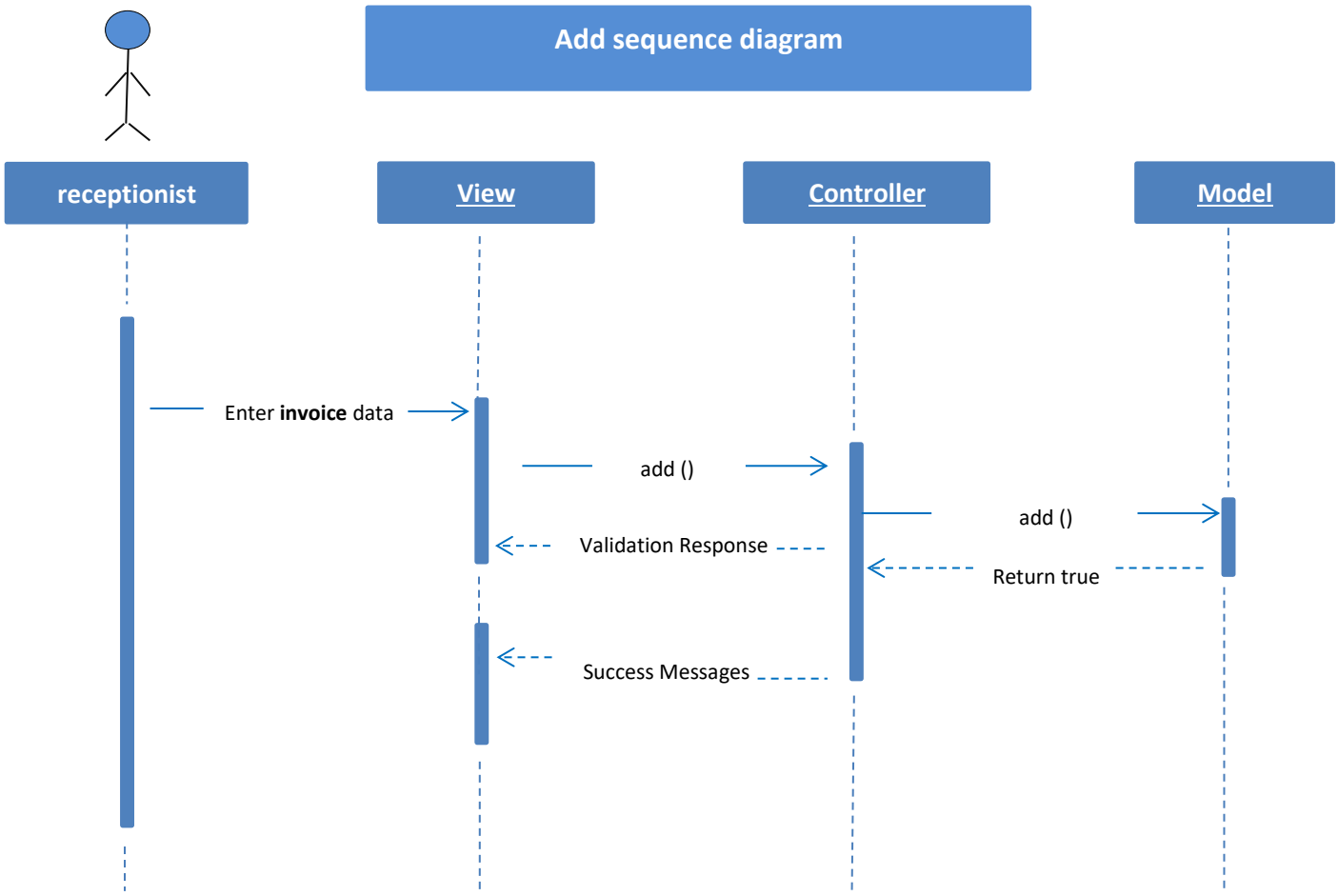


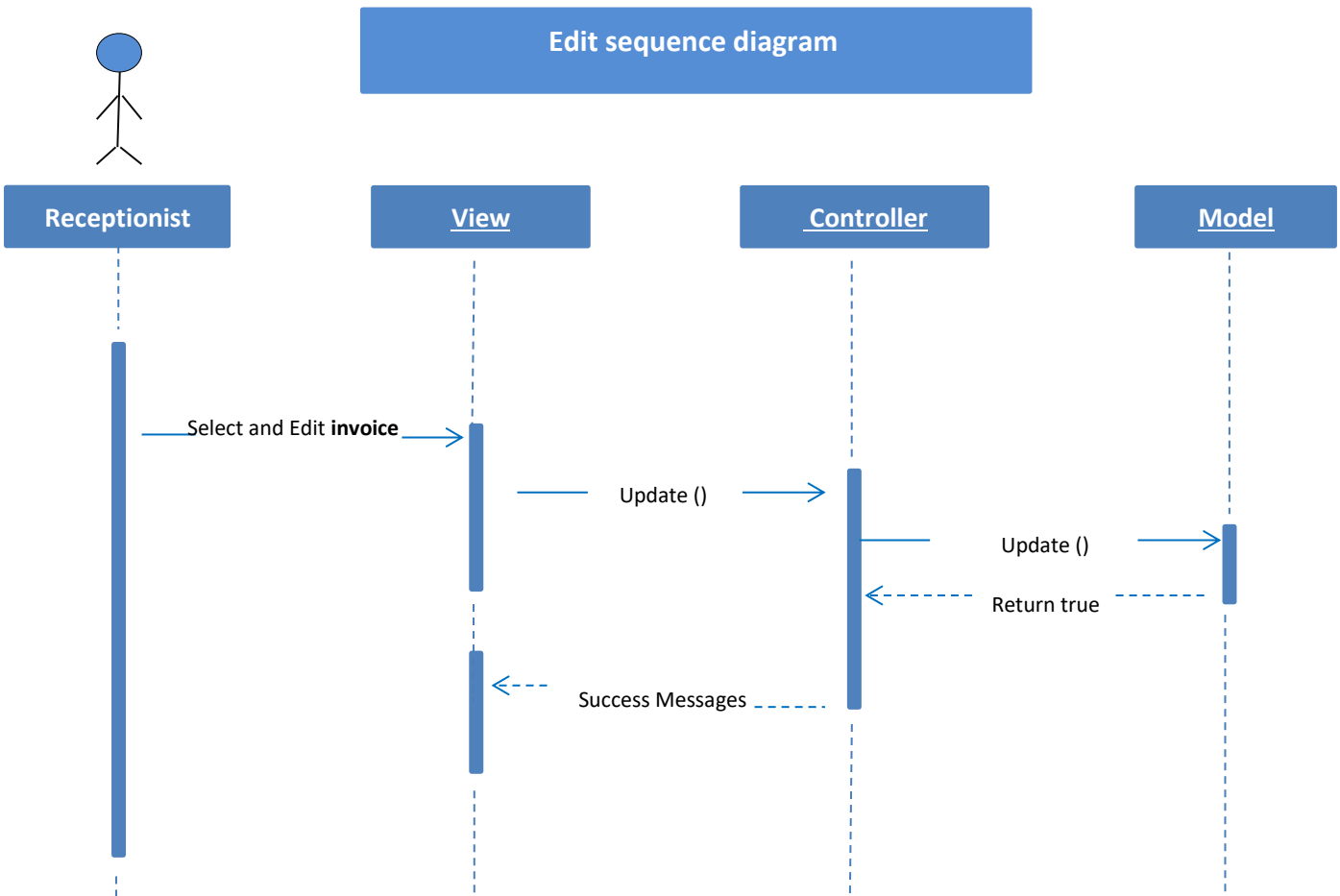


4.11.1 Use Case Diagram of Manage invoice

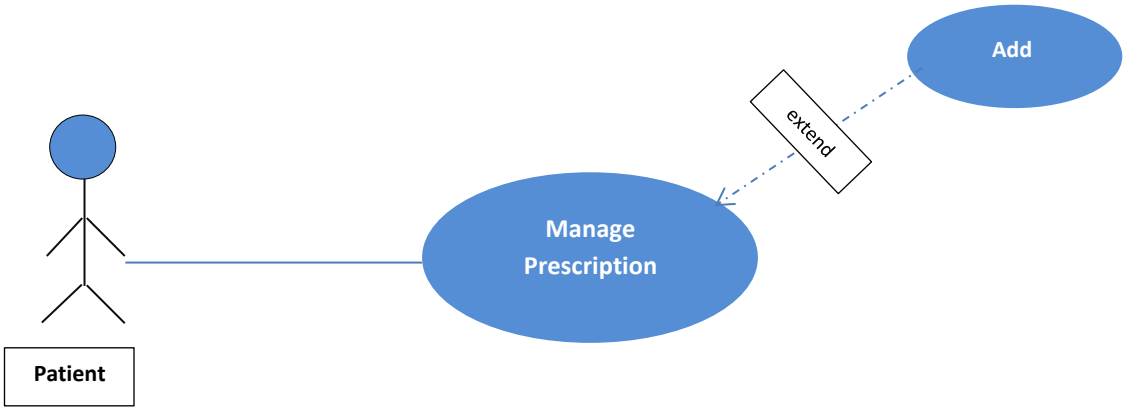


4.11.2 Sequence Diagram of Manage invoice

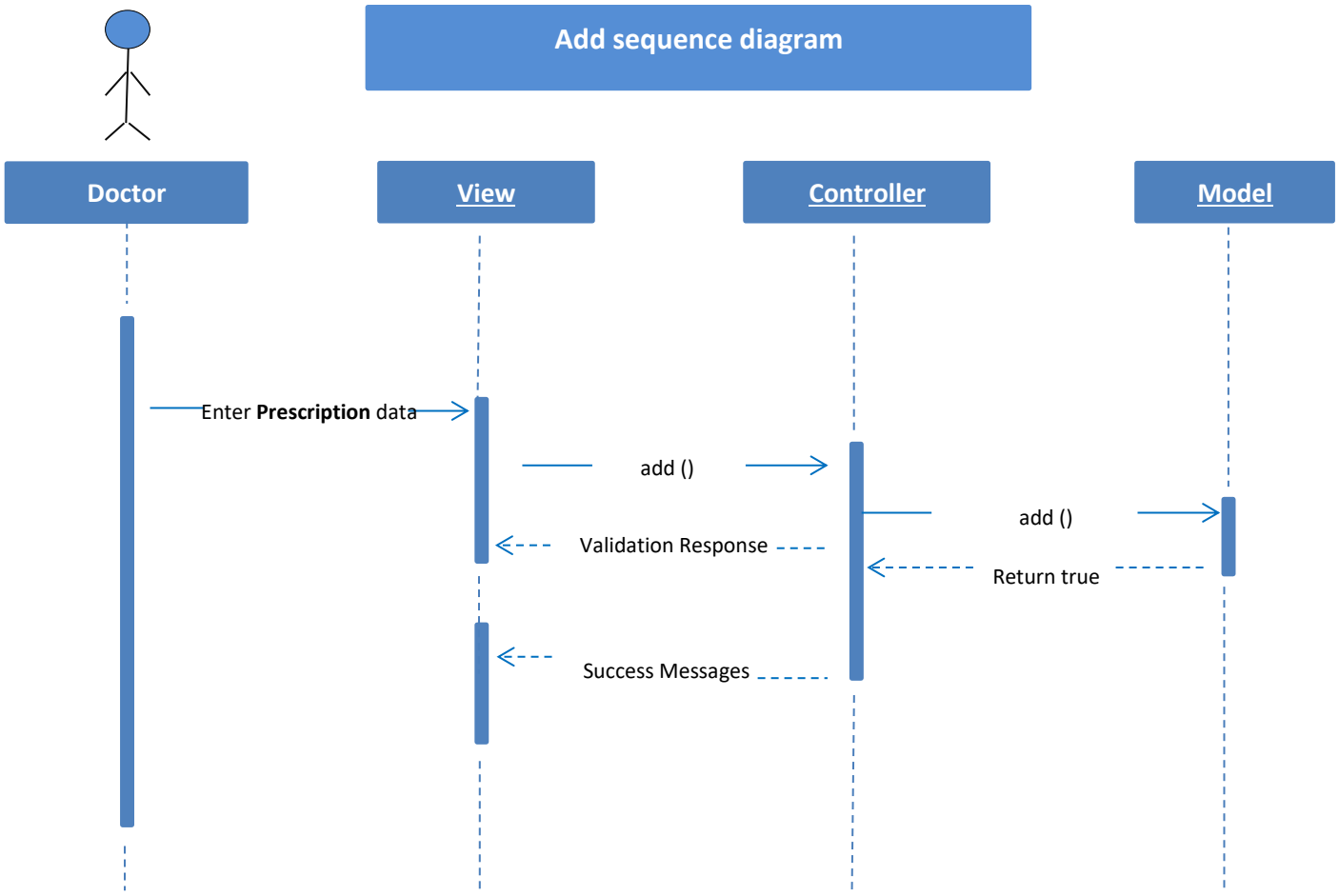




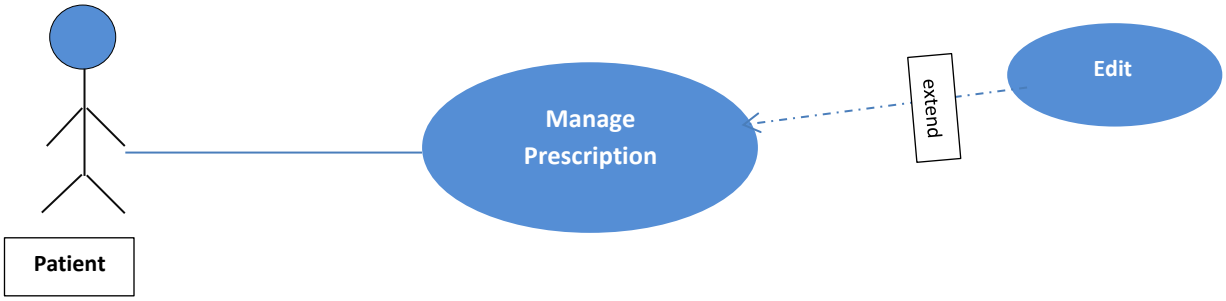
4.12.1 Use Case Diagram of Manage Prescription



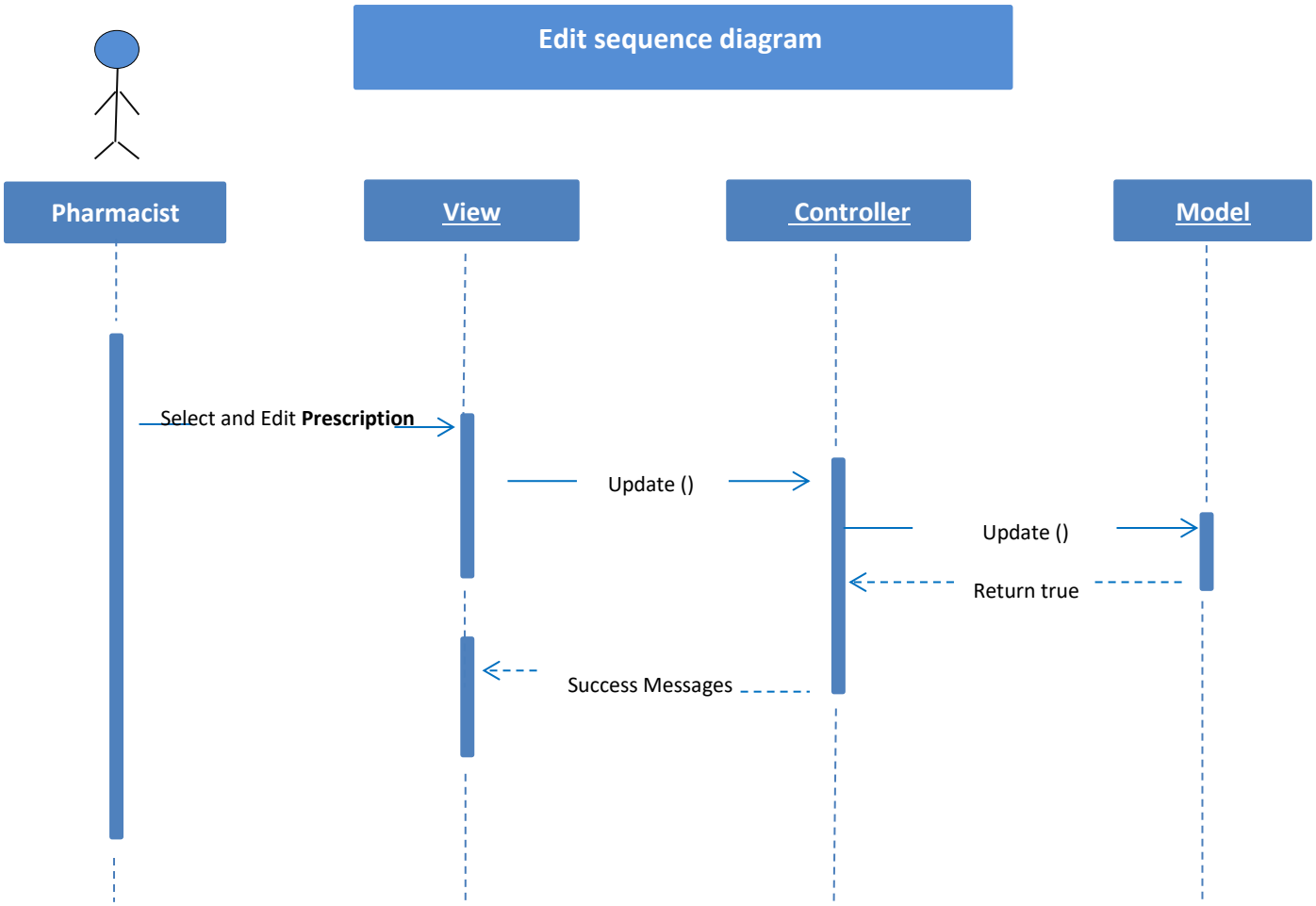
4.12.2 Sequence Diagram of Manage Prescription



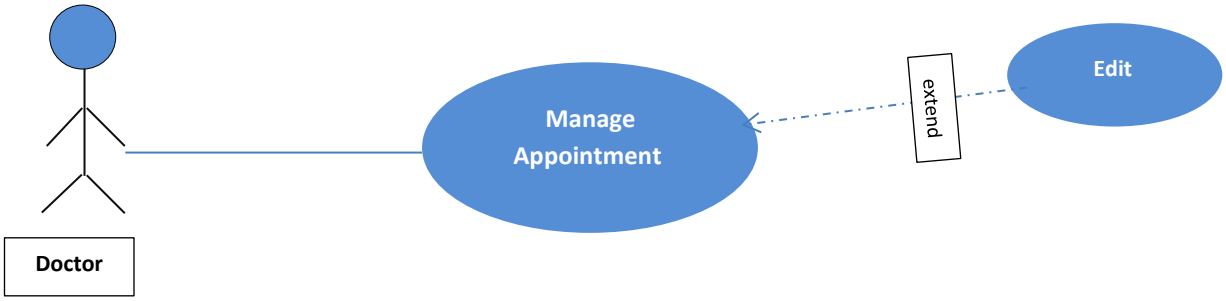
4.13.1 Use Case Diagram of Manage Prescription



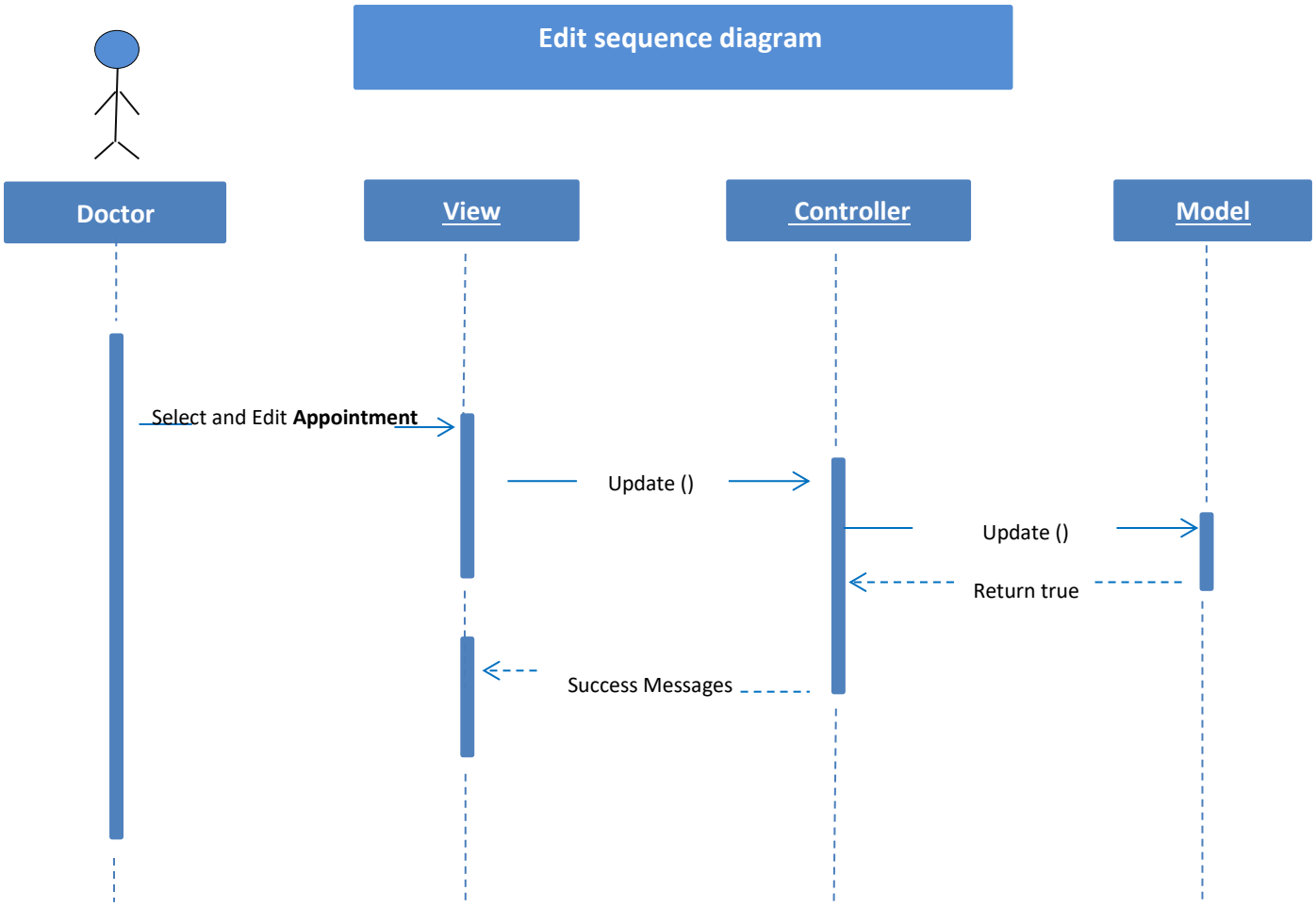
4.13.2 Sequence Diagram of Manage Prescription



4.14.1 Use Case Diagram of Manage Appointment



4.14.2 Sequence Diagram of Manage Appointment



4.15 Entity Relationship Diagram

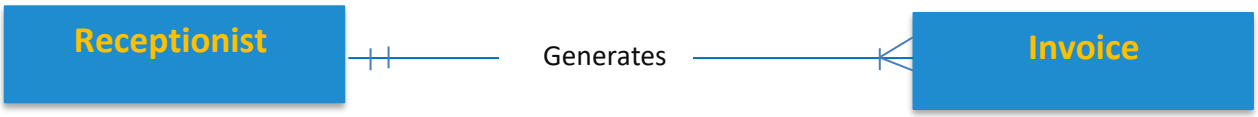
4.15.1 List of Entities

There are 14 entities used in this system. All entities are listed below:

4.15.2 Relationship with Entities

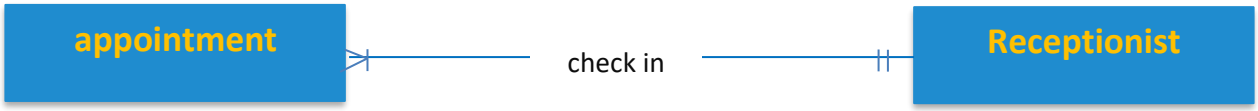
The figures below show the relationship between the entities:

4.1.2.1 Receptionist – Invoice



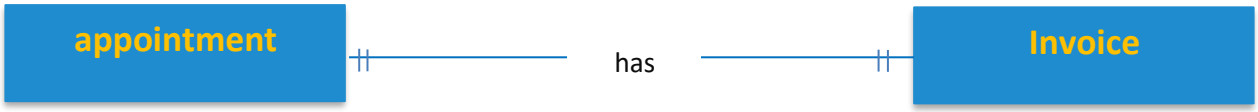
- Invoice has one Receptionist
- Receptionist Make many Invoice

4.15 2.2.appointment– Receptionist



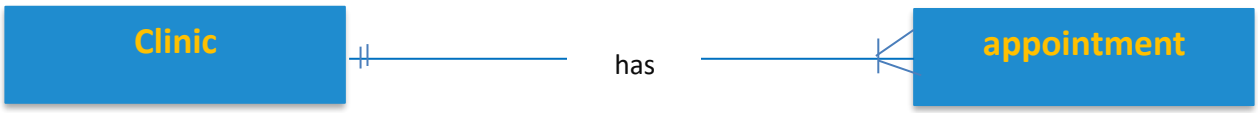
- appointment has one Receptionist
- Receptionists have Many appointments

4.152.3. Books – Authors



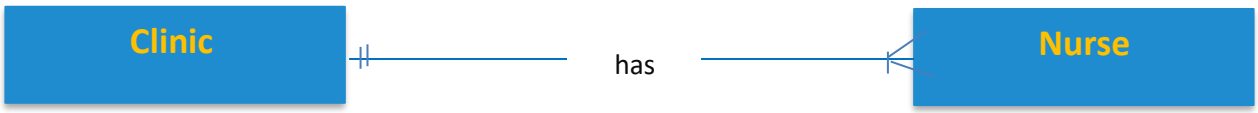
- One appointment has one Invoice

4.15..2.4 appointment – Clinic



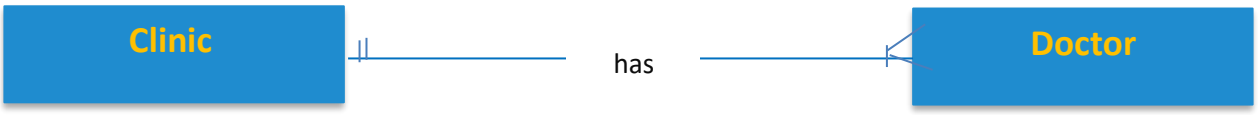
- Clinic has many appointments
- Appointment has one clinic

4.15 2.5.Clinic – Nurse



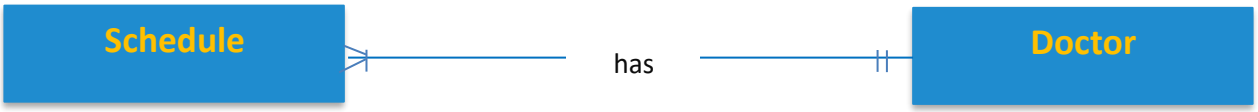
- Nurse has one Clinic
- Clinic has many Nurse

4.15 2.5.Clinic – Doctor



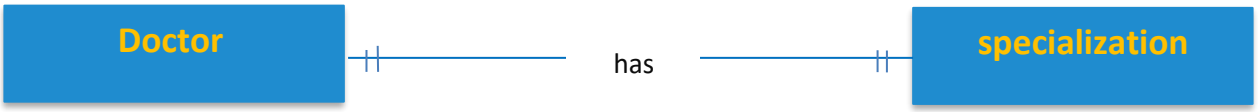
- Doctor has one Clinic
- Clinic has many Doctor

4.15..2.schedule – Doctor



- Schedule has one doctor
- Doctor has many Schedule

4.15..2.7specialization – Doctor

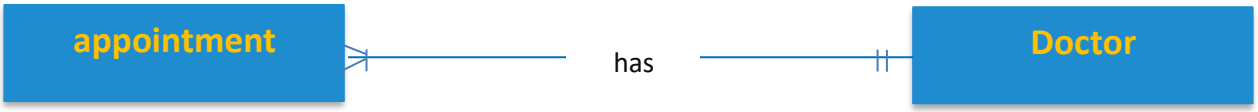


- One Doctor has on specialization and revise

4.15.2.8 Medication– ActiveSubstance

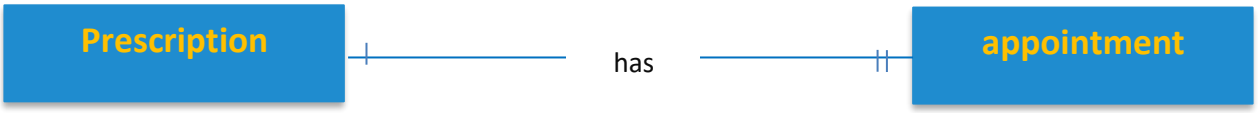
- Medication has many ActiveSubstance
- ActiveSubstance has many Medication

4.15.2.9 Doctor –Appointment



- appointment have one doctor
- One Doctor has Many appointments

4.15.2.11 Prescription – appointment



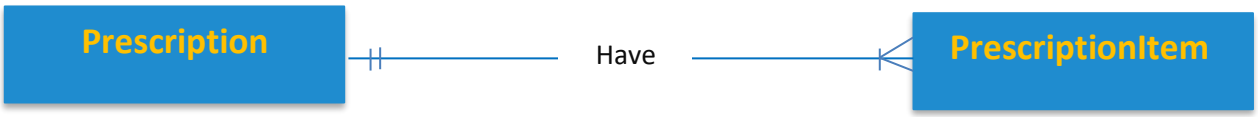
- Appointment may have Prescription

- One Prescription must have appointment

4.15.2.12Borrowed PrescriptionItem – Medication

- PrescriptionItem may has many Medications

4.15.2.13 Prescription– PrescriptionItem



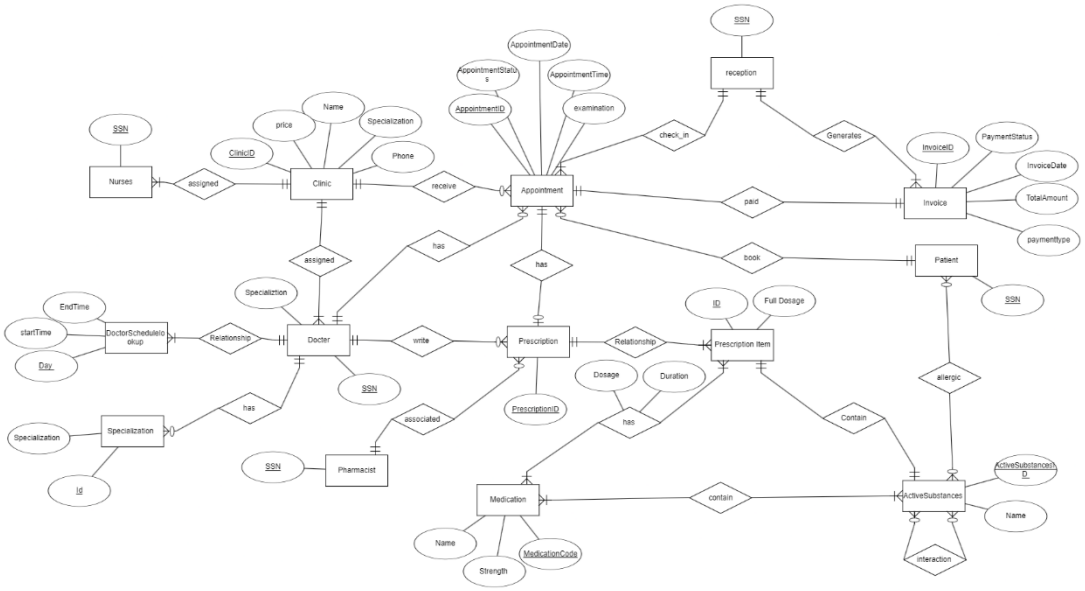
- Prescription may have many PrescriptionItem
- PrescriptionItem must has one Prescription

4.152.14. Pharmacist – Prescription

- Pharmacist has many Prescriptions
- Prescription has one Pharmacist

4.15..2.14 Patient – ActiveSubstance

4.16 ER – Diagram



Database Schema: