Software Requirements Specification

for

Doctor Appointment Booking System

Version 1.0 approved

Prepared by Snigdha Taduru

15IT149

January 9, 2018

Table of Contents

Table of Contents ..... **ii**
Revision History ..... **ii**
1. Introduction **1**
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Intended Audience and Reading Suggestions 1
1.4 Product Scope 1
1.5 References 2
2. Overall Description **2**
2.1 Product Perspective 2
2.2 Product Functions 2
2.3 User Classes and Characteristics 3
2.4 Operating Environment 3
2.5 Design and Implementation Constraints 3
2.6 User Documentation 3
2.7 Assumptions and Dependencies 3
3. External Interface Requirements **3**
3.1 User Interfaces 3
3.2 Hardware Interfaces 4
3.3 Software Interfaces 4
3.4 Communications Interfaces 4
4. System Features **4**
4.1 System Feature 1 4
4.2 System Feature 2 5
4.1 System Feature 3 5
5. Other Nonfunctional Requirements **6**
5.1 Performance Requirements 6
5.2 Safety Requirements 6
5.3 Security Requirements 6
5.4 Software Quality Attributes 6
5.5 Business Rules 6
6. Other Requirements **7**
Appendix A: Glossary **7**
Appendix B: Analysis Models **8**
1. Activity Diagram 8
2. Entity Relation Diagram 9
3. Navigation Diagram 9
4. Sequence Diagram 10
5. Use case Diagram 11
Appendix C: To Be Determined List **12**

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Doctor Appointment booking website. This website will be a single subsystem. This document will explain the features of the system, the interfaces of the system, the functions of the system and the constraints under which it must operate.

1.2 Document Conventions

Bold text has been used to emphasize section and sub-section headings. Italicized text is used to label and recognize diagrams.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) is the requirements work product that formally specifies Doctor Appointment Booking System. The objective of this document is to formally describe the system, its high level requirements including functional requirements and non-functional requirements and constraints. It is primarily intended to the developer. The detail structure of this document is organized as follows:

Section 2 of this document provides an overview of the product. This includes a general description of the product, user characteristics, general constraints and any assumption for this system. This section helps to maximize the developer's ability to build a system that supports the requirements.

Section 3 presents the interface requirements such as user, hardware, software and communication interfaces.

Section 4 specifies the systems' features.

Section 5 deals with the non-functional requirements.

1.4 Product Scope

This software application system will be a Doctor Appointment booking website called 'bookadoc'. In India 26.3% of deaths (as per 2016) are due to the inefficiency of the doctors. This system 'bookadoc' provides a list of the best doctors in our nation, their speciality and their location, based on your preferences and conditions you can book an appointment. The aim of the system is to minimize the death rate due to medical insufficiency.

More specifically, anyone can see the list of available doctors their areas of speciality and their location. However in order to book an appointment the user has to login and enter their details. Based on certain criteria like income, area of location, etc.

'bookadoc' will be a user-friendly website which can be easily understood by the common people.

1.5 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.*

Doctors and Surgeons in India www.transearthmedicaltourism.com D.S Harris, Stephanie Riley

2. Overall Description

2.1 Product Perspective

The system 'bookadoc' is a self-contained system that manages schedules of the available doctors and the patients' information.

2.2 Product Functions

The system functions can be described as follows:

Registration: A person can register either as a patient or as a doctor. When a patient registers information such as their name, date of birth, blood group, residence, etc are entered and stored into the system. The doctors' information such as there are of speciality and their schedule is stored in the system.

Login: A patient or doctor can login using their username and password. The patient can then proceed to booking an appointment with the required doctor.

Logout: The patient can logout after use to prevent misuse of his/her details.

Delete Account: Once the patient receives treatment he/her can delete their account if wanted.

Search for Doctors: Any user can visit the website and browse the list of the highly qualified doctors.

Book Appointment: A person once logged in can book an appointment for their desired doctor.

Display Schedule for Doctors: The doctors can view their schedule as well as details of their patients.

2.3 User Classes and Characteristics

This system can be used by any individual. Given the condition that basic computer operating skills will be needed. The system is also designed to be user friendly. It uses graphical user interface (GUI).

General Viewers: Anyone with internet access can browse the website and search for their preferred doctor.

Patients: The patients can book an appointment with their preferred doctors.

Doctors: The doctors can view their patients' medical conditions and other details, they can also view their schedule hence manage time efficiently.

2.4 Operating Environment

This application will be developed in Windows provided python, xampp are available on the system.
The application can also run on a Linux system containing python and xampp.

2.5 Design and Implementation Constraints

The system shall be a web based application.
The system must be user friendly.
The development environment shall be Windows 8.
The language used in the system must be easily understandable to the public.
The computer must have sufficient memory.

2.6 User Documentation

For user documentation and information, please consult section 3: External Interface Requirements.

2.7 Assumptions and Dependencies

It is assumed the patients and doctors can operate and use the system.

3. External Interface Requirements

3.1 User Interfaces

This application will be a web application. The backend used for this project will be python whereas the frontend will be developed using several languages like html, css, javascript, php, bootstrap, etc.

The GUI of the application will be highly user friendly and minimal English skills will be required to easily migrate through the various pages of the application.

3.2 Hardware Interfaces

The hardware requirements include a computer with 64GB RAM.

3.3 Software Interfaces

This system is built on Windows operating system.

The backend of this application will be built in python whereas the frontend involves html, css, javascript, php, etc. SQL will be used as the querying language and the system will use MySQL database.

3.4 Communications Interfaces

The system requires an HTTP to communicate with the server. The system can be configured to be accessed via any available port.

The web based UI is the only means of communication between the user and the system.
A proper internet connection is recommended.

4. System Features

4.1 System Feature 1

Patient Management Feature

4.1.1 Description

Patient Management feature allows proper organization of patients and provides a facility for storing their necessary details. Details stored will include the patient's username, password, pid, residence details, blood groups, previous medical reports, etc. This feature includes functions such as login, logout, register. This feature is of high importance as the patients' can book appointments with their preferred doctors, and doctors in turn can also view their patients previous medical history through this feature.

4.1.2 Stimulus/Response Sequences

When the user logs in the web page will get reloaded and now the user can access the book appointment feature. The user may sign out after appointment booking is complete.

4.1.3 Priority

This requirement is of high priority.

4.1.4 Functional Requirements

REQ-1: The system must have sufficient memory so as store the details of all the patients in the database.

4.2 System Feature 2

Booking Appointment

4.1.1 Description

This feature allows the user to book an appointment with their preferred doctor after viewing the doctor's area of speciality. The patient can book an appointment on their preferred date and time in accordance with the doctors' schedule. This feature is the main feature of this system.

4.1.2 Stimulus/Response Sequences

After the users books an appointment with his/her preferred doctor then the patient's appointment will get added to the doctor's schedule. The doctor then can view the patients' details.

4.1.3 Priority

The requirement is of high priority.

4.1.4 Functional Requirements

REQ-1: The system must update the doctors' schedule as and when an appointment is booked, hence the system must be considerably fast.

4.3 System Feature 3

Doctor Schedule Management

4.1.1 Description and Priority

The doctors can view their daily schedule along with their patients' previous medical history. They can also view their patients details like name, address, blood type, etc.

4.1.2 Stimulus/Response Sequences

The doctors' schedule gets updated automatically at the time of appointment booking. The doctor can view his patients' details.

4.1.3 Priority

This requirement is of medium priority.

4.1.4 Functional Requirements

REQ-1: Retrieval of information must be fast.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system is required to support multiple terminals simultaneously. The system should handle reasonable number of users without break or inconsistency. The system must input and retrieve information from the database considerably fast.

5.2 Safety Requirements

The application should be password protected , and modification and deletion should only be done by authorized users.

5.3 Security Requirements

The system, at any time, should be accessed only by authenticated users. The system is required to end a session automatically, when an open session is not used for a specific period of time.
The details of a patient cannot be accessed by other patients and the patient's password will be encrypted and stored safely.

5.4 Software Quality Attributes

MySQLi functions are used to access MySQL database servers. MySQL version 4.1.13 or newer should be used.

PHP version 5.0.0 or newer should be used.

5.5 Business Rules

None.

6. Other Requirements

MySQL database will be used. XAMPP server must be installed on the system.

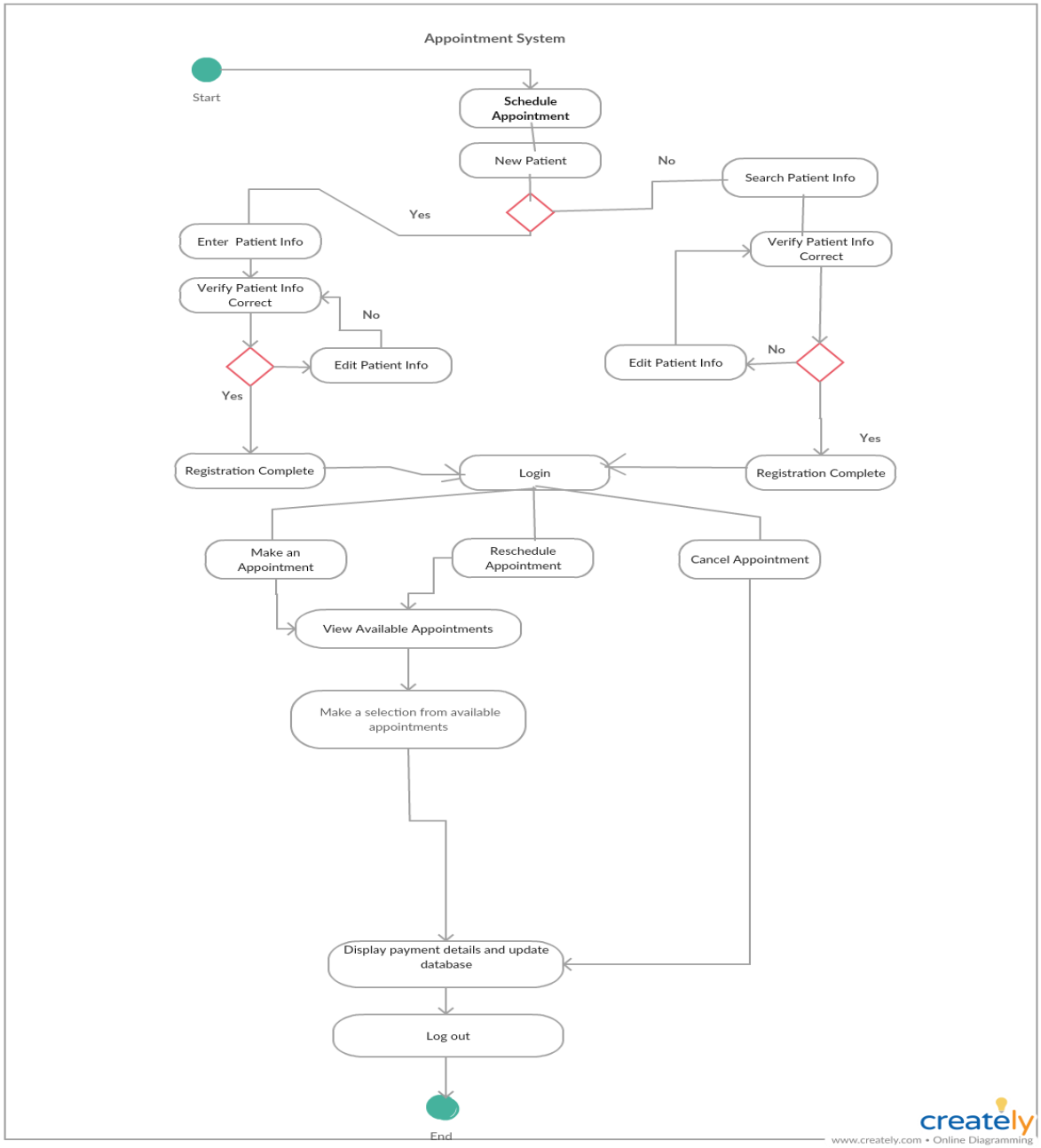
MySQL version should be 4.1.13 or newer.

Appendix A: Glossary

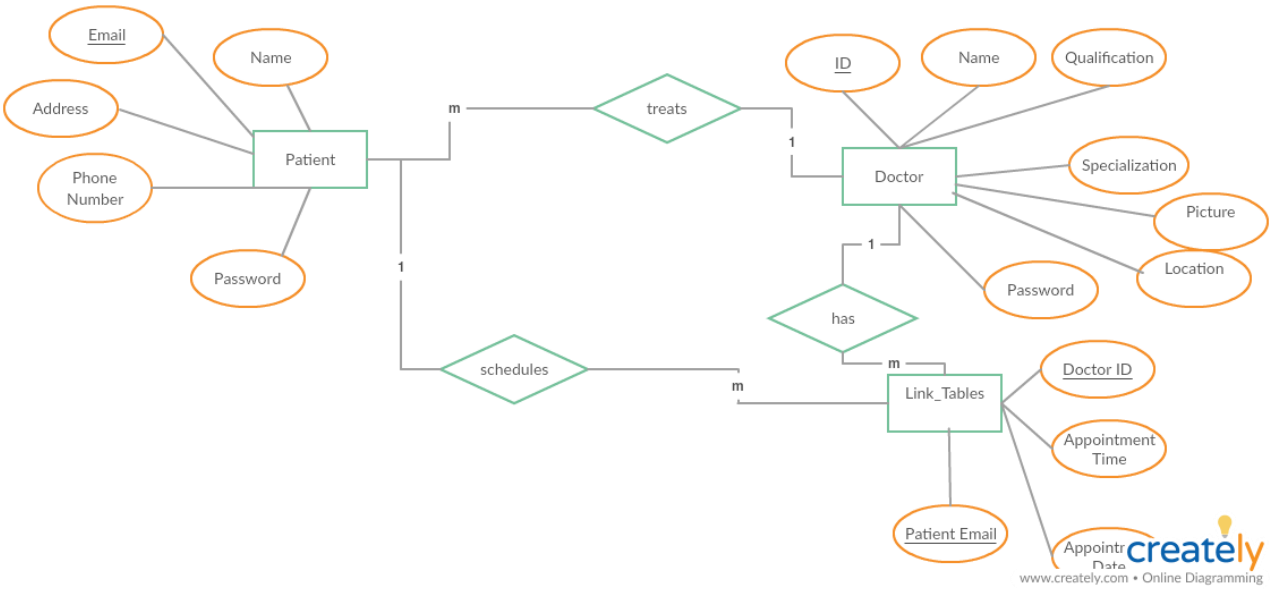
SRS- Software Requirements Specifications
GUI- Graphical User Interface
Pid number- Patient Identification Number
HTTP- Hyper Text Transfer Protocol
SQL- Structured Query Language
CSS- Cascading Style Sheet

Appendix B: Analysis Models

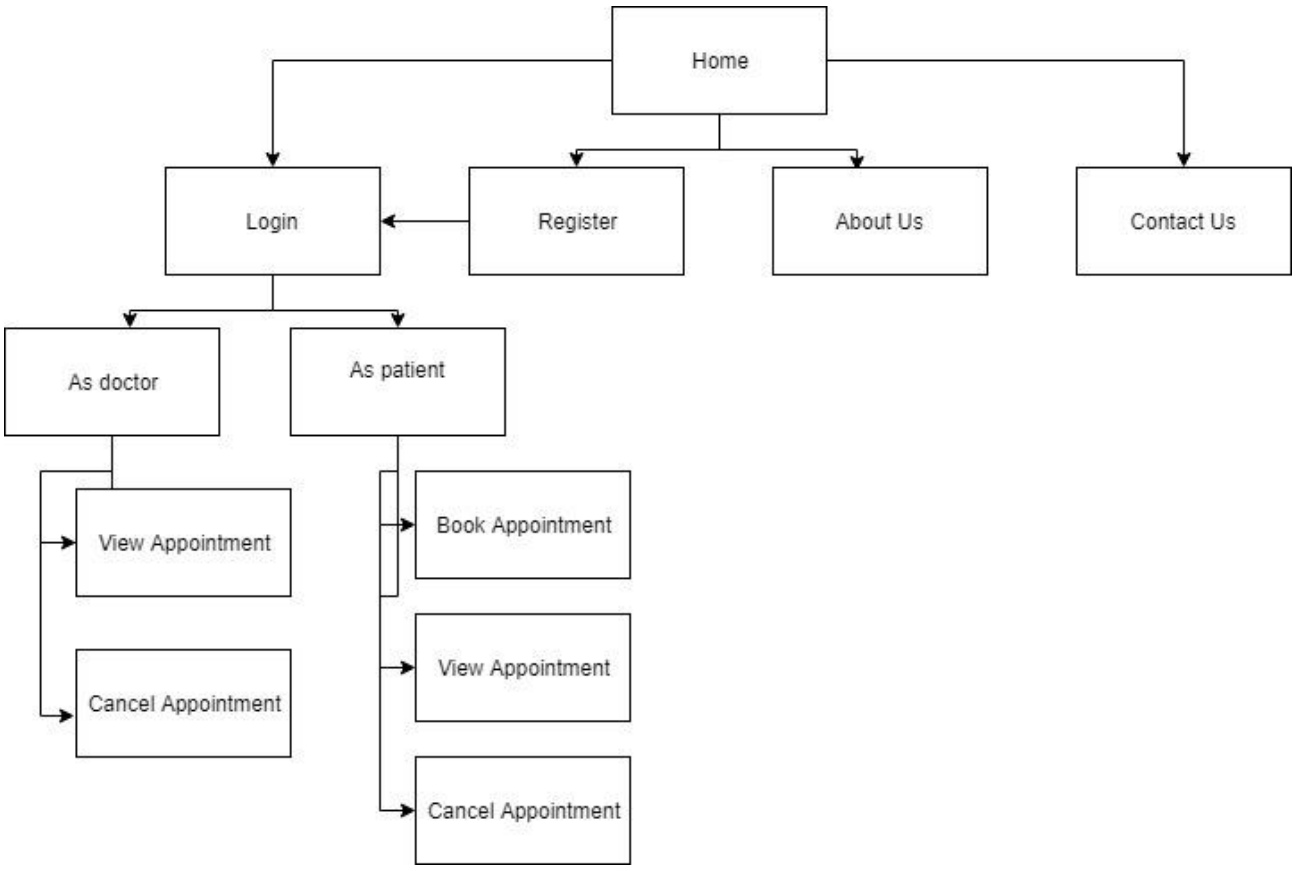
1. Activity Diagram



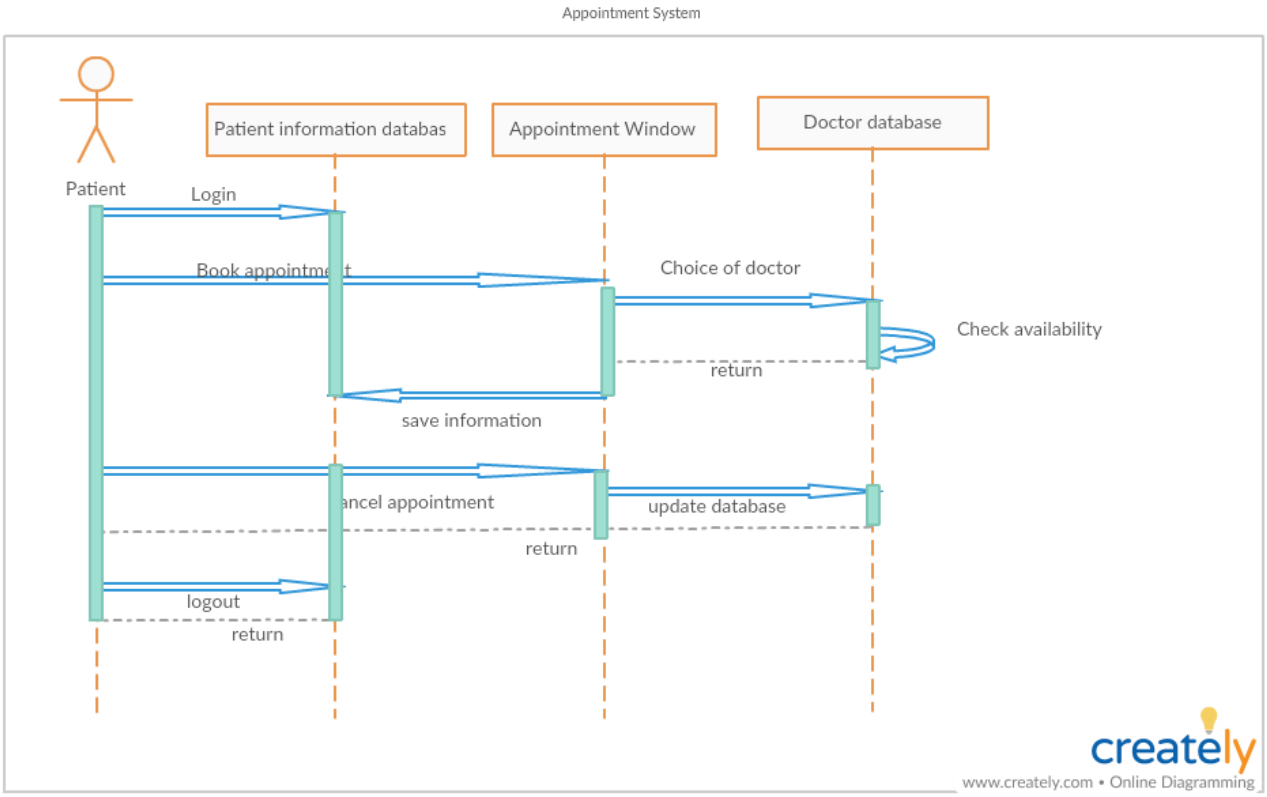
2. Entity Relation Diagram

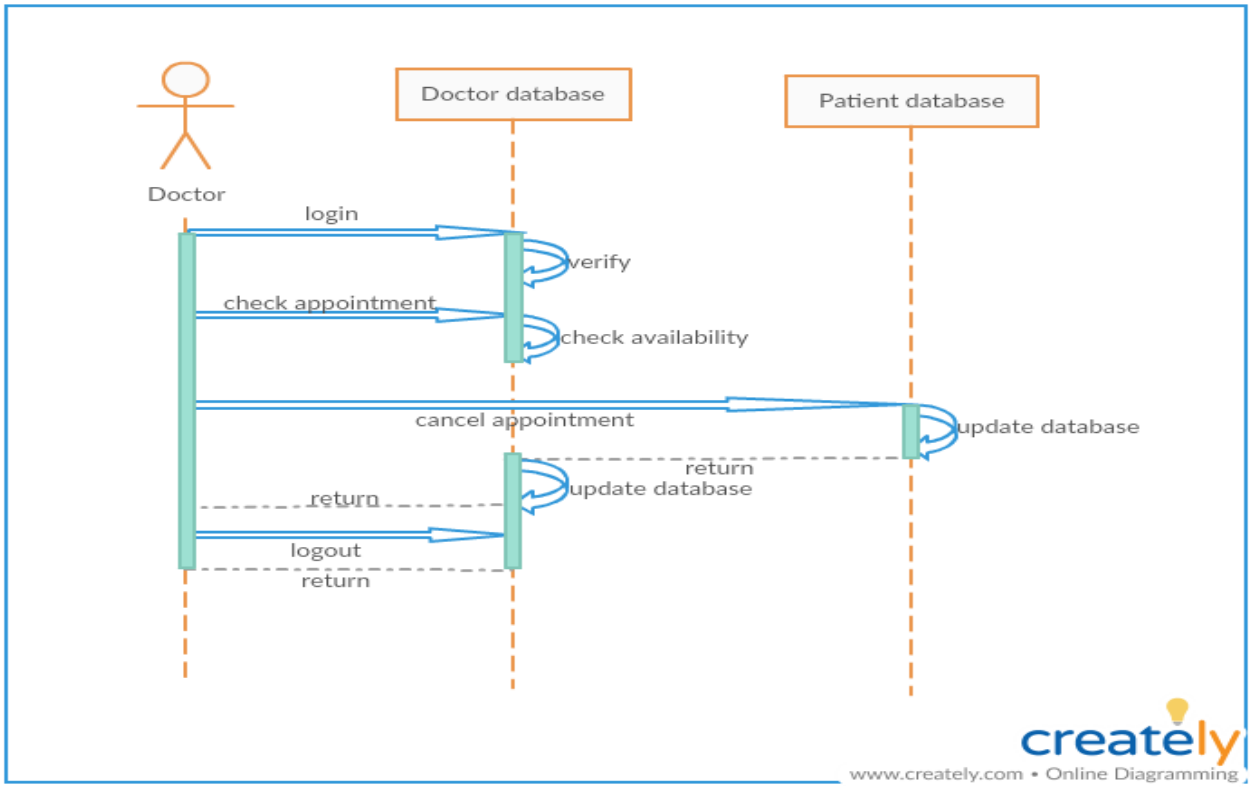


3. Navigation Diagram

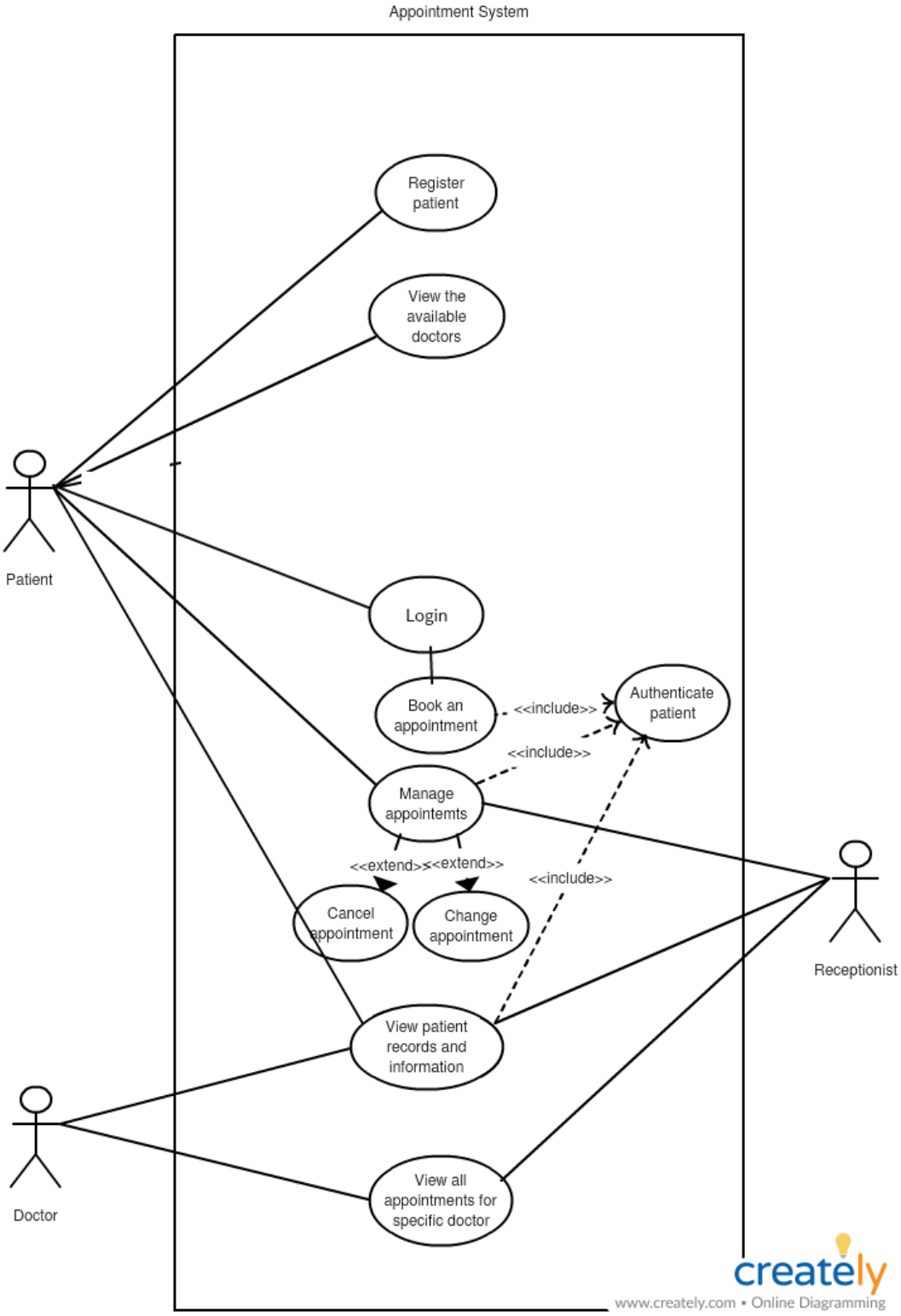


4. Sequence Diagram





5. Use case Diagram



Appendix C: To Be Determined List

Business Rules