Software Requirement Specification, Analysis and Design

Doctor’s Appointment System

Course: SE 801

Submitted by:

Khairul Araf
BSSE 0631

Supervised by:

Amit Seal Ami
Lecturer
Institute of Information Technology
University of Dhaka

LETTER OF TRANSMITTAL

26th October, 2017

4th year program committee
Institute of Information Technology (IIT)
University of Dhaka

Sir,
I have prepared the report on Software Requirements Specification of "**Doctor's Appointment System"** for your approval. This report details the requirements I gathered for the project.

The primary purpose of this report is to summarize our findings from the work that I completed as our Software Requirements Specification and Analysis course project. This report includes the details of each step I followed to collect the requirements.

Sincerely yours,

Khairul Araf

BSSE 0631

Document Authentication

This document is verified by the following individuals

……………………………………………………….	……………………………………………......
Amit Seal Ami	Khairul Araf
Lecturer	BSSE 0631
Institute of Information Technology	
University of Dhaka	

Acknowledgements

By the Grace of almighty Allah, I have completed the final report of "**Doctor's Appointment System**". I am grateful to my supervisor Amit Seal Ami for his supervision throughout the working time. He helped me a lot by sharing his valuable knowledge to complete this document.

Table of Contents

Figures

Chapter 1

Introduction

1.1 Purpose

This document is the SRS for ‘Doctor Appointment System (DAS)’ application which is capable to run on any device with a browser. For making an appointment, DAS will provide an automated system which will give assistance to the doctors, their assistants and the patients.

Requirements is one of the key component of SRS document. So the elaborated, common, conflicting, and final requirements are incorporated by this chapter, establishing a requirements baseline for the development of the system. The requirements accommodated in this SRS are independent, uniquely numbered and organized by topics. The SRS will be evolved over time as users and developers work together to validate, clarify and expand its contents.

1.2 Intended Audience

This SRS targets audience of a wide range, the client and the developer being an integral part of it.

- The client will use this SRS to substantiate that the developer has created a product that is acceptable.
- In order to ensure that the developer is on track during development of the system. The supervisor of the developer will use this SRS to plan milestones and a delivery date.
- For creating the system’s design, the designer will use this SRS as a guideline.
- To ensure that the system they are designing will fulfill the clients need, the designer will continuously refer back to their SRS.
- For improving the system’s functionality, the developer will use this SRS as a foundation.
- The tester will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are being completed, the tester will execute their tests on the software to verify that the software is successful in attaining the demands documented on the SRS. The testers will again run their tests on the entire system, when

it is completed and more use of that all requirements documented in the SRS have been fulfilled.

1.3 Definitions, Acronyms, and Abbreviations

Definitions, Acronyms and abbreviations used in the system are given:-

- Doctor’s Appointment System can be abbreviated as DAS.
- SRS stands for Software Requirements Specification.

1.4 Conclusion

In the above discussion a compact and primary introduction is given about SRS of DAS. The clarification of the utility of this document is made for different readers. For better understanding of this document, the illustration of the short forms are given.

The next chapter is Inception. It portrays the overall vision or plan for the DAS project. It will also define the stakeholders’ view of the product to be developed, specified in terms of the stakeholders’ key needs and features.

Chapter 2

Inception

2.1 Introduction

Requirement Engineering comprehends in several sequential steps. Inception is an essential part for which it is executed in the beginning. For the requirements analysts, inception creates the portal to the project. It refers them how the project should commence. It also provides a fundamental idea to the engineers about the problems ahead which are needed to be resolved and how critical obstacles may come during the project. The main aim of Inception phase is to distinguish the people related to the project and their requirements. In order to complete this phase, the points which are focused are:

- Identifying stakeholders
- Recognizing multiple viewpoints
- Working towards collaboration
- Asking the first questions

2.1.1 Identifying Stakeholders

Stakeholders are entities that have an interest in a given project. These stakeholders may be inside or outside an organization which:

1. Subsidize a project, or
2. Have an interest or a gain upon a successful fulfillment of a project,
3. May have a positive or negative influence in the project finalization

Stakeholders are given below:

1. **Doctors:** One of the main target users of our system. He will have a profile and is one of the direct beneficiary of the project.
2. **Doctor's Assistant**: Doctor's assistant will help doctor to manage the system.
3. **Patients:** The largest user group of the system. They will search for doctors, schedule and apply for schedule. They are one of the direct beneficiary of the system.

2.1.2 Recognizing multiple viewpoints

These viewpoints are collected by discussing with the doctors, doctor’s assistants, patients and developers.

Doctor’s Viewpoint:

Viewpoints of the stakeholder are given below:-

- System should be easily usable
- Web based interfaces
- Can easily access registered patient’s information
- Can update doctor’s schedule
- Can assign appointment fee
- Search patient’s schedule

Doctor’s assistant’s Viewpoint:

Viewpoints of the stakeholder are given below:-

- System should be easily usable
- Web based interfaces
- Can easily access registered patient’s information
- Can update doctor’s schedule
- Can assign appointment fee
- Search patient’s schedule

Patient’s Viewpoint:

Viewpoints of the stakeholder are given below:-

- User friendly interface
- Can request for schedule and after approval get the schedule easily and can cancel the schedule
- View doctor’s profile
- View doctor’s schedule and fee
- Search doctor’s by area and dimension
- Precise appointment time

Stakeholders were asked for their requirements of the DAS project and found out that each of them has their own needs which need to be dealt with. Some of the requirements are common as well as conflicting. Thus, the steps given below to merge the requirements are needed:

- Find out the familiar and conflicting requirements.
- Fractionate the requirements into distinct categories.
- Identify the distinctive requirements that the stakeholders have.
- In order to prioritize the stakeholder's priority points which would be prioritized through voting, the requirements should be identified.
- Take final decision about the requirements.

2.1.4 Asking the first questions

First set of context-free questions focusing on the customer and other stakeholders, overall project goals and benefits are –

- Who is subsidizing for the project?
- Who will be benefited from the project?
- Who gets to make the final verdict about the project (if this is different from the money source)?
- Who has resources which I need in order to get the project done?
- Whose work will my project have an impact on? (During the project and also once the project is completed).

These questions helps to identify all stakeholders, measurable benefit of the successful execution and possible alternatives to custom software development.

2.2 Conclusion

In this project, a demonstration is made for a basic understanding of the problem, the nature of the solution that is desired and the effectiveness of preliminary communication and collaboration between the stake-holders. More studies and communication will help both side (developer and client) to understand the future anticipation of the project. Hopes are high that the full functioning document will help us to define that future expectations.

Chapter 3

Elicitation

3.1 Introduction

Elicitation is a task that helps the customer to define what is requirement. For producing a consistent and complete set of security requirements elicitation method can be used. But it is not possible to complete this task without any problem. To complete this elicitation step, numerous problems need to be faced, like problem of scope, problems of understanding and problem of volatility. To help overcome these problems, the eliciting requirements activity were worked with in a well ordered and systematic manner.

3.2 Eliciting Requirements

During inception basic questions and answers establish the scope of the problem and the overall perception of a solution. Elicitation makes use of requirement elicitation formats that combines elements of problem solving, elaboration, negotiation, and specification. It requires the cooperation of a group of end-users and developers to elicit requirements. To elicit requirements these four works were completed:-

- Collaborative requirements gathering
- Quality function deployment
- Usage scenery
- Elicitation work products

3.2.1 Collaborative Requirements Gathering

The goal is to look out for the problem, propose elements of the solution, negotiate different approaches, and specify a preliminary set of solution requirements in an atmosphere that is conducive to the completion of the goal. Many different approaches to collaborative requirements gathering have been suggested. Each makes use of a faintly different scenario. The following steps were done in order to complete the project.

- The meetings were conducted with many doctor’s assistant and patients. They were questioned about their requirements and expectations from the system.

- They were asked about the complications they are facing with the current system.
- Every meeting was done carefully and eventually the final requirements from these meeting are selected.

3.2.2 Quality Function Deployment

Quality function deployment (QFD) is a quality management technique that translates the needs of the customer into technical requirements for software. QFD “concentrates on increasing the rate of customer satisfaction from the software engineering process”. With respect to our project the following requirements are identified by a QFD.

Normal Requirements

Normal requirements consist of objectives and goals that are stated during the meeting with the customers. Normal requirements of our project are:-

- Easy user interface.
- Accessible via the Internet.
- Allow user to search for items.
- Allow valid users to login and logout.
- Limit access to functionality of the system based upon user roles.

Expected Requirements

These requirements are implicit to the system and may be so fundamental that the customer does not explicitly state them .Their absence will be a cause for discontent:

- Provide appropriate hints and error message.
- Provide help guide for users.
- Avoid using textbox.
- Provide search facilities for patients and doctor’s assistants.

Exciting Requirements

These requirements are for features that exceed customer's expectations and prove to be very satisfying when presented.

1. Connect user account with Facebook or other social media
2. Provide advertisement facilities.

3.2.3 Usage Scenarios

People very often need medical tips and services. They have to visit the doctor's chamber. But the main problem, especially in any city is scheduling problem. Most often patients have to wait for hours to get services even if the appointments are scheduled. Thus, the patients suffer and they somehow end up blaming the doctors or the system.

Another problem is people very often do not know where doctors are available and when. Fraud people are found impersonating to be doctors and people are unable to get information of a doctor. So choosing doctors has become problematic for the people and popular Doctors are finding it difficult to manage their chambers and schedule appointments efficiently.

The web system 'Doctor's Appointment System' can be the solution of the difficulties faced by doctors and patients in our country.

Admin of the system will be added from the backend of the system. Admins can activate his/her account after activating the account using activation code sent to his mobile number and email.

Patients/users can sign up using his/her mobile number or email. He /she will need to activate his/her account using an activation code which will be sent to their mobile or email.

The assistant of the doctor will be provided with an online web interface. At first, he will have to register in the DAS using username, password and email address. The account will be created after verifying information and approval of the doctor.

Doctors can sign up using their mobile and email and other required field such as field of specialization and registration number. The system will not show any record of the doctor until he is verified by an admin of the system.

Any user who has an account which is activated can log into the system, system will recognize the user type and behave accordingly.

Once verified the doctor will set the scheduling configuration. He will set the time for each type of patient (not visited before, visited but not more than five times, more than five times) and how much fee he wants from each type of patient. He will also set what will happen if an appointment is cancelled by the patient, the doctor will have two options those are, he can let the latest requester of the same category to have the slot or he can choose to notify already scheduled appointees that a slot is open and they can request to take that slot. Doctor will set the time period and the days of the week he wants to attend the appointments. Once this is configured by the doctor the system will show the doctor's profile to the patients and he will also appear in the search list.

In case the scheduled appointment is rescheduled, notifications will be sent to the effected party via SMS/Email.

Appointees should be able to see the current serial number of the appointment that the doctor is attending and estimated time of his/her appointment.

Logged in patient or any anonymous user should be able to search for doctors by name or specialization of the doctors. The system should provide a list of doctors. User can click to see the profiles of the doctors and there logged in users will be able to request for an appointment.

3.2.3 Elicitation Work Products

The work products of the requirement elicitation can be different depending on the size of the system or product to be built. The work products include:

- A statement of requirements for our system.
- A bounded statement of scope for our system.
- A list of customer, users, and other stakeholders who participated in requirement elicitation.
- A detailed description of the system's technical environment.
- A set of usage scenarios.

Each of these work products are verified by all people who have participated in requirements elicitation.

3.3 Conclusion

All the necessary requirements to fulfill our project scope were identified. A generated scenario of the project has been developed. Based on this scenario, a scenario based model is presented in the next chapter.

Chapter 4

Scenario Based Modeling

4.1 Definition of Use case

This chapter describes the scenario-based modelling of DAS system. It will contain the use case diagrams, activity diagrams and swim lane diagrams.

- A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.
- The UML activity diagram supplements the use case by providing a graphical representation of the flow of interaction within a specific scenario.
- The UML swim lane diagram is a useful variation of the activity diagram and allows us to represent the flow of activities described by the use case and at the same time indicate with actor or analysis class has responsibility for the action described by an activity rectangle.

After completing the model, I can understand the problem space of the DAS project better. That’s why this modelling is called the first step towards problem space.

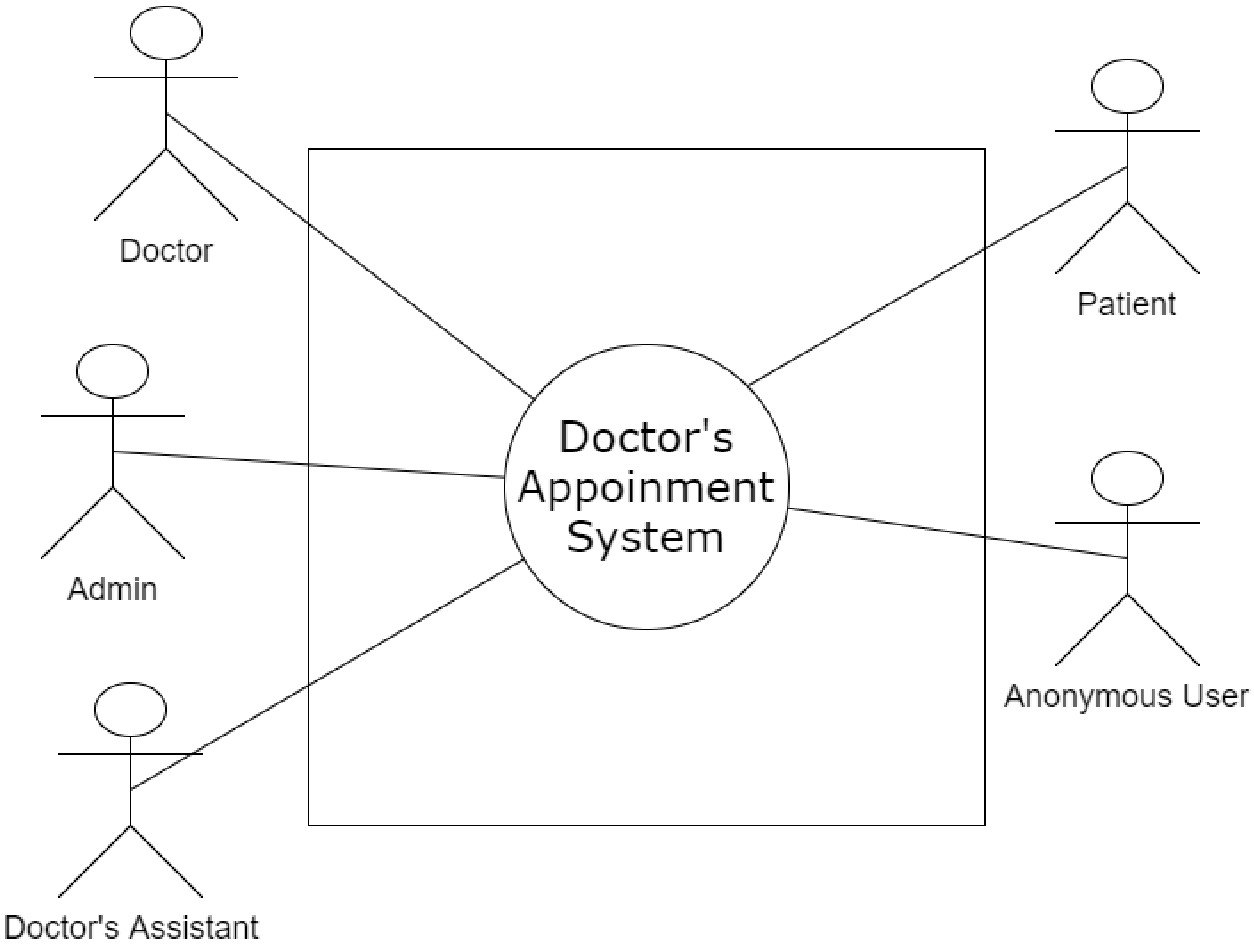

Figure 1: Level-0 Use Case Diagram

Usecase Name: Doctor’s Appointment System.

Usecase ID: 0

Primary Actors: Doctor, admin.

Goal in Context: To set an appointment.

Preconditions: User must have a web browser.

Trigger: The user decides to run the system to create account.

Exception: Null.

Frequency of use: Moderate frequency.

Channel to actors: Via Computer and android mobile.

Secondary actor: Assistant and Patient.

Actions of the actors: The actions of the actors will be discussed in level 1.

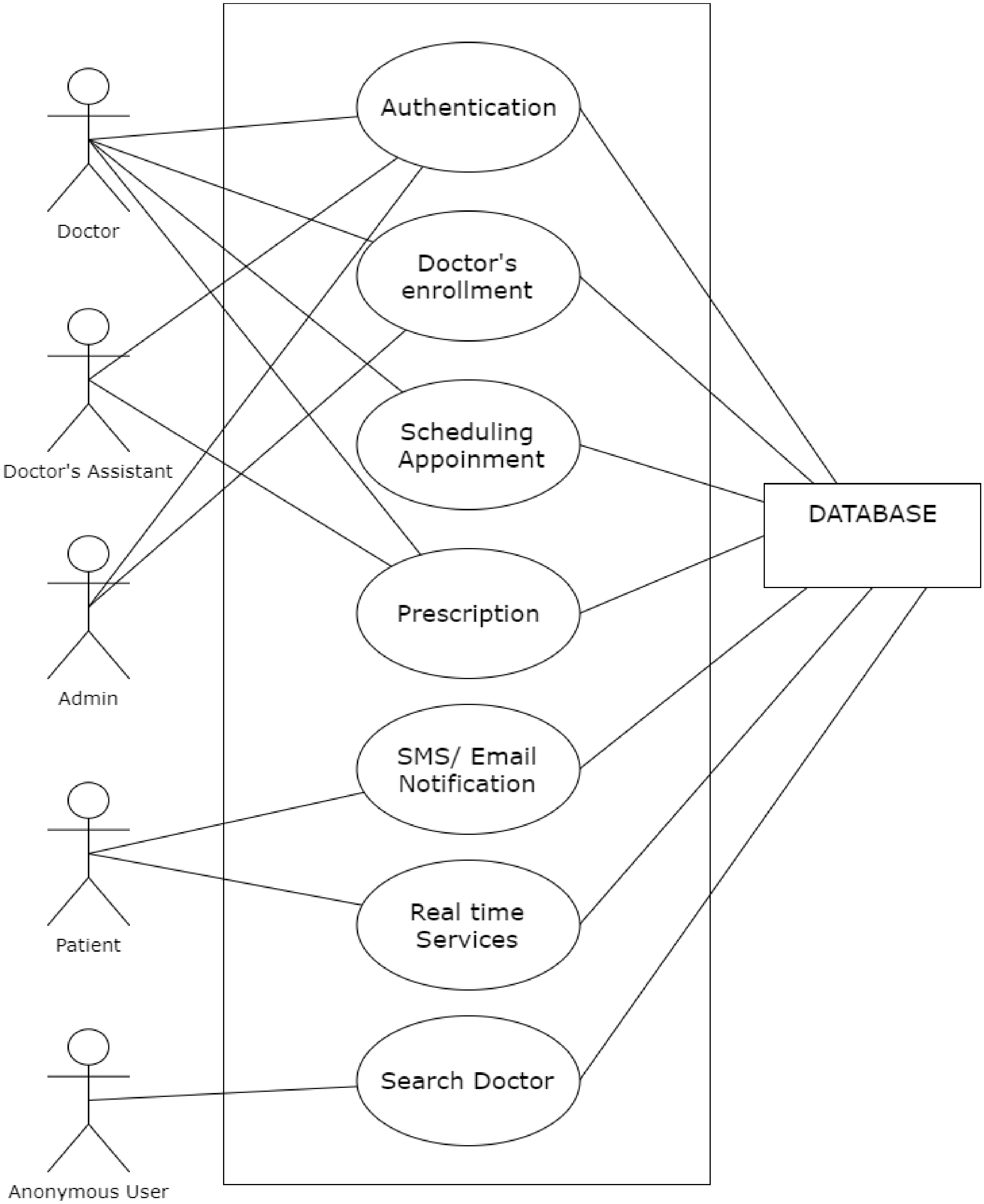

Figure 2: Use case diagram level 1

Usecase name: Doctor’s Appointment System.

Usecase ID: 0

Primary actors: Doctor, admin.

Goal in context: To set an appointment.

Preconditions: User must have a web browser.

Trigger: The user decides to run the system to create account.

Exception: Null.

Frequency of use: Moderate frequency.

Channel to actors: Via Computer and android mobile.

Secondary actor: Assistant and Patient.

Actions of the actors:

The actions of the actors are given below:

1. Authentication Subsystem (Level 1.1) includes the following actions

- Action: click on ‘Sign Up’ option.
  Reply: show account creation form.
- Action: fill up and submit form.
  Reply: account created/ not created.
- Action: Activate account.
  Reply: activated/not activated.
- Action: click on ‘Sign in/ sign out’ option.
  Reply: successful/ failed.

2. Doctor’s enrollment Subsystem (Level 1.2) includes the following actions

- Action: fill up form and submit.
  Reply: submitted/ not submitted.

- Action: Verify Doctor.
  Reply: Doctor’s verified/ not verified.

3. Scheduling appointment Subsystem (Level 1.3) includes the following actions

- Action: setting scheduling parameters.
  Reply: parameters settled/ not settled.
- Action: make request for appointment.
  Reply: request made/ not made.
- Action: check schedule for appointment request.
  Reply: checked/ not checked.
- Action: generate serial number.
  Reply: generated/ not generated.

4. ‘Prescription’ (1.4) subsystem includes the following actions

- Actions: Enter disease symptoms.

  Reply: entered / not entered.
- Actions: Enter Medicine and advice.

  Reply: entered / not entered.
- Action: click on ‘Print’ option.
  Reply: prescription printed/not printed

5. ‘SMS/Email notification’ (1.5) subsystem includes the following actions

- Actions: Click cancel Appointment

Reply: Appointment canceled and notification sent via SMS and Email/ not sent

- Action: send message when doctor is unavailable.
  Reply: sent/ not sent.

6. The subsystem ‘Real Time Services subsystem’ (1.6) includes the following actions

- Actions: Click ‘Chamber Status’ Appointment

  Reply: Show current serial number doctor attending/ not showed

- Actions: Click ‘estimated appointment time’

  Reply: Showed/ not showed

7. Search Subsystem (Level 1.7) includes the following actions

- Action: search for Doctor’s profile.
  Reply: showed/ not showed.
- Action: search for Patient’s schedule.
  Reply: showed/ not showed.

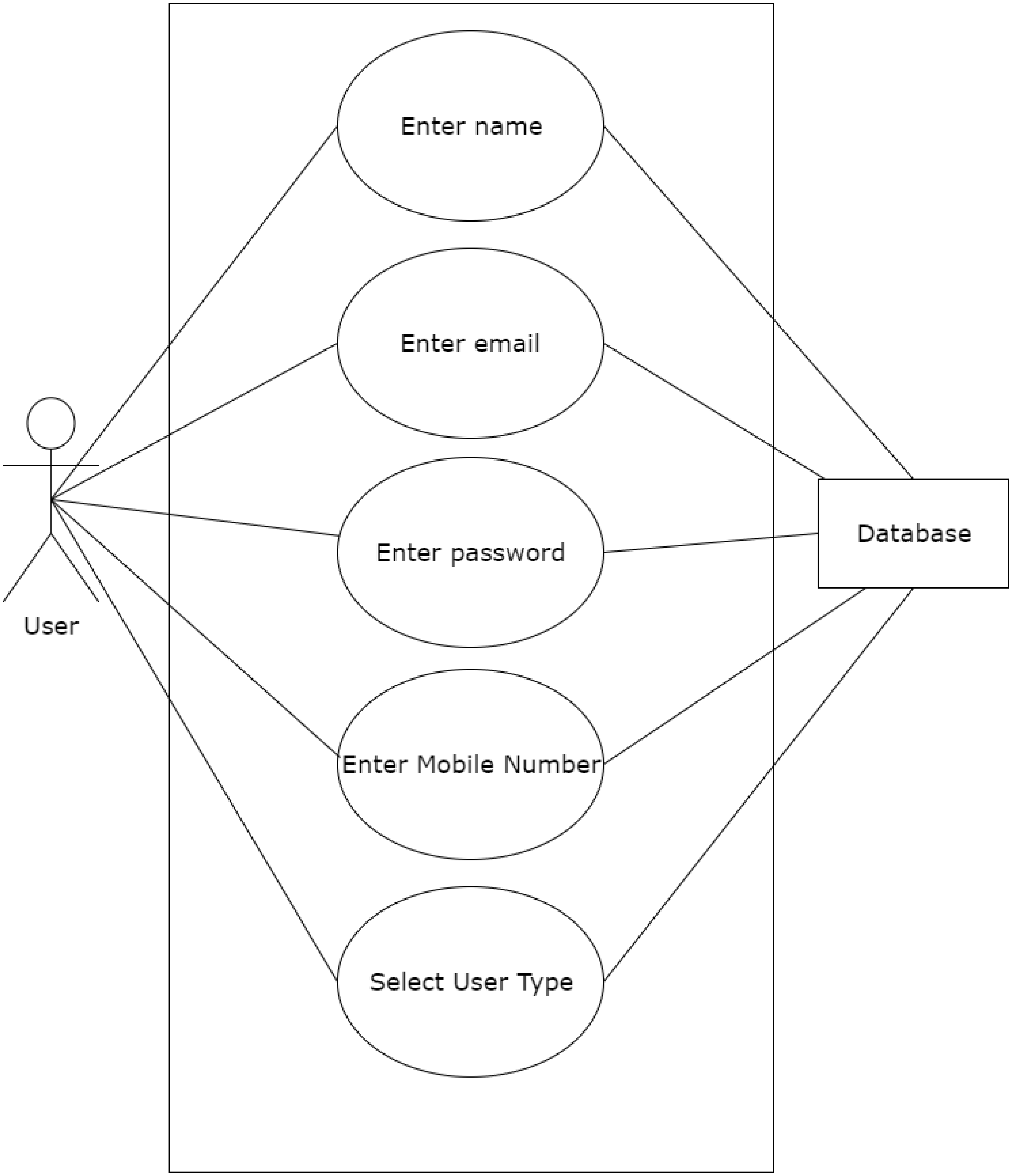

Figure 3: Use case diagram level 1.1.1 (Sign up)

Usecase Name: Authentication subsystem (Sign up)

Usecase ID: 1.1.1

Primary actor: Assistant, Patient

Goal in context: To manage account.

Preconditions: Internet connection.

Trigger: The Users decides to create account.

Exception: Internet connection failure during account creation process.

Priority: Essential.

Frequency of use: Only once for one user.

Channel to actor: Via internet connection.

Secondary Actor: Null.

Actions of the actors:

The actions of the actors Ire described in level 1.

Scenario for level 1.1.1:

Patients/users can sign up using his/her mobile number or email. He /she will need to activate his/her account using an activation code which will be sent to their mobile or email.

The assistant of the doctor will be provided with an online web interface. At first, he will have to register in the DAS using username, password and email address. The account will be created after verifying information and approval of the doctor.

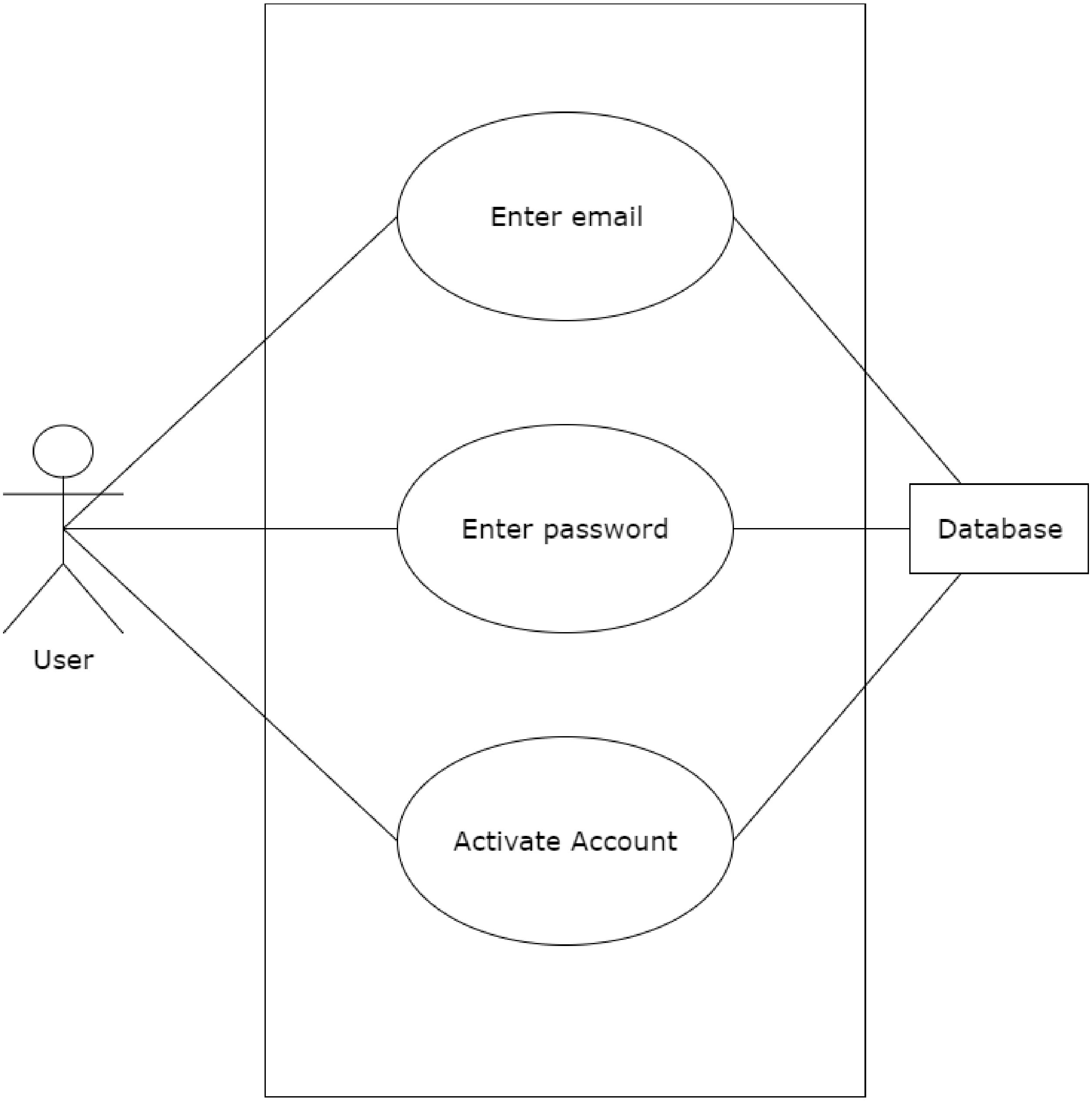

Figure 4: Use case diagram 1.1.2 (Sign in)

Usecase Name: Authentication subsystem (Sign in)

Usecase ID: 1.1.1

Primary Actor: Assistant, Patient, Doctor

Goal in Context: To manage account.

Preconditions: Internet connection.

Trigger: The Users decides to create account.

Exception: Internet connection failure during account creation process.

Priority: Essential.

Frequency of use: Only once for one user.

Channel to actor: Via internet connection.

Secondary Actor: admin

Actions of the actors:

The actions of the actors Ire described in level 1.

Scenario for use case level 1.1.2:

Any user who has an account which is activated can log into the system, system will recognize the user type and behave accordingly.

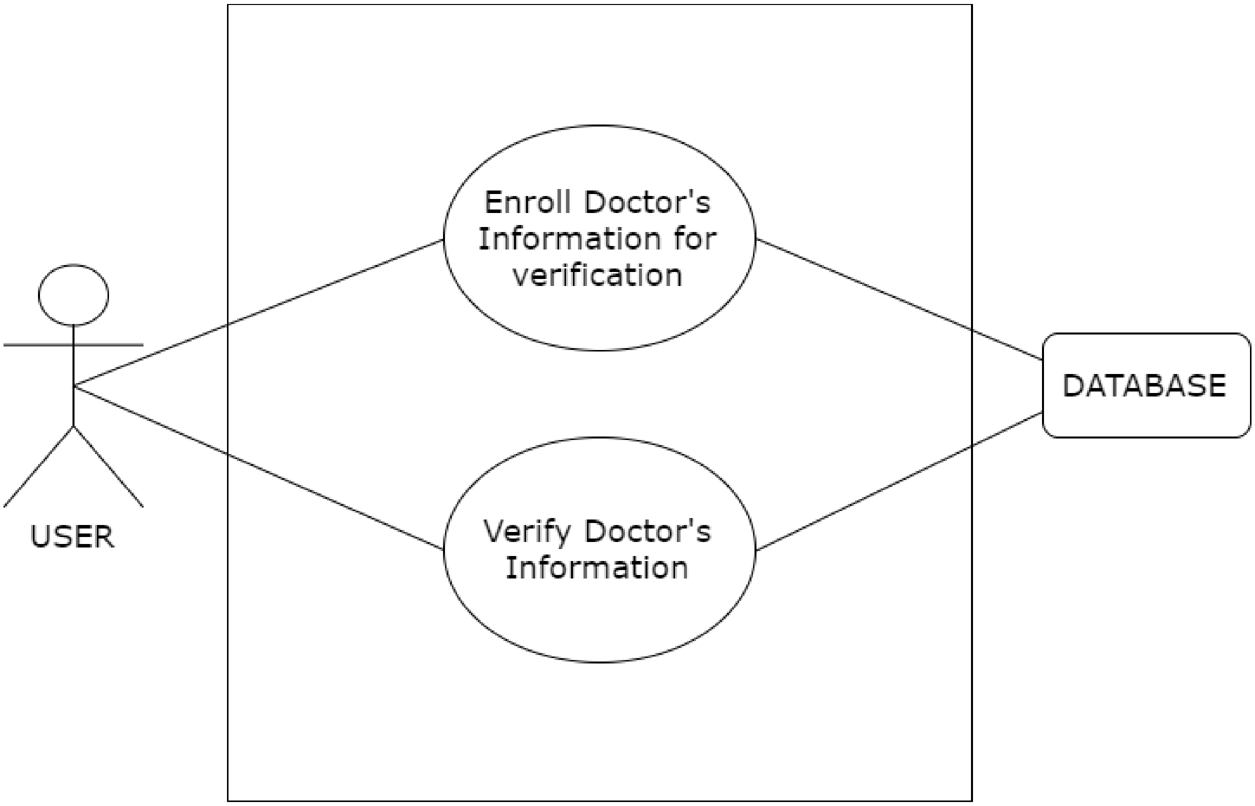
Enroll Doctor's Information for verification
Verify Doctor's Information
USER
DATABASE

Figure 4: Use case diagram level 1.2 (Doctor Enrollment)

Usecase Name: Doctor Enrollment subsystem

Usecase ID: 1.2

Primary Actor: Doctor

Goal in Context: To verify doctor and enroll them to the system

Preconditions: Internet connection.

Trigger: Doctor opens an account

Exception: Internet connection failure during account creation process.

Priority: Essential.

Frequency of use: Only once for one user.

Channel to actor: Via internet connection.

Secondary Actor: admin

Actions of the actors:

The actions of the actors Ire described in level 1.

Scenario for use case level 1.2:

Doctors can sign up using their mobile and email and other required field like field of specialization, registration no. The system will not show any record of the doctor until the he is verified by an admin of the system.

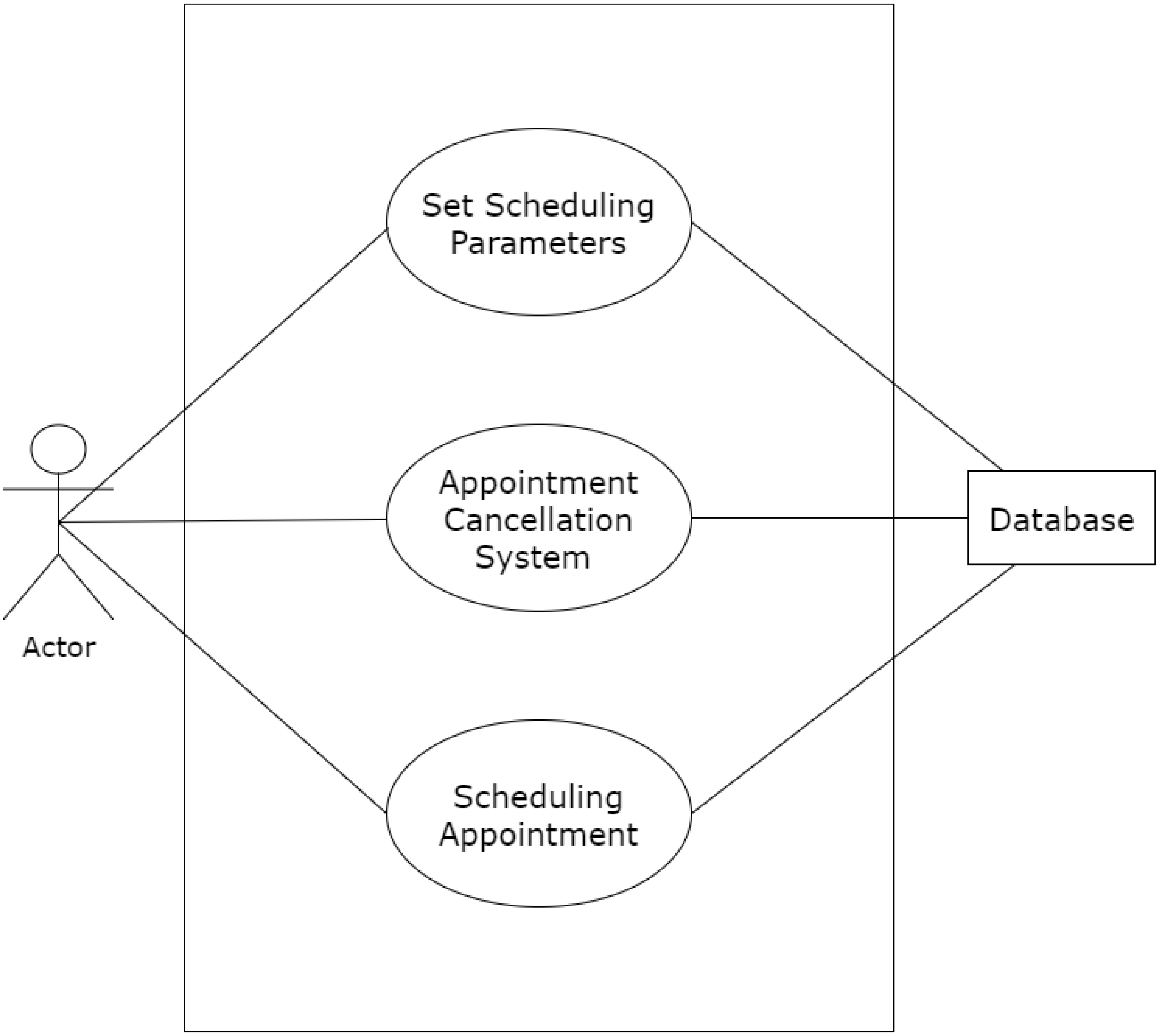

Figure 5: Use case diagram level 1.3 (Scheduling Appointment)

Usecase Name: Scheduling Appointment subsystem

Usecase ID: 1.3

Primary Actor: Doctor

Goal in Context: To schedule the appointment

Preconditions: Internet connection.

Trigger: Doctor opens an account

Exception: Internet connection failure, system failure

Priority: Essential.

Frequency of use: Only once for one user.

Channel to actor: Via internet connection.

Secondary Actor: null

Actions of the actors:

The actions of the actors Ire described in level 1.

Scenario for level use case 1.3:

Once verified the doctor will set the scheduling configuration. He will set the time for each type of patient (not visited before, visited but not more than five times, more than five times) and how much fee he wants from each type of patient. He will also set what will happen if an appointment is cancelled by the patient. The doctor will have two options which are, he can let the latest requester of the same category to have the slot or he can choose to notify already scheduled appointees that a slot is open and they can request to take that slot. Doctor will set the time period and the days of the week he wants to attend the appointments.

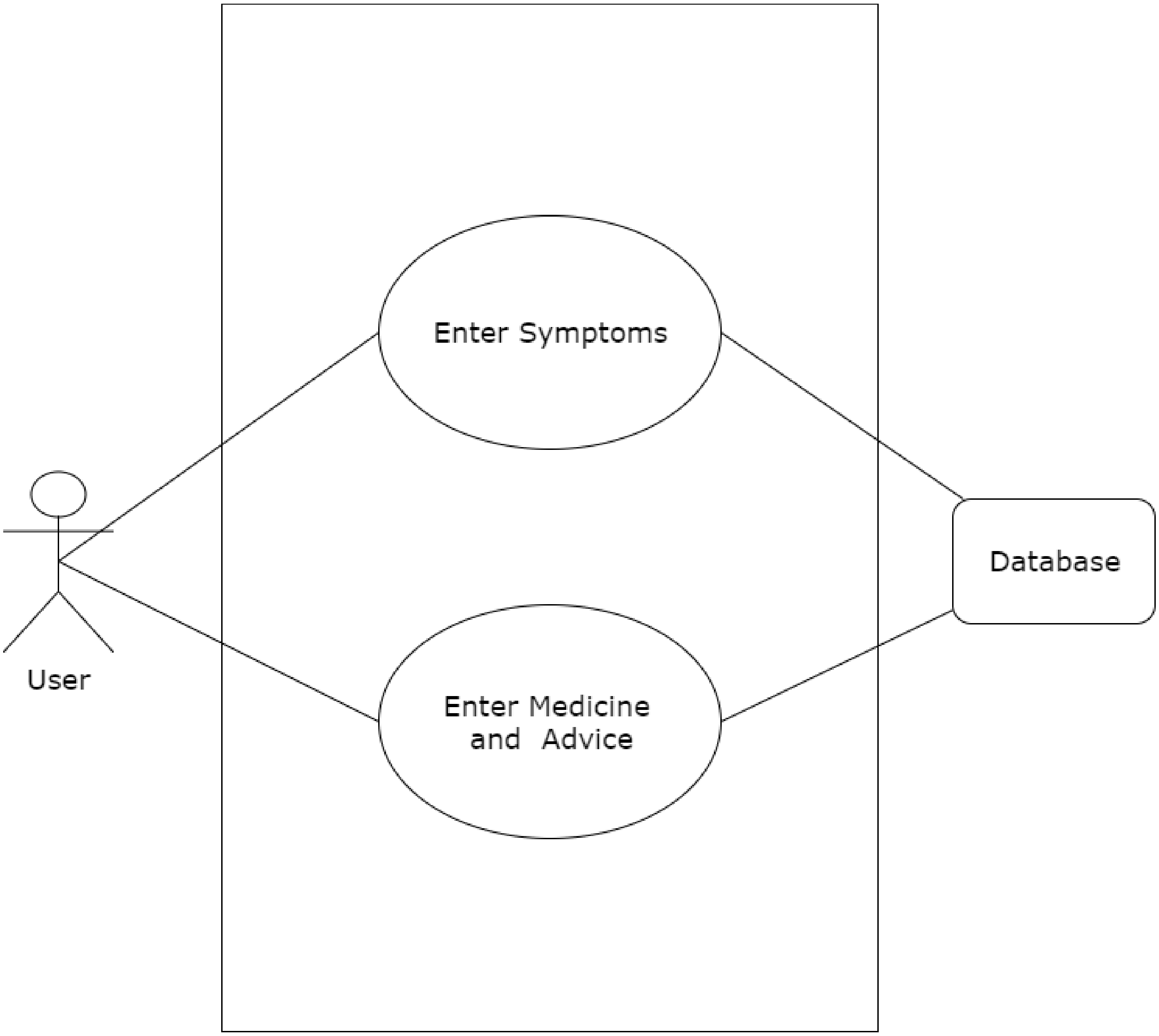

Figure 6: Use case diagram level 1.4 (Prescription)

Usecase Name: Prescription

Usecase ID: 1.4

Primary Actor: Doctor, Assistant

Goal in Context: attend the appointment

Preconditions: Inter

Trigger: Doctor opens an account

Exception: Internet connection failure, system failure

Priority: Essential.

Frequency of use: Only once for one appointment.

Channel to actor: Via internet connection.

Secondary Actor: patient

Actions of the actors:

The actions of the actors Ire described in level 1.

Scenario for use level 1.4:

Once the patient enters the room the assistant or the doctor will press the start button which will start the appointment. Doctor or assistant both will be able to enter the medicines and the advices, they can also enter the symptoms of the patients which can be valuable information for the future treatment. At the end of the appointment finish button should be pressed which will tell the system the appointment has been attended.

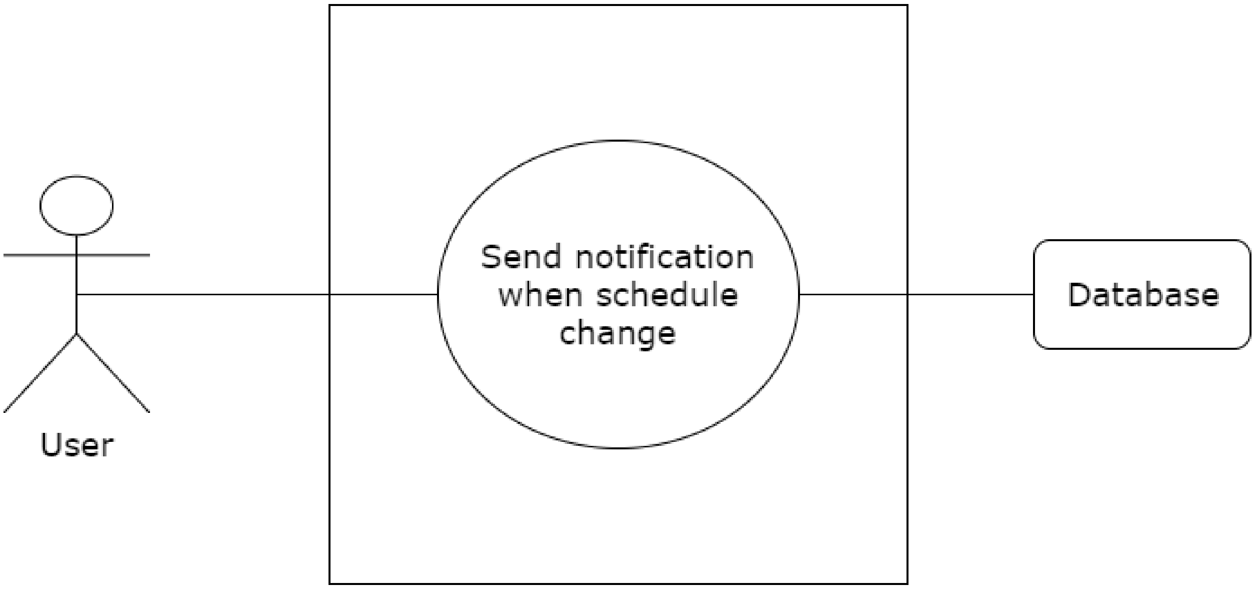

Figure 7: Use case diagram level 1.5 (SMS/Email Notification)

Usecase Name: SMS/Email Notification

Usecase ID: 1.5

Primary Actor: null

Goal in Context: attend the appointment

Preconditions: Inter

Trigger: Rescheduling of the appointment

Exception: Internet connection failure, system failure

Priority: Essential.

Frequency of use: any time

Channel to actor: Via internet connection.

Secondary Actor: patient, doctor

Actions of the actors:

The actions of the actors Ire described in level 1.

Scenario for use level 1.5:

In case the scheduled appointment is rescheduled, notifications will be sent to the effected party via SMS/Email.

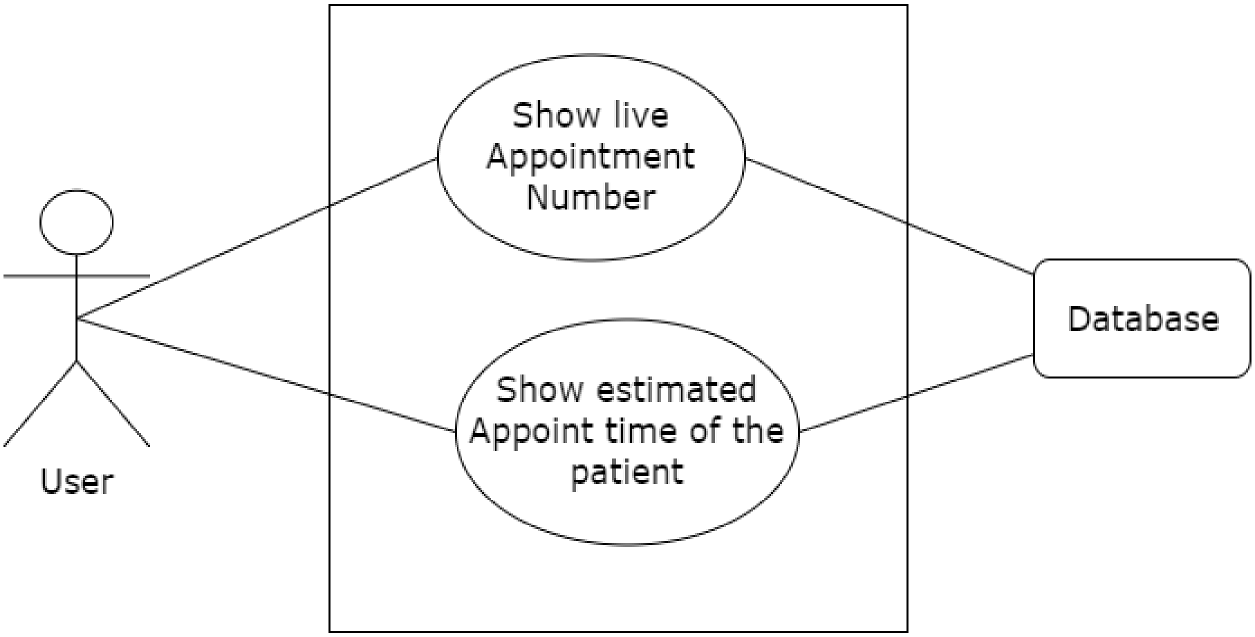
Show live Appointment Number
Show estimated Appoint time of the patient
User
Database

Figure 8: Use case diagram 1.6 (Real Time Services)

Usecase Name: real time services

Usecase ID: 1.6

Primary Actor: null

Goal in Context: providing smooth services to the patients

Preconditions: Internet

Trigger: Patient logs in and goes to appointments

Exception: Internet connection failure, system failure

Priority: essential

Frequency of use: any time

Channel to actor: Via internet connection.

Secondary Actor: patient

Actions of the actors:

The actions of the actors Ire described in level 1.

Scenario for use case level 1.6:

Appointees should be able to see the current serial number of the appointment that the doctor is attending and estimated time of his/her appointment.

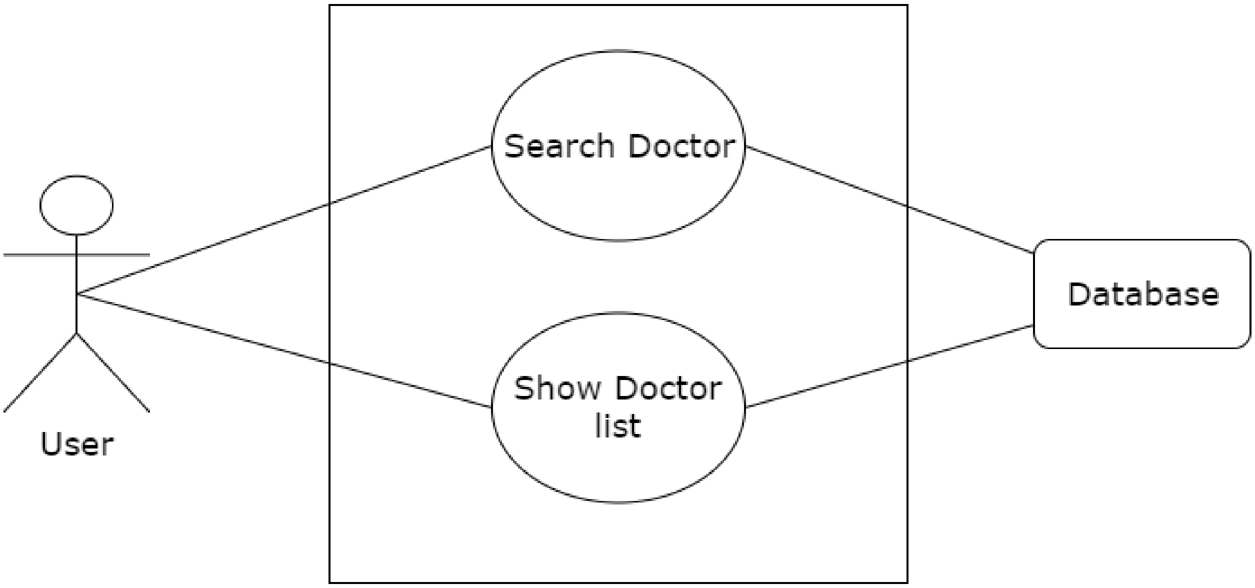

Figure 9: Use case diagram level 1.7 (Search Doctor)

Usecase Name: Search doctors

Usecase ID: 1.7

Primary Actor: patient, anonymous user

Goal in Context: help finding doctors

Preconditions: Internet

Trigger: Patient logs in and goes to appointments

Exception: Internet connection failure, system failure

Priority: essential

Frequency of use: any time

Channel to actor: Via internet connection.

Secondary Actor: patient

Actions of the actors:

The actions of the actors Ire described in level 1.

Scenario for use case level 1.7:

Logged in patient or any anonymous user should be able to search for doctors by name or specialization of the doctors. The system should provide a list of doctors. User can click to see the profiles of the doctors and there logged in users will be able to request for an appointment.

4.2 Activity and Swim lane

Activity diagram shows the technical view of the system for every use case from which I can understand how the system actually works and how the actors interact with the system. Here is the activity and swim lane diagrams for use the use cases stated before.

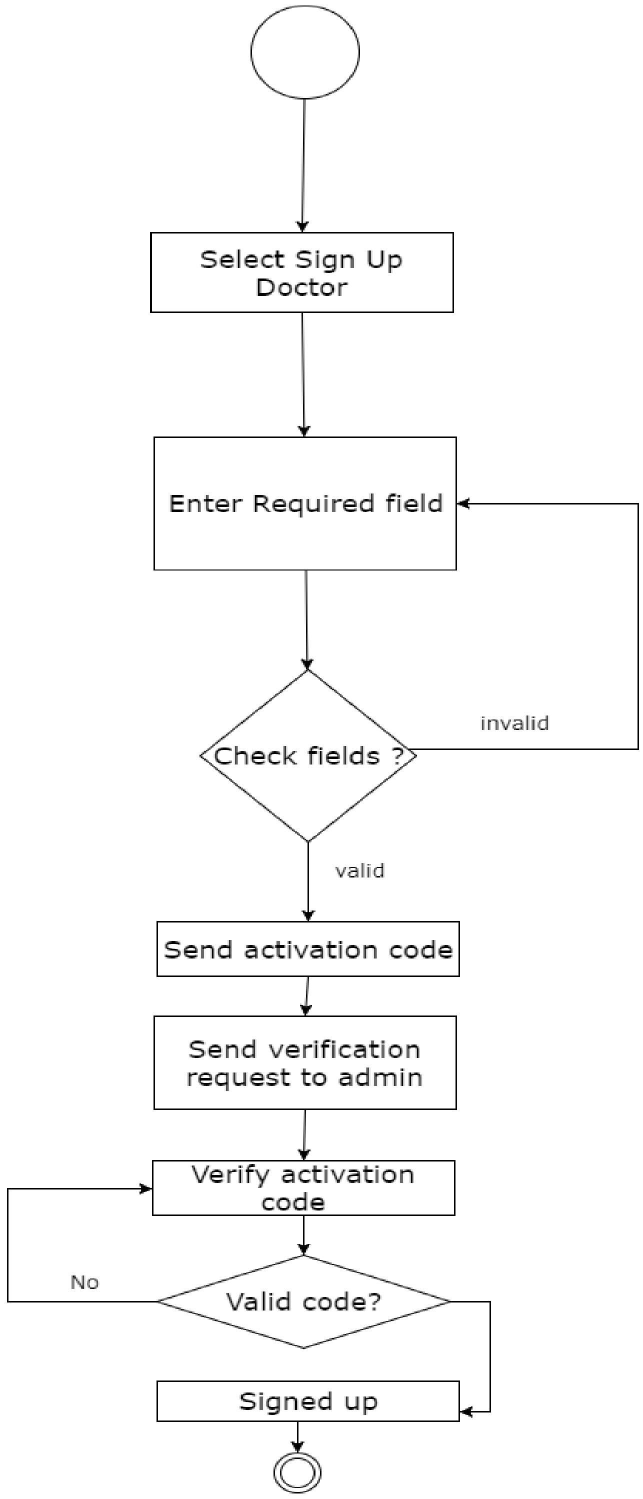

Figure 10: Activity Diagram of use case level 1.1.1 (Doctor’s sign up)

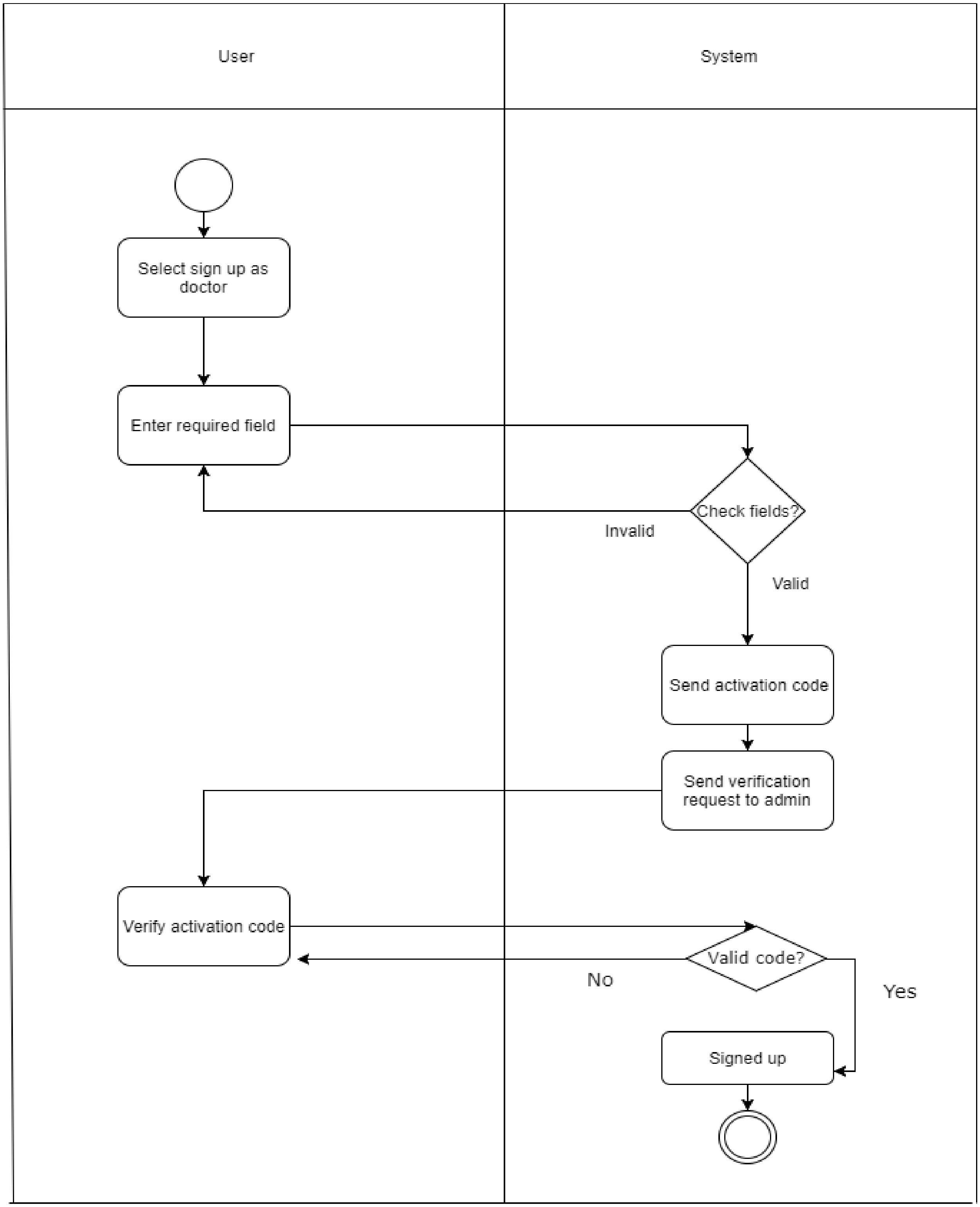

Figure 11: Swim lane diagram of level 1.1.1 (Doctor's sign up)

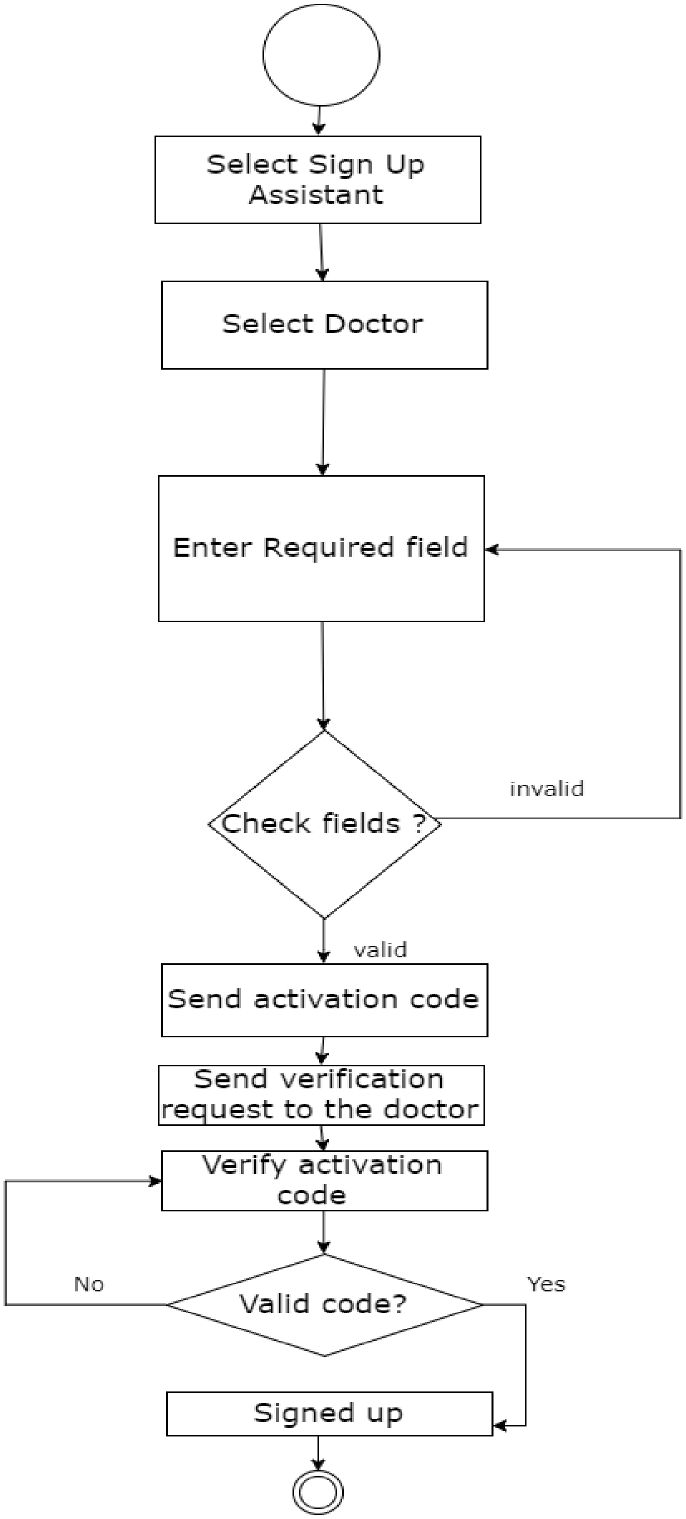

Figure 12: Activity Diagram of use case level 1.1.1 (Assistant's sign up)

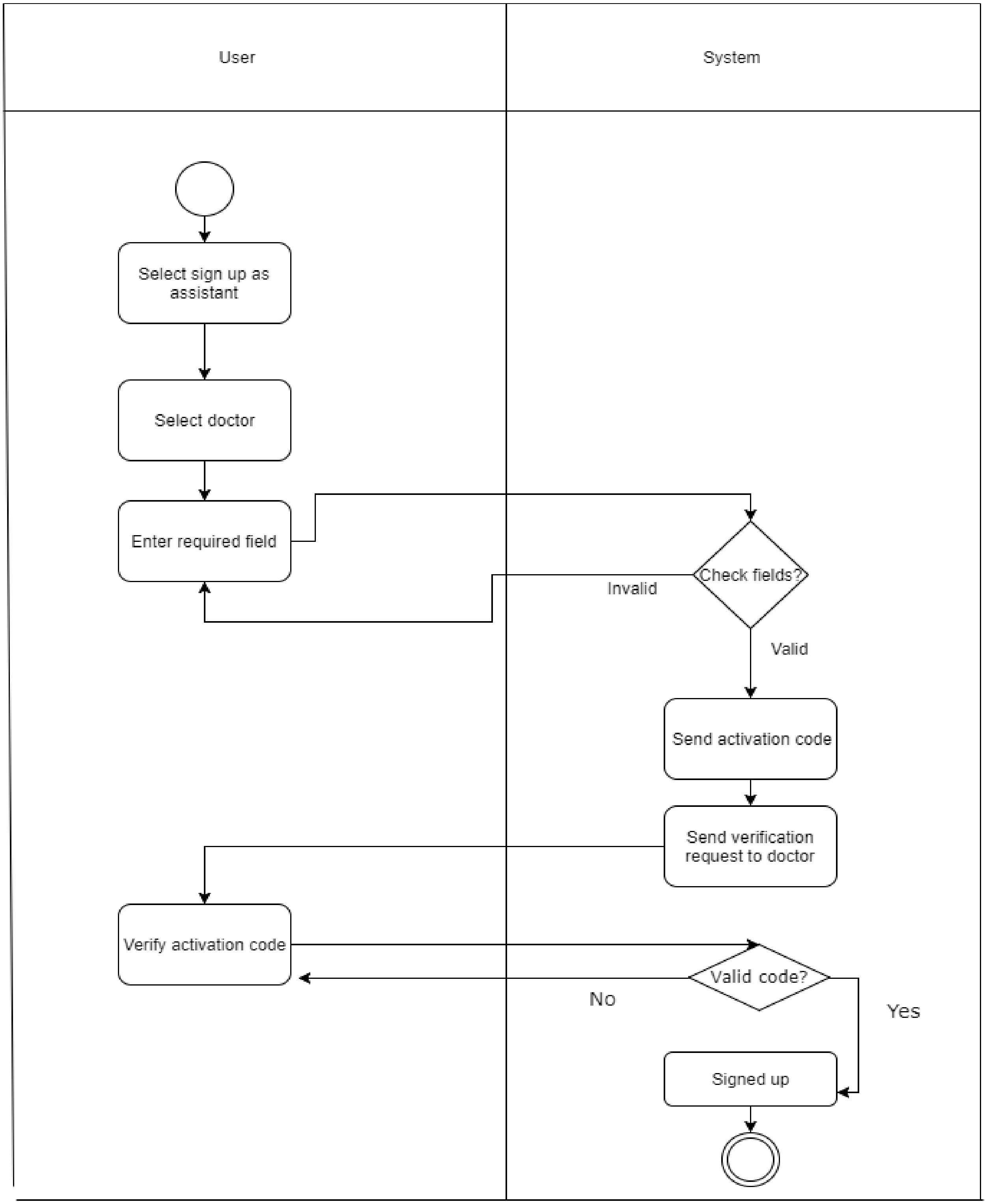

Figure 13: Swim lane diagram of use case level 1.1.1(Assistant's sign up)

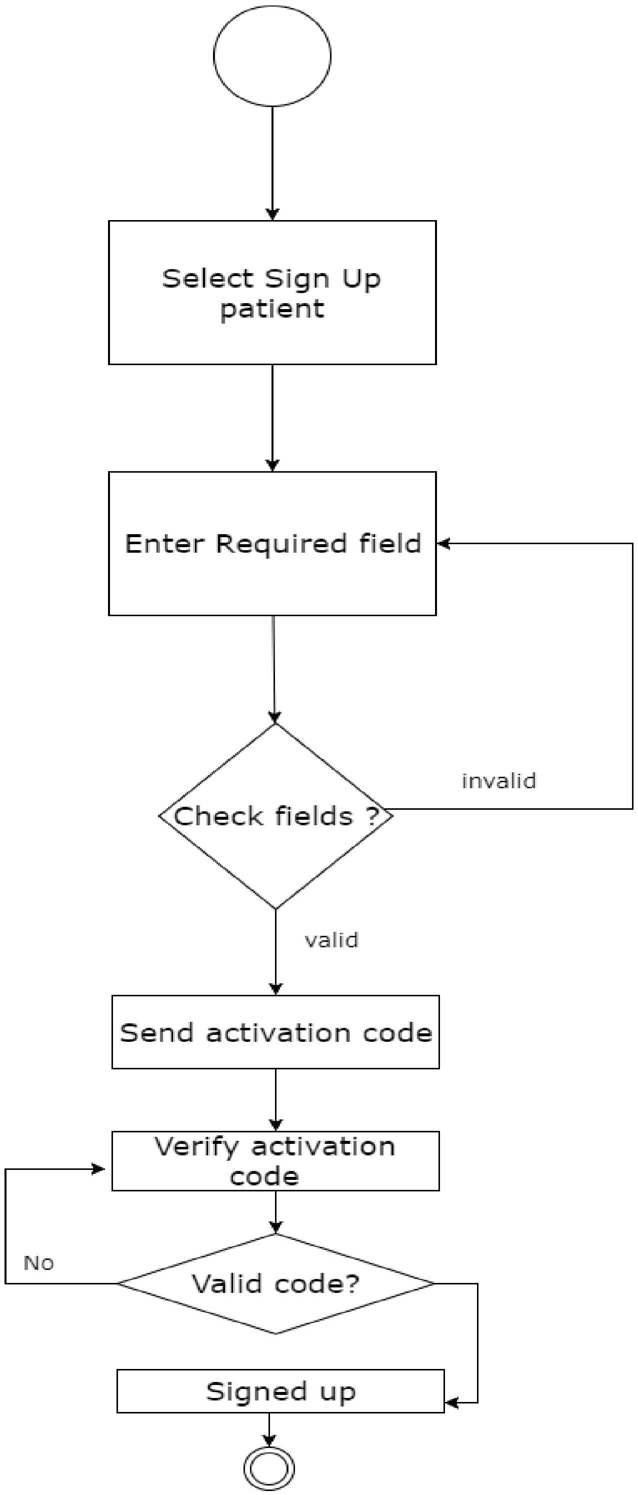

Figure 14: Activity diagram of use case level 1.1.1 (Patient's sign up)

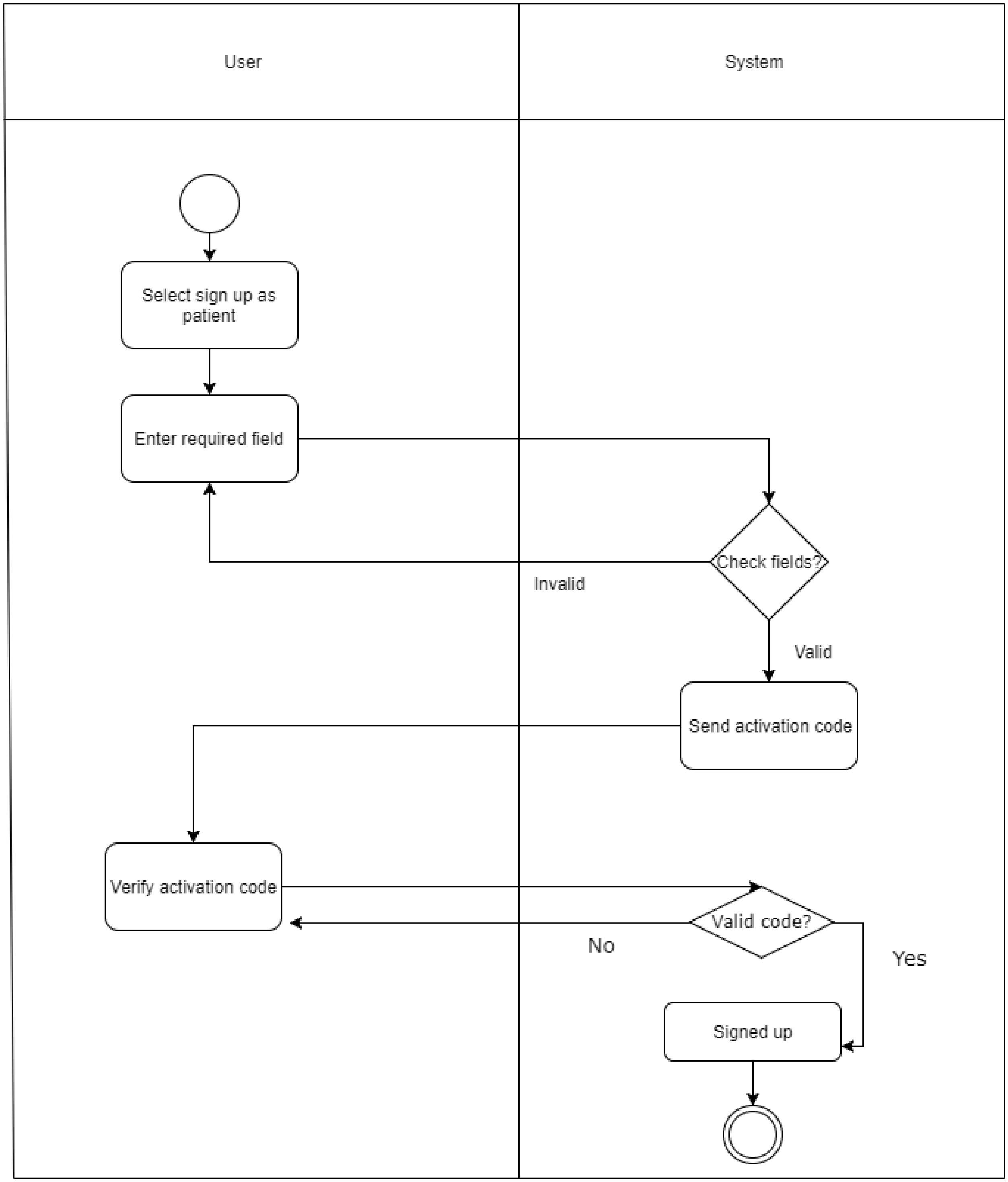

Figure 15: Swim lane diagram of use case level 1.1.1 (Patient's sign up)

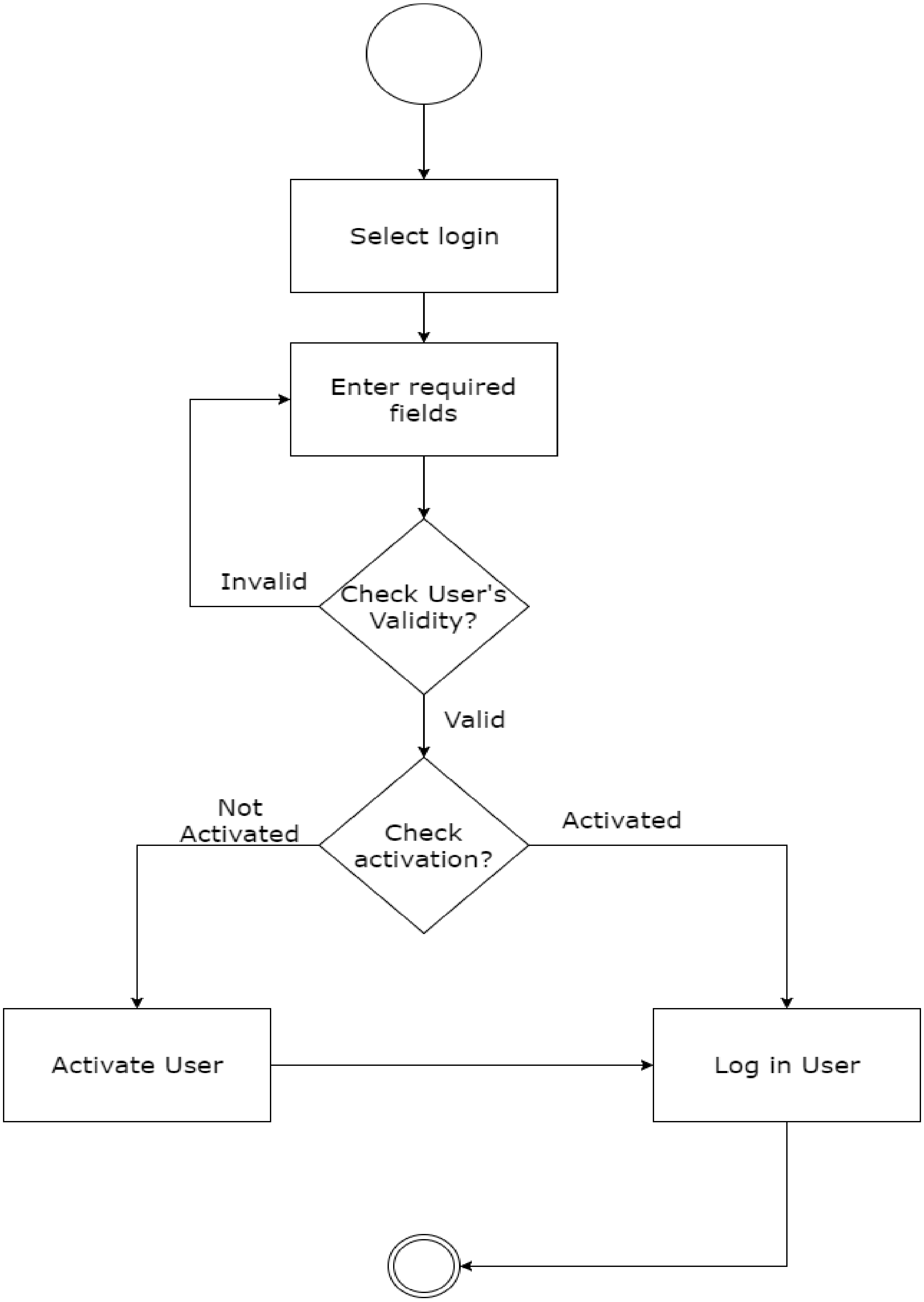

Figure 16: Activity Diagram of use case level 1.1.2 (Sign In)

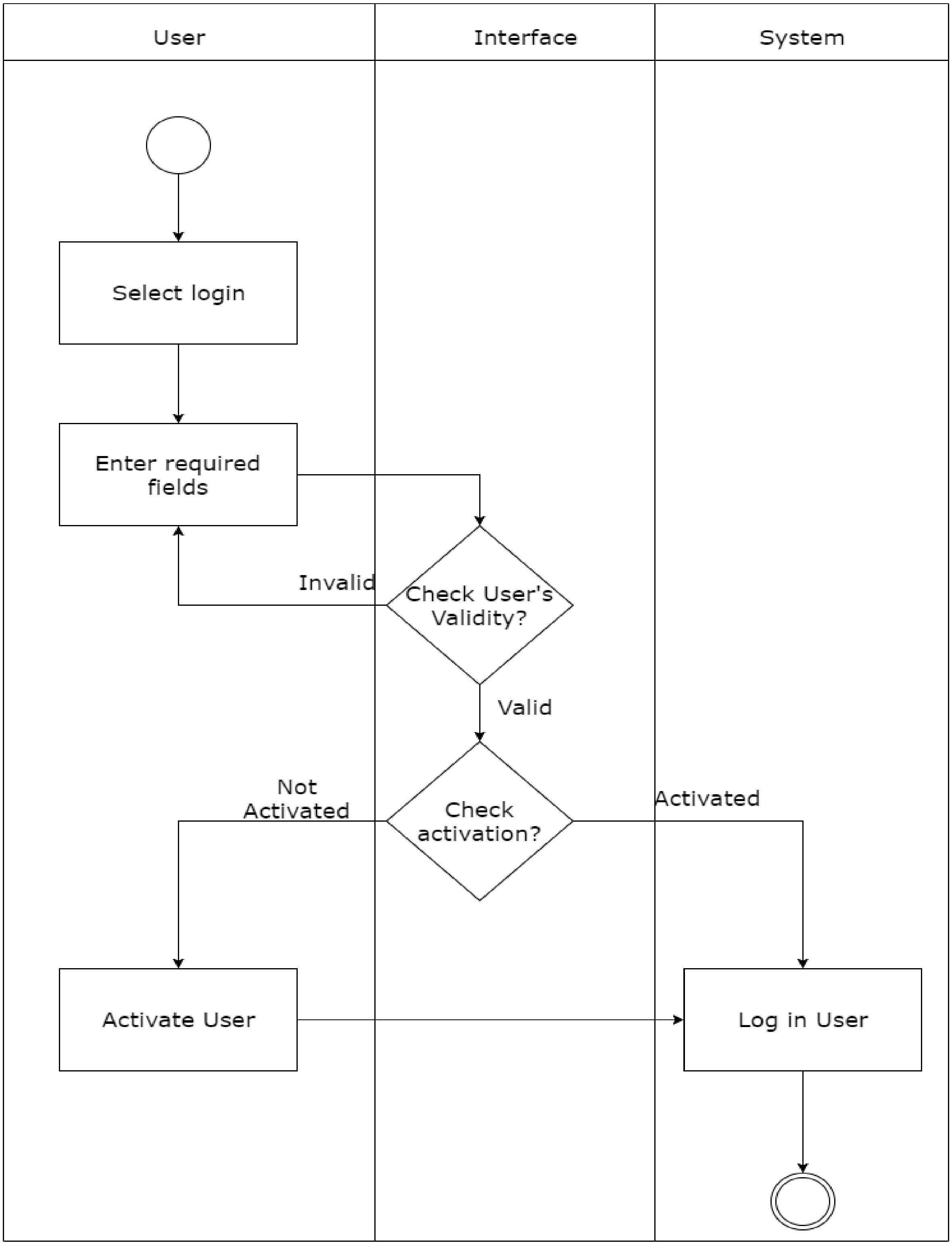

Figure 17: Swim lane diagram of use case level 1.1.2 (sign in)

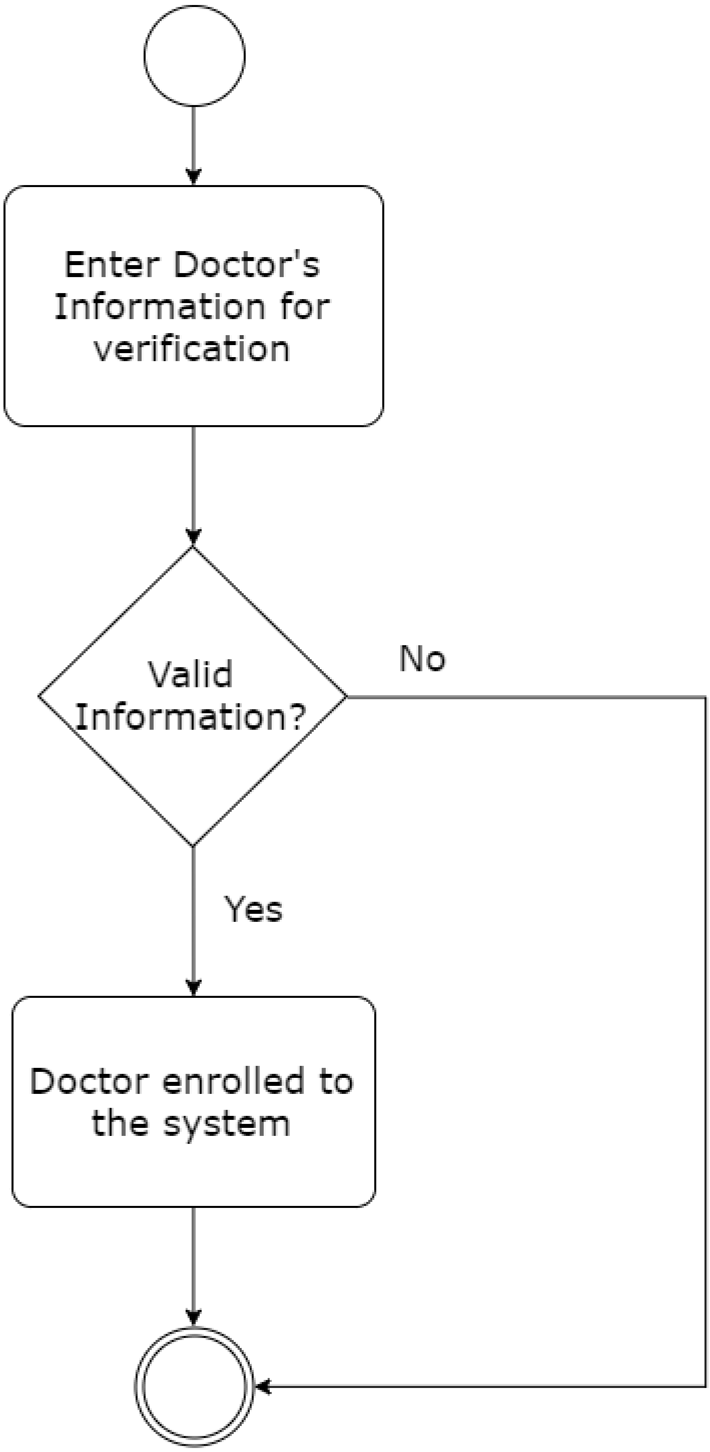
Enter Doctor's Information for verification
Valid Information?
No
Yes
Doctor enrolled to the system

Figure 18: Activity Diagram of use case level 1.2 (Doctor's enrollment)

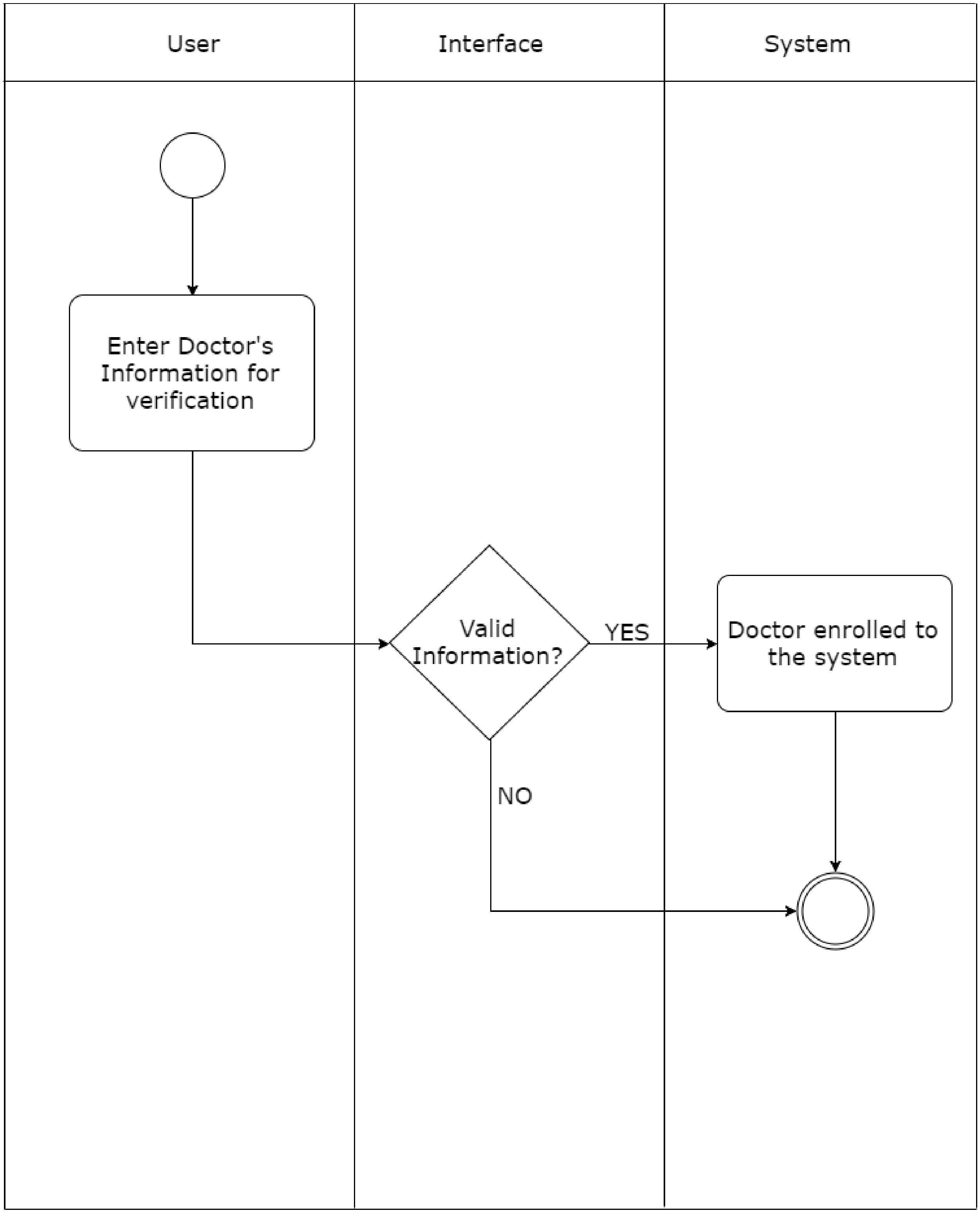
User
Interface
System
Enter Doctor's Information for verification
Valid Information?
YES
NO
Doctor enrolled to the system

Figure 19: Swim lane Diagram of use case level 1.2 (Doctor's enrollment)

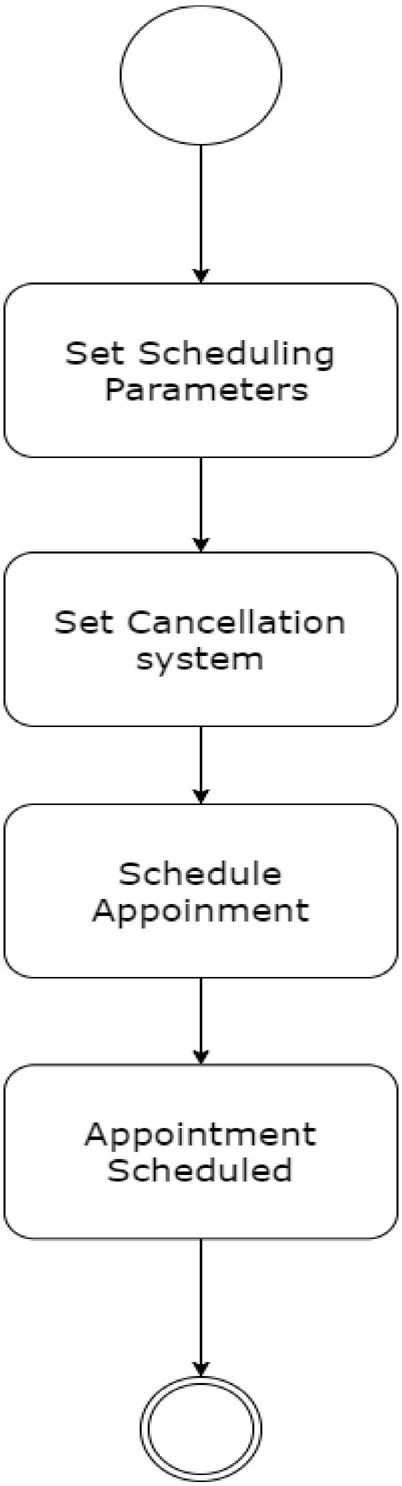

Figure 20: Activity diagram of use case level 1.3 (Scheduling appointment)

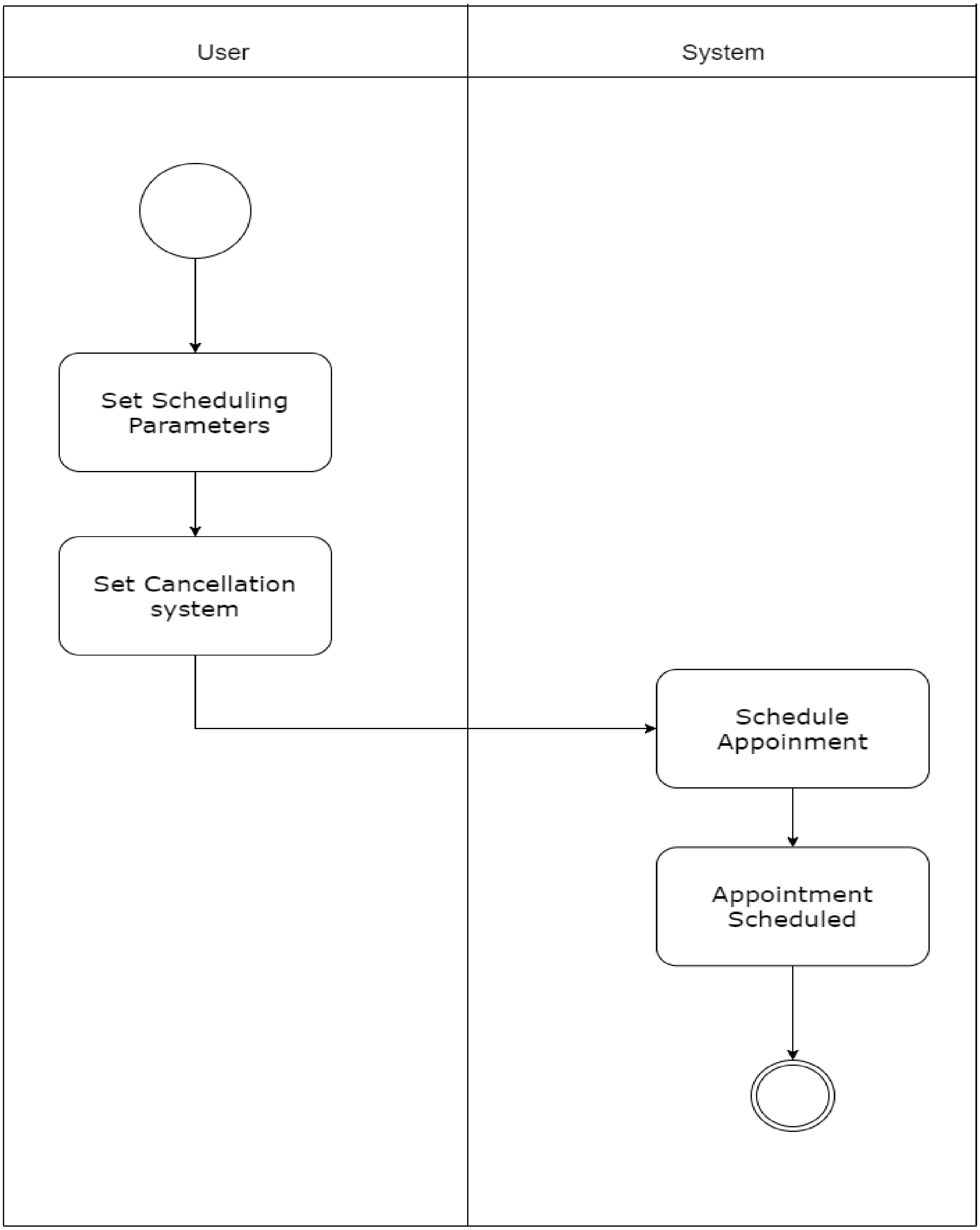

Figure 21: Swim lane diagram of use case level 1.3 (Scheduling appointment)

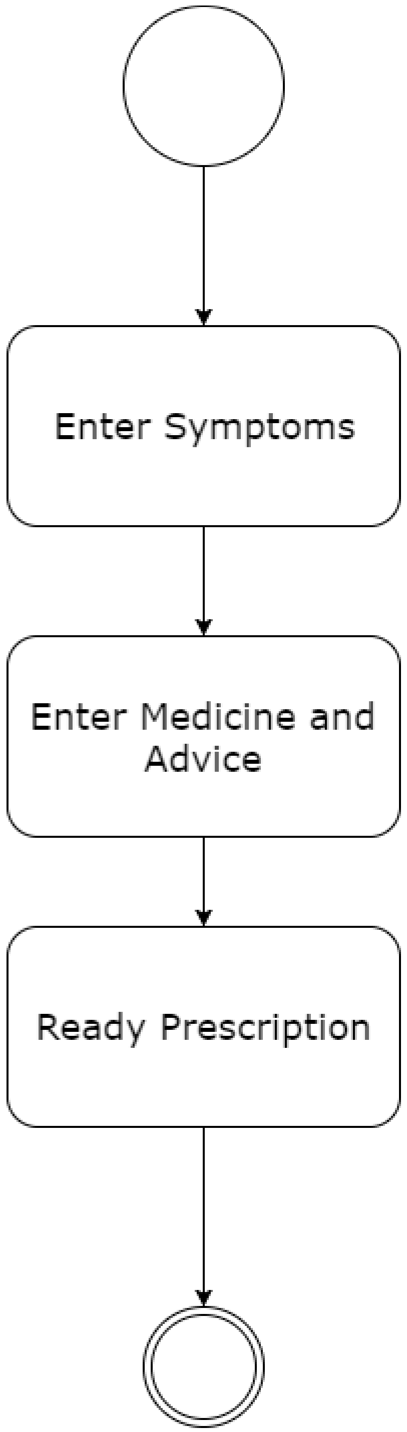

Figure 22: Activity diagram of use case level 1.4 (Prescription)

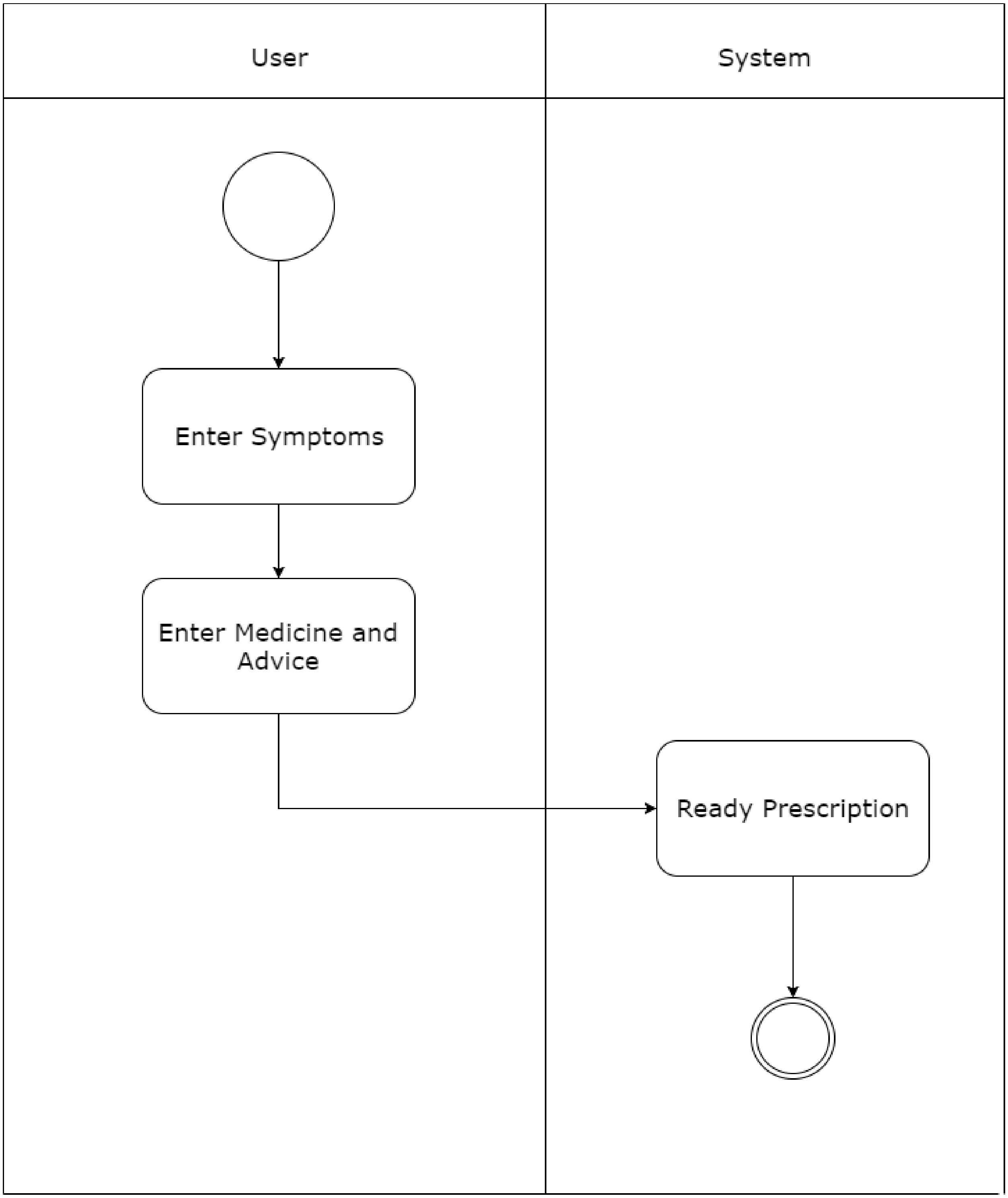

Figure 23: Swim lane diagram of use case level 1.4 (prescription)

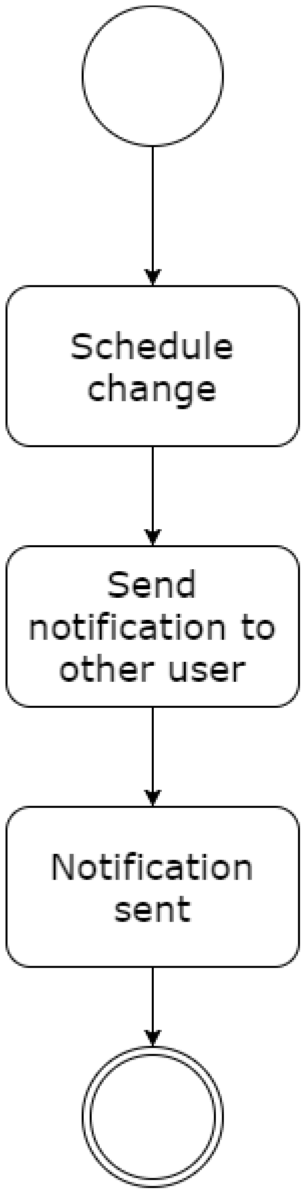

Figure 24: Activity diagram of use case level 1.5 (SMS/Email notification)

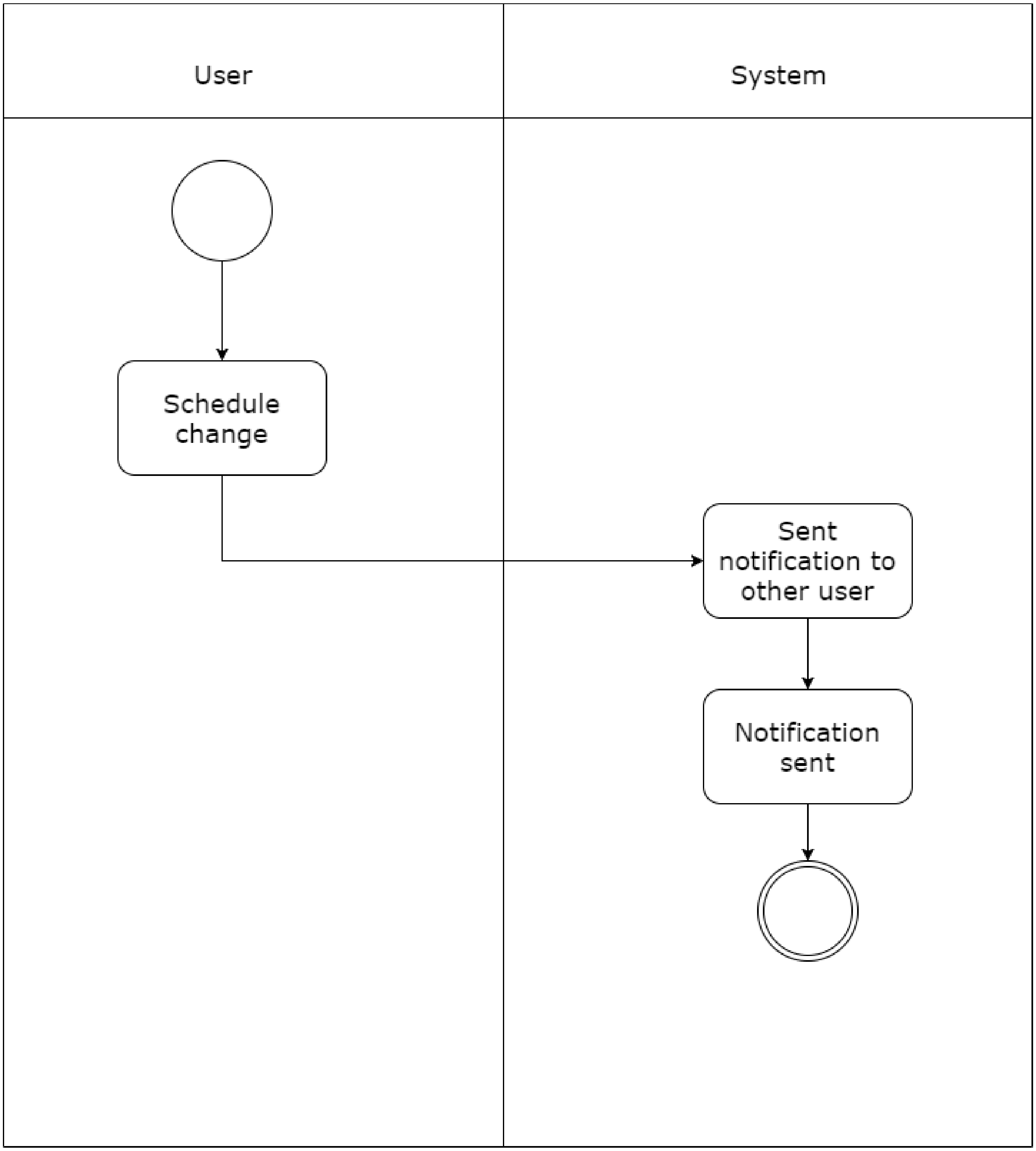

Figure 25: Swim lane diagram of use case level 1.5 (SMS/Email Notification)

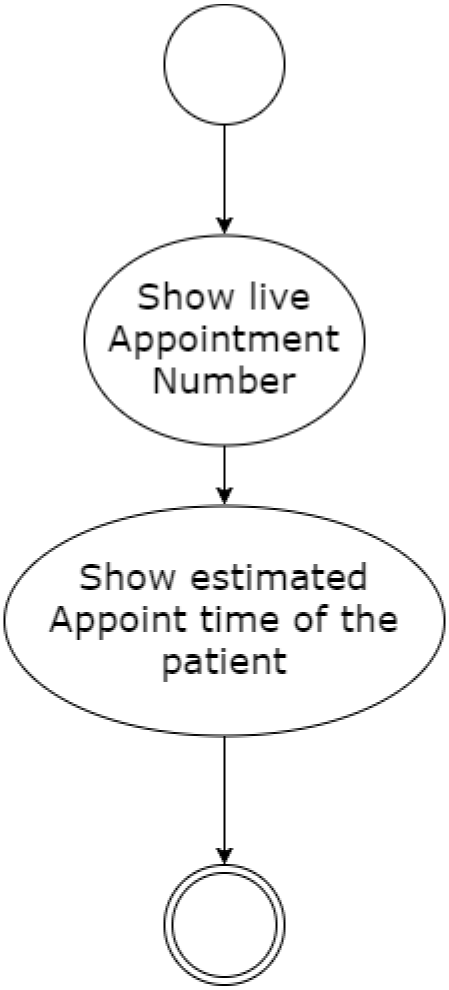
Show live Appointment Number
Show estimated Appoint time of the patient

Figure 26: Activity diagram of use case level 1.6 (real time services)

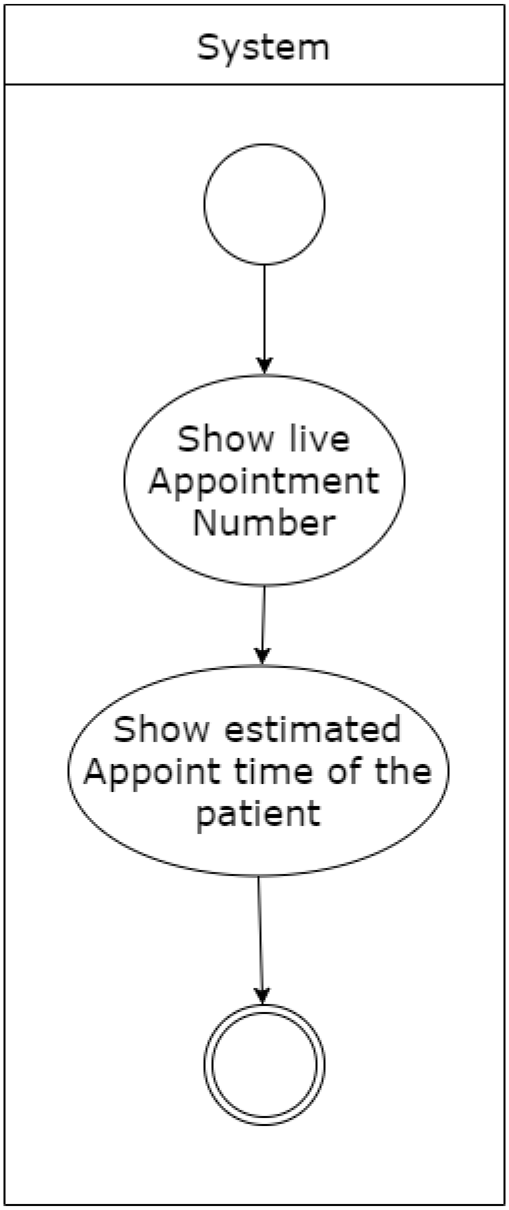
System
Show live Appointment Number
Show estimated Appoint time of the patient

Figure 27: Swim lane diagram of use case 1.6 (real time services)

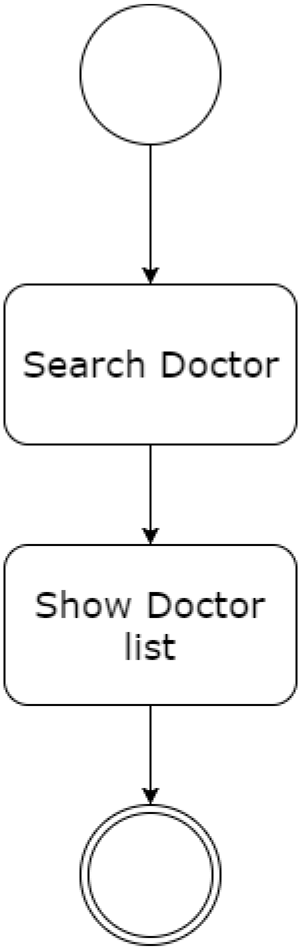

Figure 28: Activity diagram of use case level 1.7 (Search doctor)

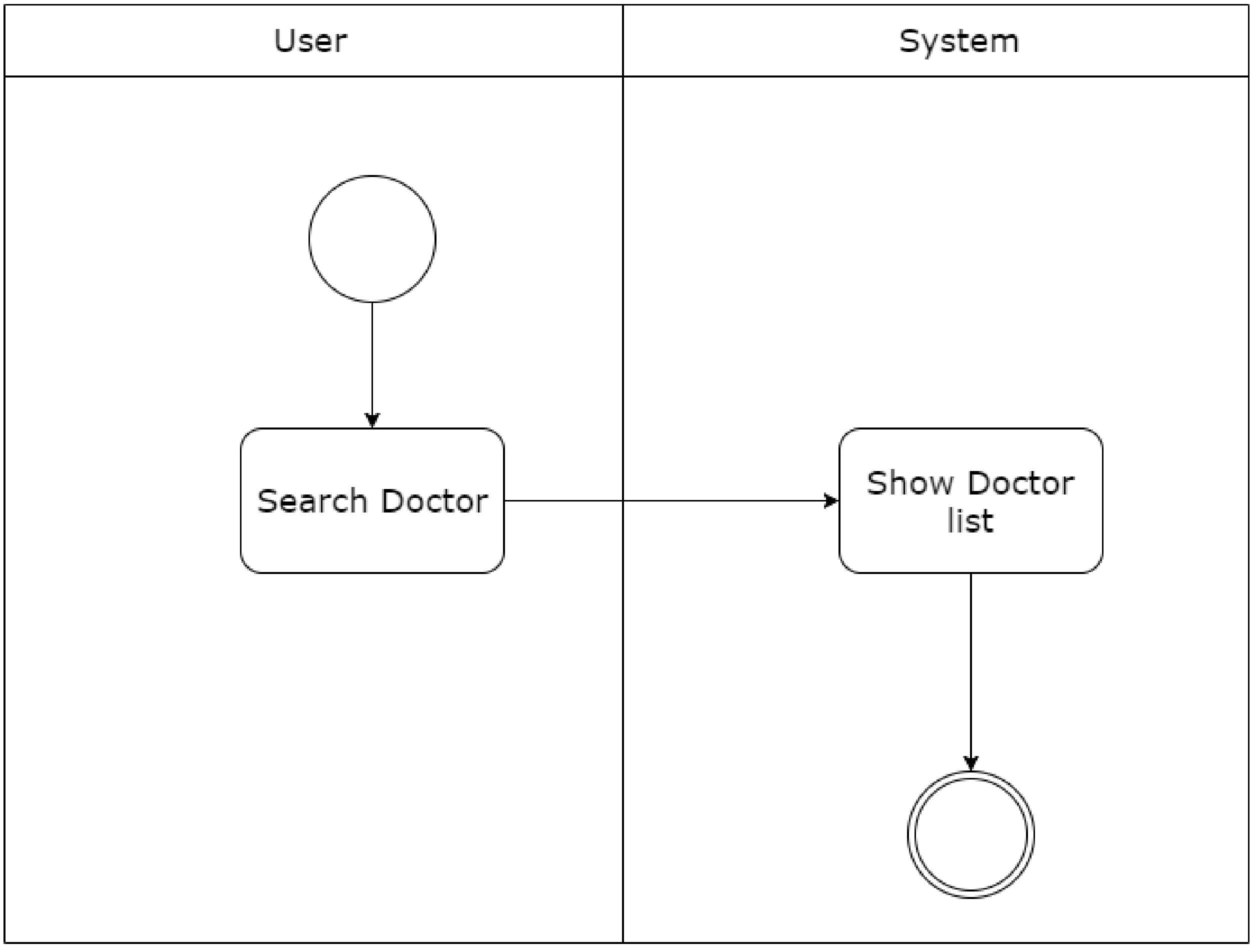

Figure 29: Swim lane diagram of use case 1.7 (search doctor)

Chapter 5

Data Model

5.1 Data Modeling Concept

If software requirements include the need to create, extend, or interface with a database or if complex data structures must be constructed and manipulated, a software team may choose to create a data model as part of overall requirements modeling.

5.2 Data Objects

A data object is a representation of information which has different properties or attributes that must be understood by software.

5.2.1 Identify Data Objects

Nouns having attributes are selected as data object. Those who doesn’t have any attributes have covered under the data objects. I found the following data objects in DAS.

Data Object: User

Attributes:

Attributes of the data object is given below:-

- User id
- Password
- First Name
- Last Name
- Email
- Mobile No.
- User Type
- Present Address
- Permanent Address
- Picture

Data Object: Doctor

Attributes:

Attributes of the data object is given below:-

- Doctor id
- User id
- Registration No.
- Educational degrees
- Specialty
- Visit amount for first comers
- Visit amount for who has visited once but not more than five times
- Visit amount for those who has visited more than five times

Data Object: Patient

Attributes:

Attributes of the data object is given below:-

- Patient Id
- User Id
- Name
- Location

Data Object: Assistant

Attributes:

Attributes of the data object is given below:-

- Assistant Id
- Doctor Id
- User ID
- Name

Data Object: Appointment

Attributes:

Attributes of the data object is given below:-

- Appointment Id
- Doctor id
- Patient id

- Assistant id
- Time
- Serial number

Data Object: Disease Symptom

Attributes:

Attributes of the data object is given below:-

- Symptom id
- Patient id
- Doctor id
- Assistant id
- Appointment id

Data Object: Prescription

Attributes:

Attributes of the data object is given below:-

- Prescription Id
- Patient id
- Doctor id
- Assistant id
- Symptom id
- Medicines

Data Object: Schedule Configuration

Attributes:

Attributes of the data object is given below:-

- Id
- Doctor id
- Visiting date
- Visiting time
- Visit buffer time
- Number of category A patients per day

- Number of category B patients per day
- Number of category C patients per day
- Appointment Rescheduling type

Data Object: User Activation

Attributes:

Attributes of the data object is given below:-

- Activation id
- User id
- Activation code

5.2.3 Entity Relationship Diagram

In this section relationship between different data entities are shown Chen diagram is used for the representation of the relationships.

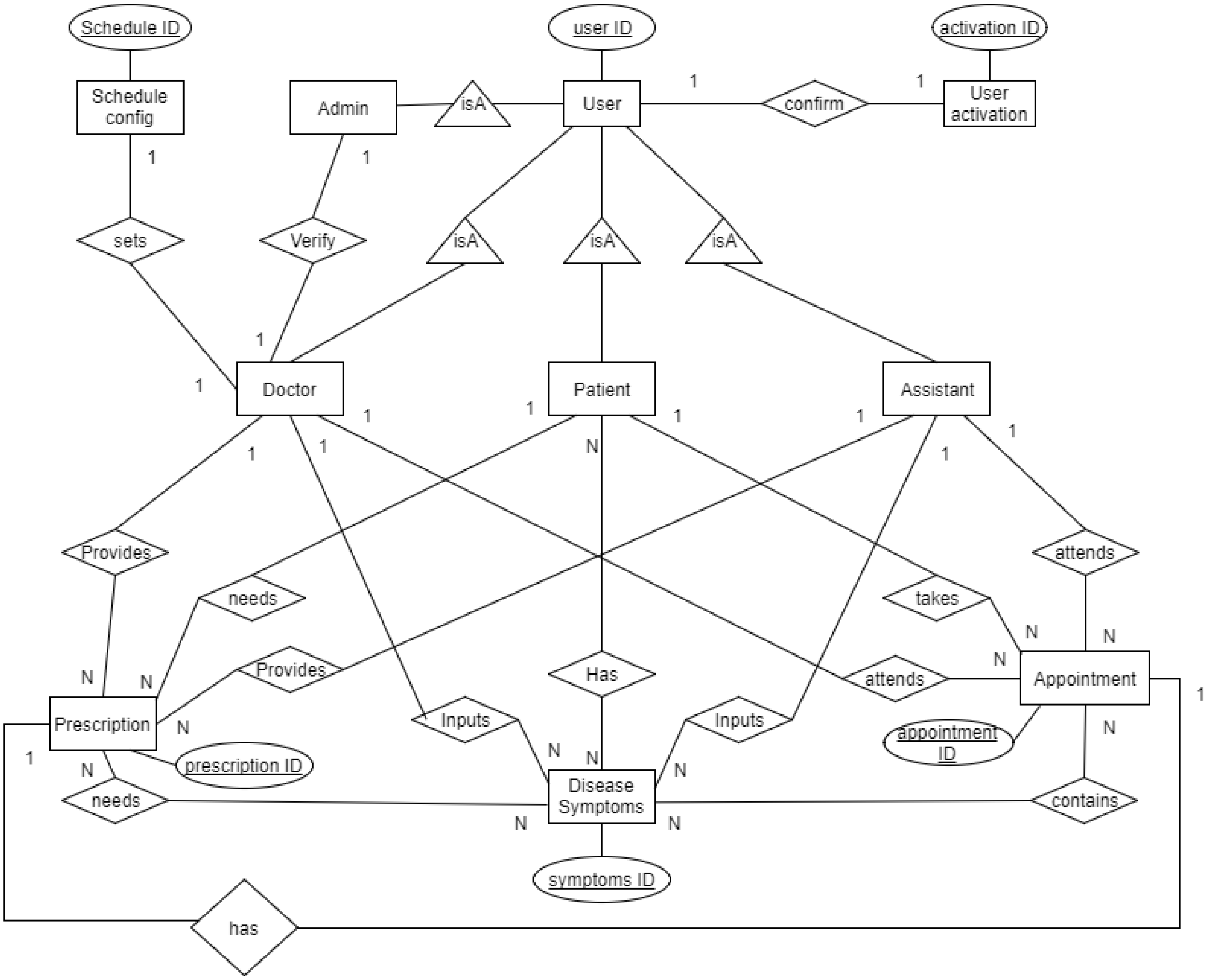

Figure 30: ER Diagram for data based modeling (Chen style convention)

Chapter 6

Class-Based Model

6.1 Introduction

Class-based modelling represents the objects that the system will manipulate, the operations that will be applied to the objects, relationships between the objects and the collaborations that occur between the classes that are defined.

6.2 Specifying Attributes

Analysis classes and their attributes, which are found from the analysis of the user scenario are given below:-

- Doctor = Name + Age + Gender + Degrees + Specialization + ChamberAddress + Contact No
- Patient= Name + Age + Gender + Address+ ContractNo
- Appointment = Date + DoctorId + PatientId + Fees + AppointmentFees
- Authentication= UserName + mobile no. + Email + Password +ActivationStatus
- Schedule configuration = PatientCategory + RescheduleType
- Admin = username

6.3 Defining Operations

The methods of the analysis classes that are found from the user scenario analysis are given below:-

- Doctor = setSchedulingConfig(), registerAssistant()
- Patient = add ( ), update ( ), delete ( )
- Appointment = take_appointment(), cancel_appointment(), getAppointmentList()
- Authentication = SignIn(),Signup(), change_password ( ), forget_password ( )
- Schedule Configuration = setpatientVisitTime(), SetAppointmenntBasedOnCategory()
- 6.Admin= registerDoctor()

6.4 Class Diagram

The interaction between classes to accomplish certain goal are show in this part.

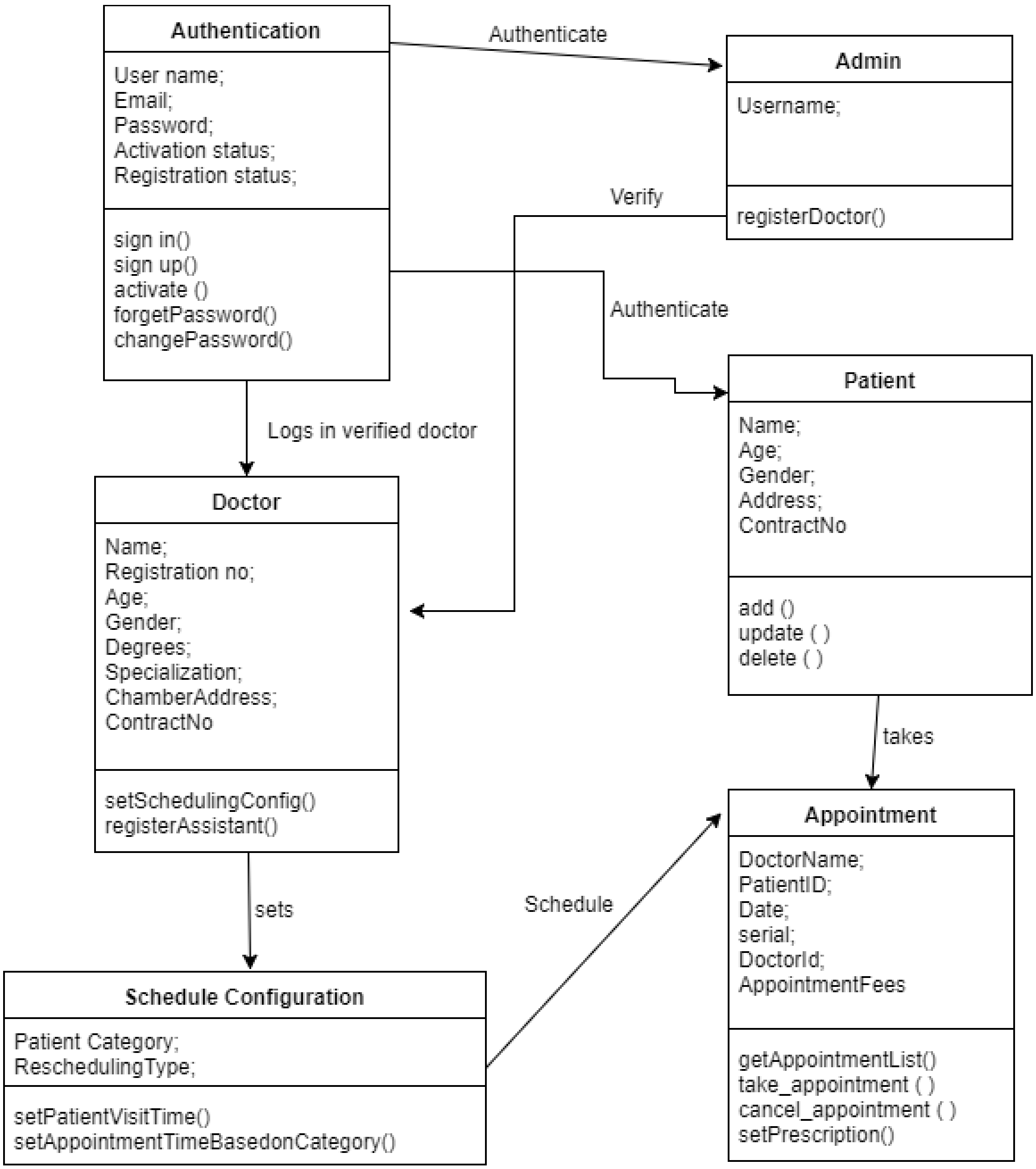

Figure 31: CRC diagram

Chapter 7

Flow-Oriented Model

7.1 Introduction

Although data flow-oriented modeling is perceived as an outdated technique by some software engineers, it continues to be one of the most widely used requirements analysis notations in use today.

7.2 Data flow diagram

A data flow diagram (DFD) is a significant modeling technique for analyzing and constructing information processes. DFD literally means an illustration that explains the course or movement of information in a process. DFD illustrates this flow of information in a process based on the inputs and outputs. A DFD can be referred to as a Process Model.

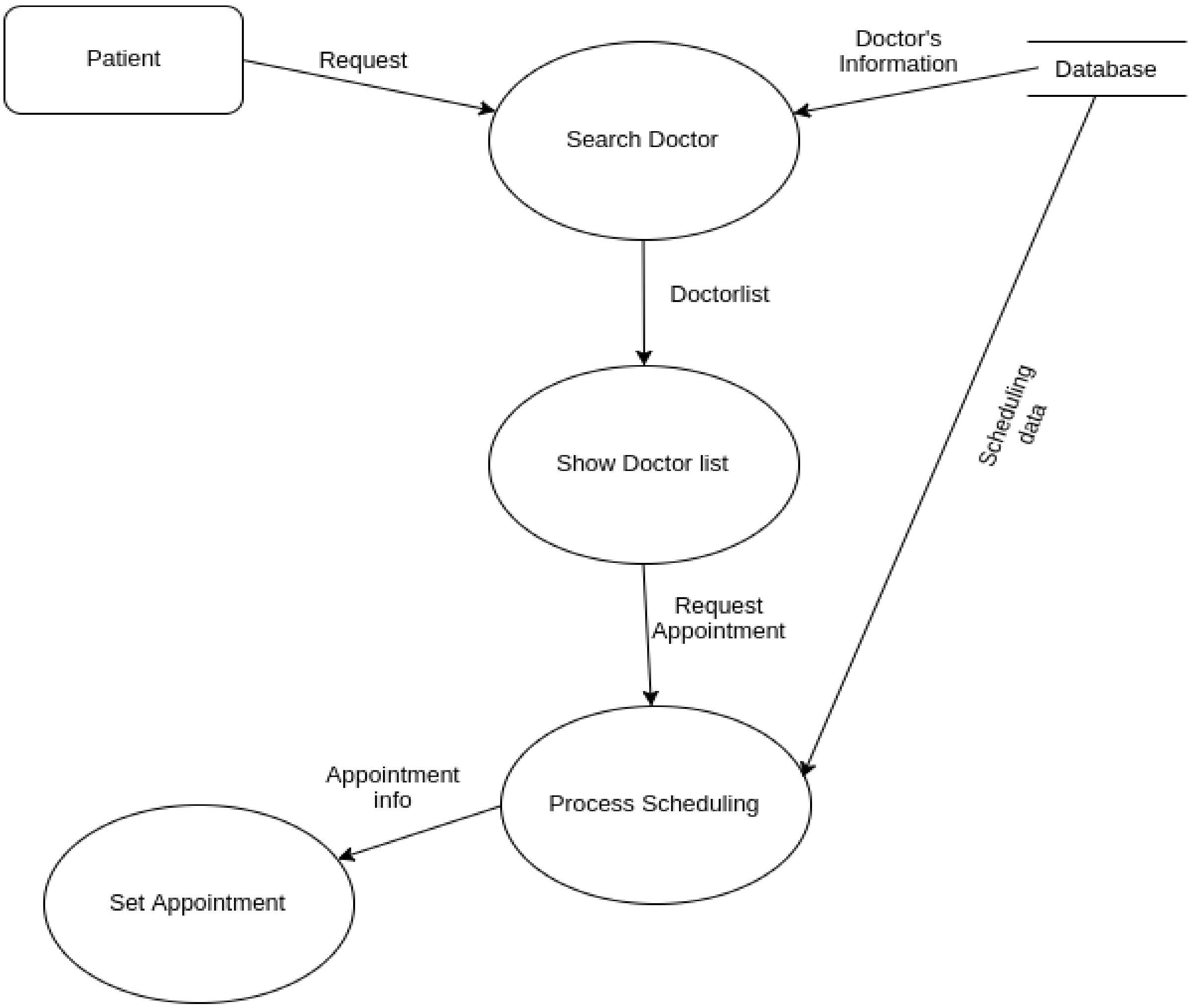

Figure 32: Level 1.1 Data Flow Diagram of searching doctor and taking appointment

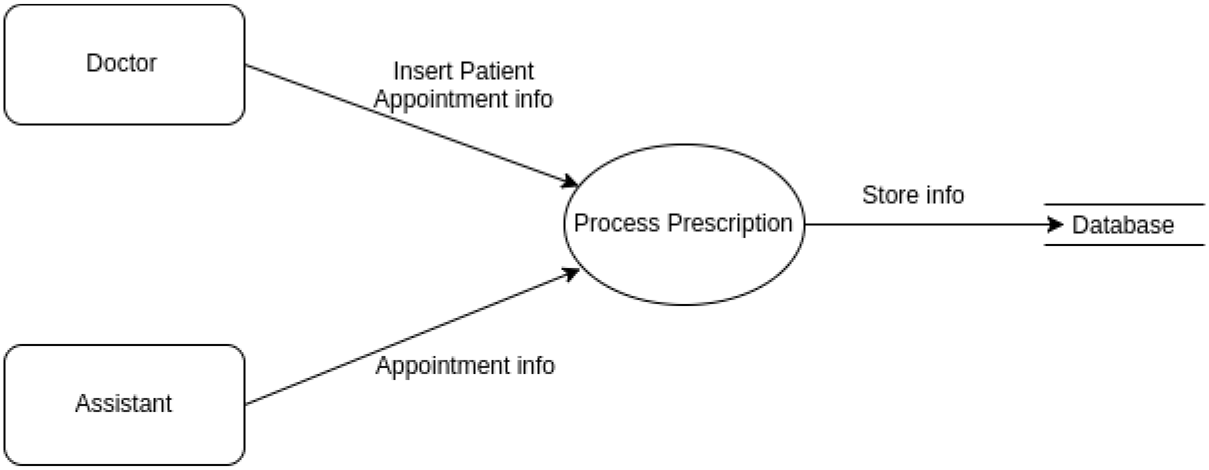

Figure 33: Level 1.2 Data Flow Diagram of prescription

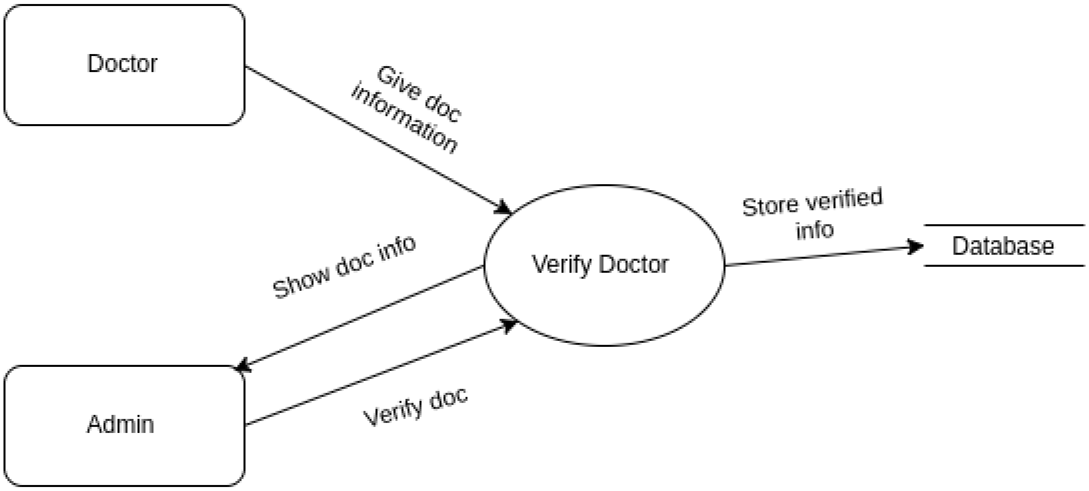

Figure 34: Level 1.3 Data Flow Diagram of verify doctor

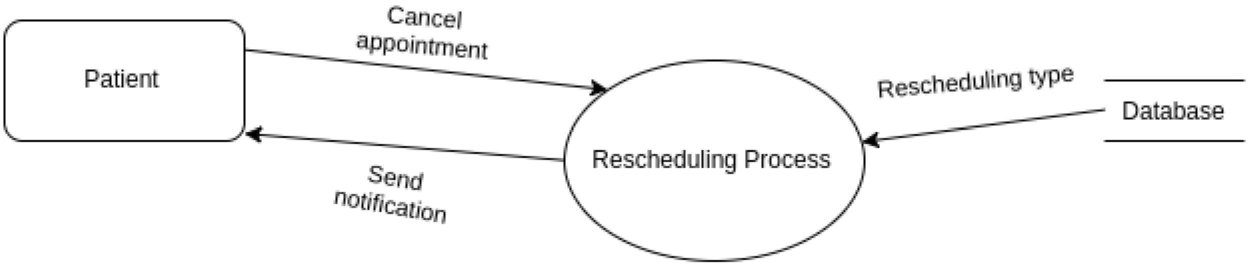

Figure 35: Data Flow Diagram of canceling and rescheduling appointment

Chapter 8

Behavioral Model

Behavioral model is used for better understanding of the system changes. For this particular web project this chapter is skipped as the other chapters are clear enough for the understanding of the system's behavioral changes.

Chapter 9

Software Design Architecture

9.1 Architectural Design for OOP

The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.

9.1.1 Representing the system in context

As architectural design initiates, the software to be established must be put into context, which is-

- The design should define the outside objects (other systems, devices, and people) that the software interacts with and the way of the interaction.
- This information can usually be attained from the requirements model and all other information gathered during requirements engineering.
- Once context is modeled and all external software interfaces have been described, you can identify a set of architectural archetypes.

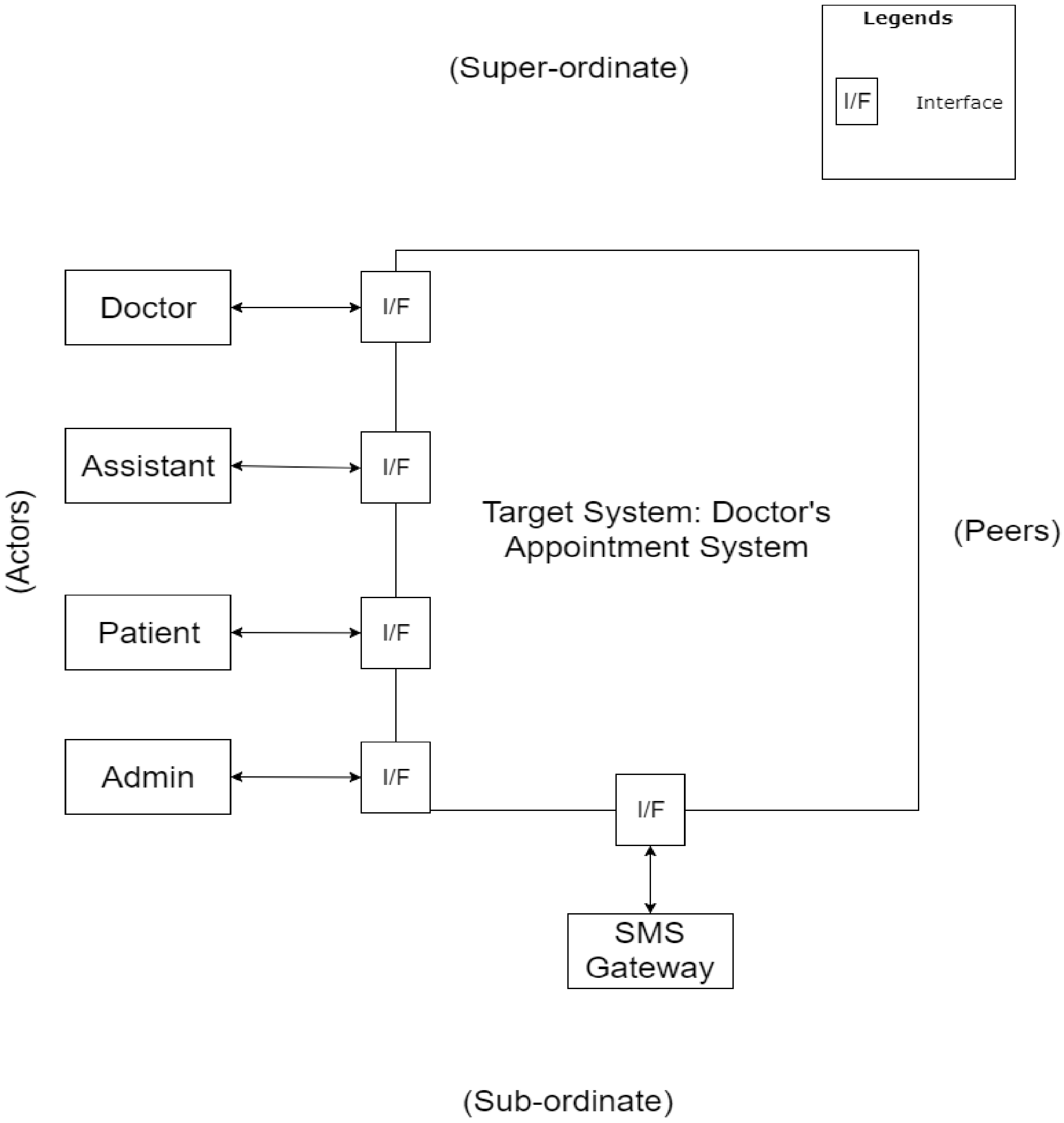

Figure 36: Representing the system in context

9.1.2 Define Archetypes

An archetype is an abstraction (similar to a class) that represents one element of system behavior.The archetypes of the project is given below:-

1. **Authentication:**

Components:

Components of the archetype is given below:-

- Sign up
- Sign in
- Account activation
- Change password
- Forget password

Class:

Classes which are related to the component are given below:-

- Doctor
- Patient
- Admin

2. **Configure appointment scheduling:**

Components:

Components of the archetype is given below:-

- Set time for patients category
- Set rescheduling system
- Schedule appointment
- Reschedule appointment

Class:

Classes which are related to the component are given below:-

- Doctor
- Schedule configuration

3. **Notification:**

Components:

Components of the archetype is given below:-

- Send SMS
- Send email

Class:

Classes which are related to the component are given below:-

- Doctor
- Patient
- Admin
- Schedule configuration

4. Search Doctor:

Components:

Components of the archetype is given below:-

- Search doctor
- Show doctor list

Class:

Classes which are related to the component are given below:-

- Doctor
- Patient

5. Appointment:

Components:

Components of the archetype is given below:-

- Show appointment list
- Take appointment
- Cancel appointment

Class:

Classes which are related to the component are given below:-

- Doctor
- Patient

- Schedule configuration

6. Doctor Verification

Components:

Components of the archetype is given below:-

- Verify Doctor

Class:

Classes which are related to the component are given below:-

- Doctor
- admin

9.1.3 Describing Instantiation of the system

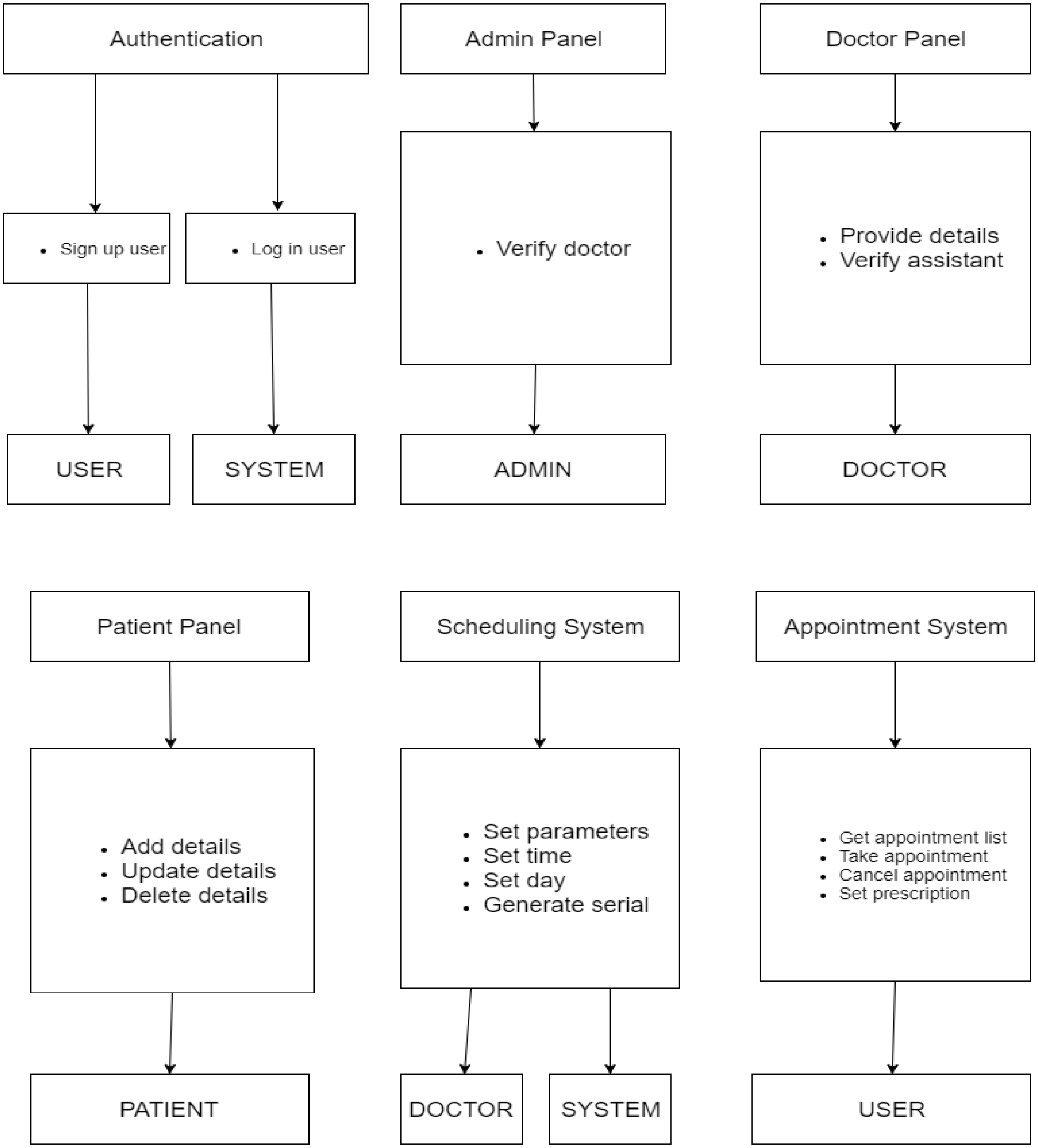

Figure 37: Refining archetype into component and classes

Chapter 10

Conclusion

I am pleased to submit the final SRS report on the system named Doctor Appointment System (DAS). DAS will eradicate the problems making appointment with doctors. This report contains use cases which are the most important part of the scenario based modeling as Ill as for the app for development. It also contains data, class based modeling. I hope that the SRS will be a conductor to the developer.

My next task is to develop DAS on the basis of this document.

References

1. Pressman, Roger S. Software Engineering: A Practitioner's Approach (7th ed.). Boston, Mass: McGraw-Hill. ISBN 0-07-285318-2
2. Database System Concepts, 5th Ed. ©Silberschatz, Korth and Sudarshan
3. Sommerville, I. Software Engineering, 7th ed. Harlow, UK: Addison Wesley, 2006