

# Srs

*by* Haniya Naseem

---

**Submission date:** 11-Sep-2020 03:39AM (UTC+0300)

**Submission ID:** 1384113276

**File name:** SRS\_ASSIGNMENT.pdf (1.5M)

**Word count:** 7176

**Character count:** 41133

---

8

# Software Requirements Specification

For

## ONLINE HOSPITAL APPOINTMENT SCHEDULE SYSTEM

Version 1.0 approved

Prepared by:

Muhammad Abdullah Hayat (SE - 041)  
Rana Muhammad Ibrahim (SE - 034)

5  
27<sup>th</sup> August, 2020

## TEAM MEMBERS

---

Name	Responsibility
Muhammad Abdullah Hayat SE-041	Section 2-Complete Section 3 Use-Case Model Section 3 Software System Attributes
Rana Muhammad Ibrahim SE-034	Section 1-Complete Section 3 Domain Model Section 3 Performance Requirements Section 3 Performance Requirements Section 3 Logical Database Requirements

## Table of Contents

<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 PURPOSE.....	1
1.2 SCOPE .....	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....	1
1.4 REFERENCES.....	2
1.5 OVERVIEW.....	2
<b>2. THE OVERALL DESCRIPTION.....</b>	<b>3</b>
2.1 PRODUCT PERSPECTIVE .....	3
2.1.1 System Interfaces .....	3
2.1.2 User Interfaces .....	3
2.1.3 Hardware Interfaces.....	4
2.1.4 Software Interfaces.....	4
2.1.5 Communications Interfaces.....	4
2.1.6 Memory Constraints .....	4
2.1.7 Operations.....	4
2.1.8 Site Adaptation Requirements.....	4
2.2 PRODUCT FUNCTIONS.....	5
2.2.1 Patients.....	5
2.2.2 Doctors.....	5
2.2.3 Administrator .....	6
2.3 USER CHARACTERISTICS .....	6
2.4 CONSTRAINTS.....	6
2.4.1 Schedule.....	6
2.4.2 Cost.....	6
2.4.3 Scope.....	6
2.5 ASSUMPTIONS AND DEPENDENCIES.....	6
<b>3. SPECIFIC REQUIREMENTS.....</b>	<b>7</b>
3.1 FUNCTIONAL REQUIREMENTS .....	7
3.1.1 Domain Model.....	7
3.1.2 Use case model.....	8
3.2 PERFORMANCE REQUIREMENTS .....	18
3.2.1 Static Requirements.....	18
3.2.2 Dynamic Requirements .....	18
3.3 LOGICAL DATABASE REQUIREMENTS .....	19
3.4 DESIGN CONSTRAINTS .....	19
3.5 SOFTWARE SYSTEM ATTRIBUTES .....	19
3.5.1 Reliability .....	19
3.5.2 Availability.....	20
3.5.3 Security .....	20
3.5.4 Maintainability .....	20
3.5.5 Usability .....	21
3.5.6 Portability.....	21
3.5.7 Correctness.....	21
3.5.8 Interoperability.....	21
3.5.9 Scalability requirements:.....	22
<b>APPENDIX A.....</b>	<b>23</b>

2

## 1. Introduction

The Software Requirements and Specifications archive (SRS) gathers, and characterizes significant and important requirements and highlights of the Online Hospital Appointment Schedule System (O.H.A.S.S.). O.H.A.S.S. will be utilized by centers, hospitals, and other clinics to deal with all parts of managing and planning appointments for their patients. This software replaces the role that would typically be given to an assistant by permitting patients and specialists to schedule their own appointments. This application can be adjusted for organizations that schedule appointments for their customers. The report will represent the requirements and essentials of the O.H.A.S.S.

12

### 1.1 Purpose

This document defines and specifies the requirements for an Online Hospital Appointment Schedule System. The purpose of this document is to explain the functions, constraints on system and design model of the system. The document includes different models that are block diagram, class diagram, sequence diagram, use-case diagram and state diagram. This document will provide an overall demonstration to the designers and developers of Online Hospital Appointment Schedule System.

### 1.2 Scope

The Online Hospital Appointment Schedule System has a great scope as it can result in a handy tool for both patients and specialists to schedule their appointments. This system will be efficient in workplace as it provides a basic necessity to work from home. O.H.A.S.S is comprised of some most important features which may include patients planning their appointment or doctors answering the appointments. There are three user interfaces in this system which include: patient view, specialist view, and an overseer view.

### 1.3 Definitions, Acronyms, and Abbreviations.

2 OS Operating System	OS is the framework software answerable for the immediate control and the board of equipment and essential framework tasks. Moreover, it gives an establishment whereupon to run application software, for example, word preparing projects and internet browsers <sup>1</sup> .
DB Database	A database is an assortment of information components (realities) put away in a PC in an efficient way, with the end goal that a PC program can counsel it to address questions <sup>2</sup> .
Web Server	A web server is a is a PC answerable for serving pages, generally HTML documents, by means of the HTTP convention to customers, for the most part internet browsers <sup>3</sup> .

<sup>1</sup> Definitions from [www.wikipedia.com](http://www.wikipedia.com)

<sup>2</sup> See foot-note 1.

<sup>3</sup> See foot-note 1.

<sup>2</sup> Patient	A patient is any individual who gets clinical consideration, care, or treatment <sup>4</sup> .
PHP	PHP is a mainstream; intelligent programming-language utilized chiefly for creating dynamic websites, and all the more as of late, other software <sup>5</sup> .
<sup>2</sup> Administrator	One who regulates issues; one who coordinates, oversees, executes, or apportions, whether in common, legal, political, or ministerial issues; an administrator <sup>6</sup> .
<sup>2</sup> Appointment	An appointment is a time reserved for something, for example, a specialist visit, much like a booking <sup>7</sup> .
<sup>2</sup> SRS	Software Requirements Specifications <sup>8</sup>
O.H.A.S.S.	Online Hospital Appointment Schedule System
SIN	Social Insurance Number
RDBMS	Relational Database Management System

## 1.4 References

[1] I.-S.-B. "IEEE Std 830-1998," IEEE.

## 1.5 Overview

This Software Requirements Specification document is partitioned in to various subsections. The main section incorporates clarifications of the Purpose, Scope and Organization of the document. The primary section additionally handles the depiction of task explicit words, abbreviations and shortened forms that will be utilized in the document. The second section of the document is isolated into the accompanying five unique sections, each specifying explicit subtleties of framework utilizes and their relating activities: Product Perspective, Product Functions, User Characteristics, Constraints, Assumptions and Dependencies. The third section is an identified posting of the entirety of the requirements depicted for this framework. The forward section contains a posting of all related reference materials utilized in this document.

<sup>4</sup> See foot-note 1.

<sup>5</sup> See foot-note 1.

<sup>6</sup> See foot-note 1.

<sup>7</sup> See foot-note 1.

<sup>8</sup> See foot-note 1.

## 2. The Overall Description

This section incorporates insights concerning what is and isn't anticipated from the O.H.A.S.S. framework notwithstanding which cases are purposefully unsupported and suppositions that will be utilized in the formation of the O.H.A.S.S. frameworks.

### 2.1 Product Perspective

O.H.A.S.S. is expected to be utilized in Medical facilities so as to help improve arrangement administrations. There as of now exists a wide assortment of comparative items available. The vast majority of these business applications are planned to be exceptionally broad and attempt to cover any conceivable business. In spite of the fact that these items exist, Medical centers despite everything exhaust a lot of time and cash planning their arrangements. This is on the grounds that Medical facilities have unmistakable necessities that vary from other organizations. O.H.A.S.S. is an online application that is custom-made to a Medical center's needs.

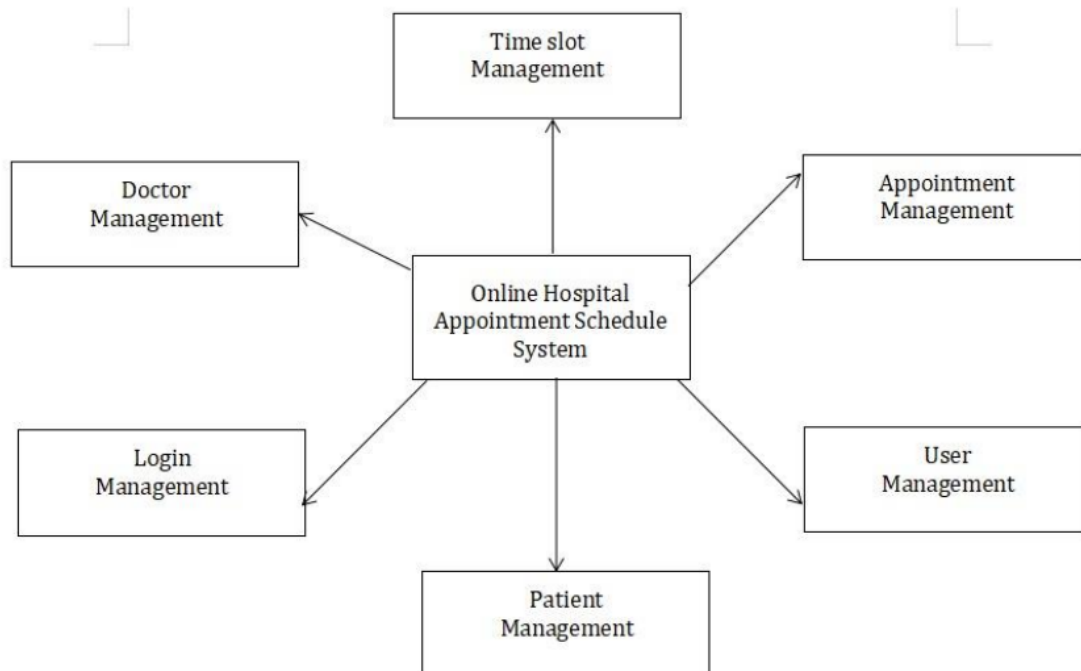


Figure 1-CONTEXT DIAGRAM OF O.H.A.S.S.

#### 2.1.1 System Interfaces

*"The system has no system interface requirements".*

#### 2.1.2 User Interfaces

O.H.A.S.S contains three distinct sorts of clients, and can be seen from alternate points of view:

1. Patients. Patients will have the option to:
  - a) Booking an appointment
  - b) Cancelling an appointment
2. Doctors. Specialist will have the option to:
  - a) Cancel appointment
  - b) Schedule uncommon appointment
3. Administrators. Director will have the option to:
  - a) Activate or deactivate a specialist
  - b) Configure worker inventories
  - c) Configure offices
  - d) Configure workers in a division
  - e) Create a specialist's record

### **2.1.3 Hardware Interfaces**

*“The system has no hardware interface requirements”.*

### **2.1.4 Software Interfaces**

*“The system has no software interface requirements”.*

### **2.1.5 Communications Interfaces**

*“The system has no communication interface requirements”.*

### **2.1.6 Memory Constraints**

O.H.A.S.S is upheld by any internet browser and can be gotten to by any PC with web access. Coming up next is suggested for the engineering of our internet browser:

1. Pentium IV
2. 2 GB RAM
3. 250 GB HDD

### **2.1.7 Operations**

Backup the database on last day of every month at 12:00 am.

### **2.1.8 Site Adaptation Requirements**

*“The system has no site adaptation requirements”.*



## **2.2 Product Functions**

### **2.2.1 Patients**

#### **2.2.1.1 Booking an appointment**

The patient will have the option to book a meeting with any specialist. There are a few kinds of arrangements that must be made by a specialist, for example a medical procedure.

#### **2.2.1.2 Cancelling an appointment**

The patient can drop his own arrangement, as long as he drops it 60 hours ahead of time.

#### **2.2.1.3 Editing Personal data**

The patient will have the option to adjust his/her own data, for example, his/her location, phone number, and so on.

#### **2.2.1.4 Creating his/her record**

The patient needs to enroll with O.H.A.S.S so as to book arrangements. The patient must give the accompanying data: Medicare card number, contact data, username and password.

#### **2.2.1.5 Consulting his/her appointment**

The patient will have the option to counsel the entirety of their active appointments.

### **2.2.2 Doctors**

#### **2.2.2.1 Canceling appointments**

On account of a crisis, a specialist may need to drop his/her arrangements. The framework will send an email to all the patients influenced by the change.

#### **2.2.2.2 Consulting his/her Appointments**

A specialist may need to recognize the entirety of the arrangements that have been planned for a particular day or week.

#### **2.2.2.3 Editing Information**

There are sure special cases in which a specialist won't have the option to follow his/her normal timetable. For instance, excursions, sickness, and so on. O.H.A.S.S permits the specialist to adjust his/her timetable for a brief timeframe.

Scenario O.H.A.S.S-FR-DR-SD-Edit-Personal-Information

Preconditions:

- 1) System must be available
- 2) Doctor account must exist
- 3) Doctor must sign into O.H.A.S.S

#### **2.2.2.4 Scheduling Special Appointments**

At the point when medical procedure is required, an executive must timetable the arrangement for the specialist's sake. The patient won't have authority over the time and date.

### **2.2.3 Administrator**

#### **2.2.3.1 Managing Catalogs**

The overseer will be answerable for the accompanying undertakings: include another specialist, include office data, and include sorts of arrangements, and set time limits for each kind of arrangement.

#### **2.2.3.2 Creating Specialists Accounts**

Each time another specialist joins the center, the head needs to add the specialist to the system.

#### **2.2.3.3 Deleting Specialists Accounts**

On the off chance that a specialist leaves the facility, the chairman will erase this present specialist's data and re-relegate the entirety of his/her arrangements to another specialist.

## **2.3 User Characteristics**

By and large, there are no particular client requirements. There is, notwithstanding, one individual liable for the web worker, the RDBMS. The duty of the RDBMS is to guarantee that the application is on-line consistently.

## **2.4 Constraints**

We should indicate what the restrictions of the venture are. The three most critical venture requirements are timetable, cost, and scope.

### **2.4.1 Schedule**

- The most extreme time of development of the item ought to be of 2 months in the most pessimistic scenario.

### **2.4.2 Cost**

- The venture can't go over the spending given by the organization i.e. Rs. 500K.

### **2.4.3 Scope**

- The clients will have the option to enroll and make arrangements online whenever of the day. Just enrolled clients can make arrangements on the web.
- Any clients can get to all the specialists' profiles.

## **2.5 Assumptions and Dependencies**

- An understanding dropping can just happen 60 hours ahead of time.
- A specialist abolition can just occur 48 hours ahead of time.
- A patient is needed to have his/her Medicare card number to enroll into O.H.A.S.S.

## 3. Specific Requirements

### 3.1 Functional Requirements

#### 3.1.1 Domain Model

*Figure 2 in Appendix A.1* records the candidate idea classes and depicts the relationship types and the properties of O.H.A.S.S.

##### 3.1.1.1 Concept Classes

- **Doctor:** The client who utilizes this framework to deal with the timetable and arrangement.
- **Patient:** The client who makes an online meeting with the specialist.
- **Agenda:** Exactly a similar idea as a plan, in actuality. It is utilized by a specialist to deal with their timetable. Every day in the plan has the accompanying property: Date, non-weekend day, and status. The status shows whether the day is an occasion, or end of the week, or livelihood day, or work day.
- **Appointment:** Patient can book or drop a meeting with the specialist.
- **Appointment Type:** Appointment reason is gathered into some schedule type. Such as blood test etc.
- **Time Slot:** One day is part into numerous little bits of time allotments with equivalent length.

##### 3.1.1.2 Class Attributes:

###### 3.1.1.2.1 Doctor

Name, phone number, address, e-mail address.

###### 3.1.1.2.2 Patient

Name, Medicare card number, SIN, phone number, address, email address.

###### 3.1.1.2.3 Agenda

- Date speaks to one schedule date on the plan.
- Weekday speaks to the non-weekend day in seven days, for example, Monday, Tuesday, and Wednesday, so on.
- Status speaks to whether the day is an occasion, or a workday.
- Office Hour Start: the opening season of that day, as 8:30AM
- Office Hour End: the ending season of that day, as 5:30PM

###### 3.1.1.2.4 Appointment

- Sequence Number: used to recognize one arrangement extraordinarily. It's an affirmation number of the arrangement. Patients and specialists can utilize this number to recognize an arrangement, and then drop it, change it and so on.
- Date: arrangement date.
- Start Time: when an arrangement begins.
- Description: more itemized data about the arrangement reason.
- Status: states whether the arrangement is legitimate, or dropped, or simply missed (the patient neglected to appear without advising the specialist ahead of time)

#### *3.1.1.2.5 Appointment Type*

- Type: the sort of arrangement reason, for example, blood test.
- Description, the detail portrayal about this kind of arrangement.
- Estimated Duration: the assessed time length of this sort of arrangement type. For instance, 20 minutes for blood test.

#### *3.1.1.2.6 Time Slot*

- Start Time: the beginning season of one bit of time.
- End time: the end season of one bit of time: Example: 8:20
- Status: the condition of this bit of time. It shows whether this bit of schedule opening is accessible or not accessible to the patient.

### **3.1.1.3 Concept Class Inter-relation**

- A specialist utilizes a plan to deal with their timetable and arrangement. One plan is just utilized on one specialist (proprietor). One specialist just has one plan.
- Agenda is spoken to by many schedule day, though each schedule day is made out of many time allotment - the equivalent length watches.
- One patient can book numerous meetings with specialists. One arrangement is just connected with one patient and one specialist.
- One arrangement has a place with one arrangement type. The assessed arrangement span is controlled by the arrangement type.
- Appointment can change the status of a lot of nonstop schedule opening on the arrangement day on the specialist's plan.

## **3.1.2 Use case model**

*Figure 3 in Appendix A.2.1.1 lists the candidate concept classes and describes the relationship types and the attributes of the O.H.A.S.S.*

### **3.1.2.1 Use Case O.H.A.S.S-Book-Appointment**

#### *3.1.2.1.1 Purpose*

This function allows a registered patient to book an appointment with a doctor through internet.

#### *3.1.2.1.2 Actors*

The main actors are registered patients.

#### *3.1.2.1.3 Pre-Condition*

The user must login the system successfully.

#### *3.1.2.1.4 Main Scenario*

- A1. The user accesses the system and chooses "book appointment" function.
- A2. The user browses the doctor list and chooses one doctor
- A3. The system shows this week's agenda of the doctor
- A4. The user can browse the calendar, for example, go to next month, or go to next week. Then, the system will display the corresponding agenda.
- A5. The user chooses the appointment date and time, select the appointment type (the reason of the appointment), and input the description of the appointment, and then submit the appointment request.

A6. The system verifies the appointment request, approves the appointment, and displays the results.

*3.1.2.1.5 Alternative Scenario:*

B2. The user can view the detail information about the selected the doctor, and the system shows the related information of the doctor.

C6. If the time of requesting appointment is conflicted with the doctor's agenda (overlapping), the system shows error message, and the patient re-books an appointment.

D6. If the requesting appointment is too close to the closing office hour (overlap with the closing office hour), the system shows error messages, and the patient re-books an appointment.

*3.1.2.1.6 Post-Condition*

The appointment updates the doctor's schedule properly.

*3.1.2.1.7 Non Functional Requirement*

Overlapping time  $\leq 5$  minutes is acceptable.

For example, supposed that the doctor's office hour is 9:00 to 5:00pm, if a patient requests an appointment from 4:20 to 5:05, this time overlapping ( $\leq 5$  minutes) is tolerable.

**3.1.2.2 Use Case O.H.A.S.S-Cancel-Appointment**

*3.1.2.2.1 Purpose*

This function enables the patients to cancel an appointment he has made.

*3.1.2.2.2 Actors*

The main actors are the patient.

*3.1.2.2.3 Pre-Condition*

The user has logged into the O.H.A.S.S. with a client account.

*3.1.2.2.4 Main Scenario*

A1. The user view the new appointment list first

A2. The user choose the appointment he or she wants to cancel.

A3. Click cancel. The system will check this record, and delete it from database.

*3.1.2.2.5 Alternative scenario*

Not available

*3.1.2.2.6 Post-Condition*

The appointment record is deleted from the system.

**3.1.2.3 Use Case O.H.A.S.S-Edit-Personal-Information**

*3.1.2.3.1 Purpose*

This function allows a registered patient to update his or her personal information, such as address, phone number etc.

*3.1.2.3.2 Actors*

The main actors are registered patients.



#### *3.1.2.3.3 Pre-Condition*

The user must login the system successfully.

#### *3.1.2.3.4 Main Scenario*

- A1. The user accesses the system and chooses “view and edit my profile” function to view and edit the user’s personal information.
- A2. The system gets the information of the login user, and displays the information to the user.
- A3. The user chooses the “edit my profile” function.
- A4. The system provides a form to user to edit the personal information data.
- A5. The user edits the data on the form and submits it.
- A6. The system check the validity of the user’s input. If the data is valid, the system returns the confirming information to the user.

#### *3.1.2.3.5 Alternative Scenario:*

- B6. If the user’s password and reentry password are not identical, the system shows an error message. Then the system displays the registration form, and the user goes to step A3.

#### *3.1.2.3.6 Post-Condition*

Not available

#### *3.1.2.3.7 Non Functional Requirement*

- Error message should be clear to the users.
- When the user is asked to refill out the form, the correct data that the user submitted last time should be there.

### **3.1.2.4 Use Case O.H.A.S.S-Create-Account**

#### *3.1.2.4.1 Purpose*

This function allows a new patient to create a user account and register with the system.

#### *3.1.2.4.2 Actors*

The main actors are new patients who have not registered to this system and do not have a valid user account.

#### *3.1.2.4.3 Pre-Condition*

Not available

#### *3.1.2.4.4 Main Scenario*

- A1. The user accesses the system and chooses the new user register function to create a user account.
- A2. The system displays a registration form, including Email address, name, password, health card number, phone number, address etc.
- A3. The user fills out the registration form, and submits it.
- A4. The system verifies the submitted information. If the data is valid, the system returns the confirming information to the user, shows the user a welcome message, the user account, and the password.

#### *3.1.2.4.5 Alternative scenario*

- B4. If the user’s email address has already existed in the system, the system shows an error message. Then the system displays the registration form, and the user goes to step A3.

C4. If the user's password and reentry password are not identical, the system shows an error message. Then the system displays the registration form, and the user goes to step A3.

#### *3.1.2.4.6 Post-Condition*

After a new patient register to the system successfully, the patient can use the user account and password to login the system and perform other tasks.

#### *3.1.2.4.7 Non Functional Requirement*

- Error message should be clear to the users.
- When the user is asked to refill out the form, the correct data that the user submitted last time should be there.

### **3.1.2.5 Use Case O.H.A.S.S-Consult-Appointments**

#### *3.1.2.5.1 Purpose*

This function enables the patients to view appointments he has made.

#### *3.1.2.5.2 Actors*

The main actors are the patient.

#### *3.1.2.5.3 Pre-Condition*

The user has logged into the O.H.A.S.S. with a client account.

#### *3.1.2.5.4 Main Scenario*

The user clicks the URL that links to this function pages. Both of the new appointment and historical appointments can be traced.

#### *3.1.2.5.5 Alternative scenario*

Not available

#### *3.1.2.5.6 Post-Condition*

Not available

### **3.1.2.6 Use Case O.H.A.S.S-Schedule-Special-Appointments**

#### *3.1.2.6.1 Purpose*

This use case allows the Administrator to schedule special appointments requested by the doctors especially for their patients (for e.g. Surgery, blood test etc.)

#### *3.1.2.6.2 Actors*

Main actor: Administrator

#### *3.1.2.6.3 Pre-Condition*

The Administrator must be authenticated.

He must have details of the patient, and the requirements for the appointment.

#### *3.1.2.6.4 Main Scenario*

The Administrator accesses the O.H.A.S.S. and chooses the Schedule-Special-Appointments option. Then:

- He chooses a specific department
- He chooses type of appointment
- The system returns available appointment slots
- He chooses a date from available dates
- He chooses a particular time slot from available time slots
- He fills in patient information

#### *3.1.2.6.5 Post-Condition*

The Administrator is able to fix special appointment for the patient. The patient receives an email regarding the details of the appointment.

### **3.1.2.7 Use Case O.H.A.S.S-Edit-Personal-Information**

#### *3.1.2.7.1 Purpose*

This functionality allows the doctors to manipulate their personal information.

#### *3.1.2.7.2 Actors*

The main actor in this scenario is a Doctor.

#### *3.1.2.7.3 Pre-condition*

- The account of doctor must exist
- Doctor must logged in O.H.A.S.S.
- O.H.A.S.S. should be on-line

#### *3.1.2.7.4 Main Scenario*

After the doctor has logged into O.H.A.S.S., s/he will choose the option “Modify Personal Information”. Doctors will be able to see their information such name, address, e-mail, phone number, and their schedule availability. They will have the opportunity to add a day off, a week off, vacations and some changes in their schedules.

#### *3.1.2.7.5 Alternative Scenario*

If the date doctors have chosen to vacations has already appointments the system will allow to select those date; however, they will have to go to the system and cancel the appointments for that period of time.

#### *3.1.2.7.6 Post-Condition*

Not available in this release. Release 2 may contain automatic cancellation at the time at day off is selected by the doctor.

### **3.1.2.8 Use Case O.H.A.S.S-Consult-Appointments**

#### *3.1.2.8.1 Purpose*

This functionality allows the doctors to select appointments based on different arguments.

#### *3.1.2.8.2 Actors*

The main actor in this scenario is a Doctor.

#### *3.1.2.8.3 Pre-condition*



The doctor must be authenticated.

#### *3.1.2.8.4 Main Scenario*

After doctors have logged into O.H.A.S.S., they will choose the option “Consult Appointments”.

Doctors will be able to see their appointments. They may choose:

- 1) Name and will go to Use Case O.H.A.S.S-View-Appointments-By-Name
- 2) Date and will go to Use Case O.H.A.S.S-View-Appointments-By-Date
- 3) All and will go to Use Case O.H.A.S.S-View-Appointments-All

#### *3.1.2.8.5 Alternative Scenario*

Not applicable

#### *3.1.2.8.6 Post-Condition*

Screen report displayed

### **3.1.2.9 Use Case O.H.A.S.S-View-Appointments-By-Date**

#### *3.1.2.9.1 Purpose*

This functionality allows the doctors to consult their appointments on a specific date.

#### *3.1.2.9.2 Actors*

The main actor in this scenario is a Doctor.

#### *3.1.2.9.3 Pre-condition*

1. The doctor must know the date and this date must be valid and the next format DD/MM/YYYY
2. The doctor must be authenticated.

#### *3.1.2.9.4 Main Scenario*

After doctors have logged into O.H.A.S.S. and have chosen “Consult Appointments by Date”, doctors will have to enter the date they want to consult. O.H.A.S.S. will show all appointments that this doctor has in that specific date.

#### *3.1.2.9.5 Alternative Scenario*

Not applicable

#### *3.1.2.9.6 Post-Condition*

Not applicable

### **3.1.2.10 Use Case O.H.A.S.S-View-Appointments-By-Name**

#### *3.1.2.10.1 Purpose*

This functionality allows the doctors to consult their appointments based on Patient’s name.

#### *3.1.2.10.2 Actors*

The main actor in this scenario is a Doctor.

#### *3.1.2.10.3 Pre-condition*

1. Doctor must know the complete name of the patient

2. The doctor must be authenticated.

#### *3.1.2.10.4 Main Scenario*

After doctors have logged into O.H.A.S.S. and have chosen “Consult Appointments by Name”, doctors will have to enter the complete Patient’s name that they want to consult. O.H.A.S.S. will show all appointments that this patient has even if they are with another doctor.

#### *3.1.2.10.5 Alternative Scenario*

Not Applicable

#### *3.1.2.10.6 Post-Condition*

Not Applicable

### **3.1.2.11 Use Case O.H.A.S.S-View-Appointments-All**

#### *3.1.2.11.1 Purpose*

This functionality allows the doctors to consult all their appointments.

#### *3.1.2.11.2 Actors*

The main actor in this scenario is a Doctor.

#### *3.1.2.11.3 Pre-condition*

The doctor must be authenticated.

#### *3.1.2.11.4 Main Scenario*

After doctors have logged into O.H.A.S.S. and have chosen “Consult ALL Appointments”, doctors will be able to see all the appointments that are pending.

#### *3.1.2.11.5 Alternative Scenario*

Not applicable

#### *3.1.2.11.6 Post-Condition*

O.H.A.S.S. will generate a report.

### **3.1.2.12 Use Case O.H.A.S.S-Cancel-Appointments**

#### *3.1.2.12.1 Purpose*

This use case allows the administrator to cancel an appointment for a patient

#### *3.1.2.12.2 Actors*

Main actor(s): Administrator

Secondary actor: N/A

#### *3.1.2.12.3 Pre-Condition*

The administrator must be authenticated.

#### *3.1.2.12.4 Main Scenario*

The administrator accesses the O.H.A.S.S. and chooses the Cancel-an-Appointment option. Then:

A1 He chooses a user’s name

A2 Chooses the specifies appointment to be cancelled

A3 The system asks for confirmation

A4 The administrator confirms

*3.1.2.12.5 Alternative scenario*

B2 The user chosen does not have an appointment

B3 The system informs the administrator.

*3.1.2.12.6 Post-Condition*

The doctor's schedule is in active and no further appointments possible with that doctor.

**3.1.2.13 Use Case O.H.A.S.S-Suspend-Doctor's-Account**

*3.1.2.13.1 Purpose*

This use case allows the administrator to make the account of a doctor inactive.

*3.1.2.13.2 Actors*

Main actor(s): administrator

Secondary actor: N/A

*3.1.2.13.3 Pre-Condition*

The administrator must be authenticated.

*3.1.2.13.4 Main Scenario*

The administrator accesses the O.H.A.S.S. and chooses the doctor-account-suspension. Then:

A1 He chooses a doctor's name.

A2 System asks for confirmation.

A3 The administrator confirms.

*3.1.2.13.5 Alternative scenario*

B2 The doctor chosen already has a suspended account

B3 The system returns the proper information message.

*3.1.2.13.6 Post-Condition*

The doctor's schedule is in active and no further appointments possible with that doctor.

**3.1.2.14 Use Case O.H.A.S.S-Create-Doctor's-Account**

*3.1.2.14.1 Purpose*

This use case allows the administrator to create a new doctor profile.

*3.1.2.14.2 Actors*

Main actor(s): administrator

Secondary actor: N/A

#### *3.1.2.14.3 Pre-Condition*

The administrator must be authenticated.

#### *3.1.2.14.4 Main Scenario*

The administrator accesses the O.H.A.S.S. and chooses the create-doctor-Account option. Then:

A1 He fills a form with a doctor's personal information.

A2 He submits the form.

A3 The system returns a unique employee username and a password.

#### *3.1.2.14.5 Alternative scenario*

B1 The doctor's SIN already exists.

B2 The system displays a proper error message.

#### *3.1.2.14.6 Post-Condition*

A new doctor-profile is created with a new unique username and a password assigned for the doctor.

### **3.1.2.15 Use Case O.H.A.S.S-Configure-Employee-x-Department**

#### *3.1.2.15.1 Purpose*

This functionality allows the administrator to add the department to the doctor's file in the clinic.

#### *3.1.2.15.2 Actors*

The primary actor in this scenario is an Administrator.

The secondary actor in this scenario is a doctor.

#### *3.1.2.15.3 Pre-condition*

1. Administrator must logged in O.H.A.S.S.
2. Doctor profile should be exist in O.H.A.S.S.
3. The department code should be exist in O.H.A.S.S.
4. O.H.A.S.S. should be on-line

#### *3.1.2.15.4 Main Scenario*

After the administrator have logged into O.H.A.S.S. and have chosen "Add Doctor to Department", the administrator will have to enter the doctor id and then assign the department.

#### *3.1.2.15.5 Alternative Scenario*

The administrator enters a non-existent doctor id. Then, The O.H.A.S.S. system prompts the administrator for the correct doctor id.

The administrator enters a non-existent department type. Then, The O.H.A.S.S. system prompts the administrator for the correct department.

The administrator enters an existent doctor id with a non-existent department type. Then, The O.H.A.S.S. system prompts the administrator to enter the department type first.

#### *3.1.2.15.6 Post-Condition*

O.H.A.S.S. will generate a report.

A department be assigned to each doctor

### **3.1.2.16 Use Case O.H.A.S.S-Configure-Clinic-Departments**

#### *3.1.2.16.1 Purpose*

This functionality allows the administrator to add the departments working in the clinic.

#### *3.1.2.16.2 Actors*

The main actor in this scenario is an Administrator.

#### *3.1.2.16.3 Pre-condition*

1. Administrator must logged in O.H.A.S.S.
2. O.H.A.S.S. should be on-line
3. Department should not be exist

#### *3.1.2.16.4 Main Scenario*

After the administrator have logged into O.H.A.S.S. and have chosen “Add New Department”, the administrator will have to enter the code and description of each department.

#### *3.1.2.16.5 Alternative Scenario*

An error message will inform the user if the department is existent.

#### *3.1.2.16.6 Post-Condition*

O.H.A.S.S. will generate a report  
Department will exists in the system

### **3.1.2.17 Use Case O.H.A.S.S-Configure-Department-x-Appointment**

#### *3.1.2.17.1 Purpose*

This functionality allows the administrator to add the appointment types for each department available in the clinic.

#### *3.1.2.17.2 Actors*

The main actor in this scenario is an Administrator.

#### *3.1.2.17.3 Pre-condition*

- Doctor must logged in O.H.A.S.S.
- The department code and the Appointment code should be exist in O.H.A.S.S.
- The Appointment type should have a duration exist in O.H.A.S.S.
- O.H.A.S.S. should be on-line

#### *3.1.2.17.4 Main Scenario*

After the administrator have logged into O.H.A.S.S. and have chosen “Add Duration for Appointment Type”, the administrator will have to enter the department code and the appointment code in order to add duration.

#### *3.1.2.17.5 Alternative Scenario*

The administrator enters a non-existent department id. Then, The O.H.A.S.S. system prompts the administrator for the correct department id.

The administrator enters a non-existent appointment type. Then, The O.H.A.S.S. system prompts the administrator for the correct appointment type.  
The administrator enters an existent department id with a non-existent appointment type. Then, The O.H.A.S.S. system prompts the administrator to enter the appointment type first.

#### *3.1.2.17.6 Post-Condition*

O.H.A.S.S. will generate a report.  
An appointment type(s) will be assigned to each department

### **3.1.2.18 Use Case O.H.A.S.S-Scheduling-Doctor's-Unavailability**

#### *3.1.2.18.1 Purpose*

This use case allows the administrator to specify the days and times for a certain doctor in which He can't receive any patients.

#### *3.1.2.18.2 Actors*

Main actor(s): administrator  
Secondary actor: N/A

#### *3.1.2.18.3 Pre-Condition*

The administrator must be authenticated.

#### *3.1.2.18.4 Main Scenario*

The administrator accesses the O.H.A.S.S. and chooses the schedule unavailability option. Then:

- A1 He chooses a doctor's name
- A2 The system returns the doctor's schedule
- A3 He selects the day and time at which the doctor is unavailable

#### *3.1.2.18.5 Alternative scenario*

- B2 The doctor chosen is inactive or suspended
- B3 The system should return the proper information message.

#### *3.1.2.18.6 Post-Condition*

The doctor's schedule is changed and will contain some time-slots of unavailability.

## **3.2 Performance Requirements**

### **3.2.1 Static Requirements**

- The System should be able to service multiple terminal connections simultaneously. At least 50 internet users should be able to use the system without any connection delay issues
- The system should process textual and numerical data

### **3.2.2 Dynamic Requirements**



- The software should be able to process more than 90% of the user queries (like reserving an appointment, modifying an appointment, etc.) in a fraction of 1 second.
- No Restriction is applied on the number of queries performed by a user as long as he is authorized to do perform them.
- The time needed to display information of users' queries on a web client, varies according to the system load and users' internet connection speed.
- For a specific time in a specific date, all users have same priority to reserve that slot. Users are served in a First-Come First-served fashion.

### **3.3 Logical Database Requirements**

- The database should store different data types of information like integers, time formats, date format, real numbers and strings.
- A fault tolerance and crash recovery plan should be implemented on the system by configuring automatic backup in the DBMS.
- The database should be able to handle up to 50 simultaneous user queries.
- The database design and relationships should be in a BCNF normal form to guarantee maximum space usage efficiency and a response time of less than three seconds.

*Class Diagram is in Appendix A.1 Figure 2*

### **3.4 Design Constraints**

- The system should work on any internet browser with GUI whether the underlying Operating System is Windows, Linux/Unix or Macintosh.
- Only the old obsolete command-line web browsers won't be able to connect to the system<sup>9</sup>.

### **3.5 Software System Attributes**

#### **3.5.1 Reliability**

The system should insure that the user actions are performed correctly as the user requires:

- RR01: The system's database should handle the maximum number of users (50) without failure.

---

<sup>9</sup> [http://www.w3schools.com/browsers/browsers\\_stats.asp](http://www.w3schools.com/browsers/browsers_stats.asp)

- RR02: If the user entered invalid data, the system should be able to recover and react robustly with the error, by sending a message back to the user
- RR03: The System should be set up on a dedicated computer machine for s 24-hours a day and 7-days a week availability.
- RR04: The system's backup plan should support the protection of the DB data and assure the functioning of the system after recovery of a crash
- RR05: The probability of failure less than 0.01%.
- RR06: Uptime of at least 99%.
- RR07: Less than 30 minutes needed to recover from system failure.

### **3.5.2 Availability**

- AR01: The system must be available 24/7.

### **3.5.3 Security**

- SR01: User authentication is needed to use the system. Users should not be able to create or delete reservations for other users. Thus, a track of each user's reservations is needed in order to apply these constraints.
- SR02: Personal information collected upon registration should not be revealed for the sake of privacy. A user needs only to know whether a certain time slot is reserved or not, without the need to reveal the patient's name.
- SR03: The password should be at least 8 characters, 1 Upper case, 1 lower case and 1 number.
- SR04: Website should use different techniques in order to have secure transfer of data to database
- SR05: All the user's data can't be sold or distributed to other entities without their previous approval

### **3.5.4 Maintainability**

- MR01: The client side scripting is of the system can be easily fixed and enhanced due to simplicity. The server side scripting is well maintainable also as all models and code is preceded by related design and this requirements document.
- MR02: Company will have a developed business plan.
- MR03: Company will have to have all the necessary personnel needed for its functioning.
- MR04: There have to be meeting each week to analyze how business is going. Everything has to be documented.



### **3.5.5 Usability**

The System's interface should be user-friendly:

- UR01: All web pages used by the users should consistent with standardized colors and fonts.
- UR02: The presentation of the web pages should be made friendly be provided the users with on-the-spot instructions concerning the step they are.
- UR03: In case of errors, help links should be provided or instructions for correct use.
- UR04: An HTML online documentation should be available and accessible to the users on all pages and steps of system usage.
- UR05: Interface elements (e.g. menus) should be easy to understand.
- UR06: The user should be able to learn to use a system in less than 30 minutes.
- UR07: Time required for registration less than 5 minutes.
- UR08: Error messages should explain how to recover from the error.
- UR09: Actions which cannot be undone should ask for confirmation.
- UR10: Responsive design should be implemented

### **3.5.6 Portability**

- PR01: Because the appointment scheduler is a web based system, it can be accessed by users having any operating system with GUI internet browser installed; thus insuring high portability and minimum system requirements.
- PR02: 100% of the client code is host independent.
- PR03: The PHP language used at the hosting machine is system independent.
- PR04: System can be work under Windows or Linux.
- PR05: A website has to be compatible with different popular web browsers (Google Chrome, Mozilla Firefox, Opera, Safari and Internet Explorer 8+).

### **3.5.7 Correctness**

- CR01: All algorithms implemented in the system should be correct; meaning they should be performed as required. The testing phase insures correctness of the software by trying all possible case and matching their output with the documentation.

### **3.5.8 Interoperability**

- IR01: Since the system relies on the web agents (browsers) in its interoperability, and all web agents interoperates with other software packages, then the system can work with other software. For example, it can provide an Acrobat Reader version of the documentation.

- IR02: The system should properly interoperate with the database (MySQL)

### **3.5.9 Scalability requirements:**

- SR01: In case of needing more bandwidth or disk space, the system should be prepared to those situations.
- SR02: When increasing the resources of the website, there can't be any penalty to response time or having more errors than usual.

## Appendix A

### A.1 Structural Model

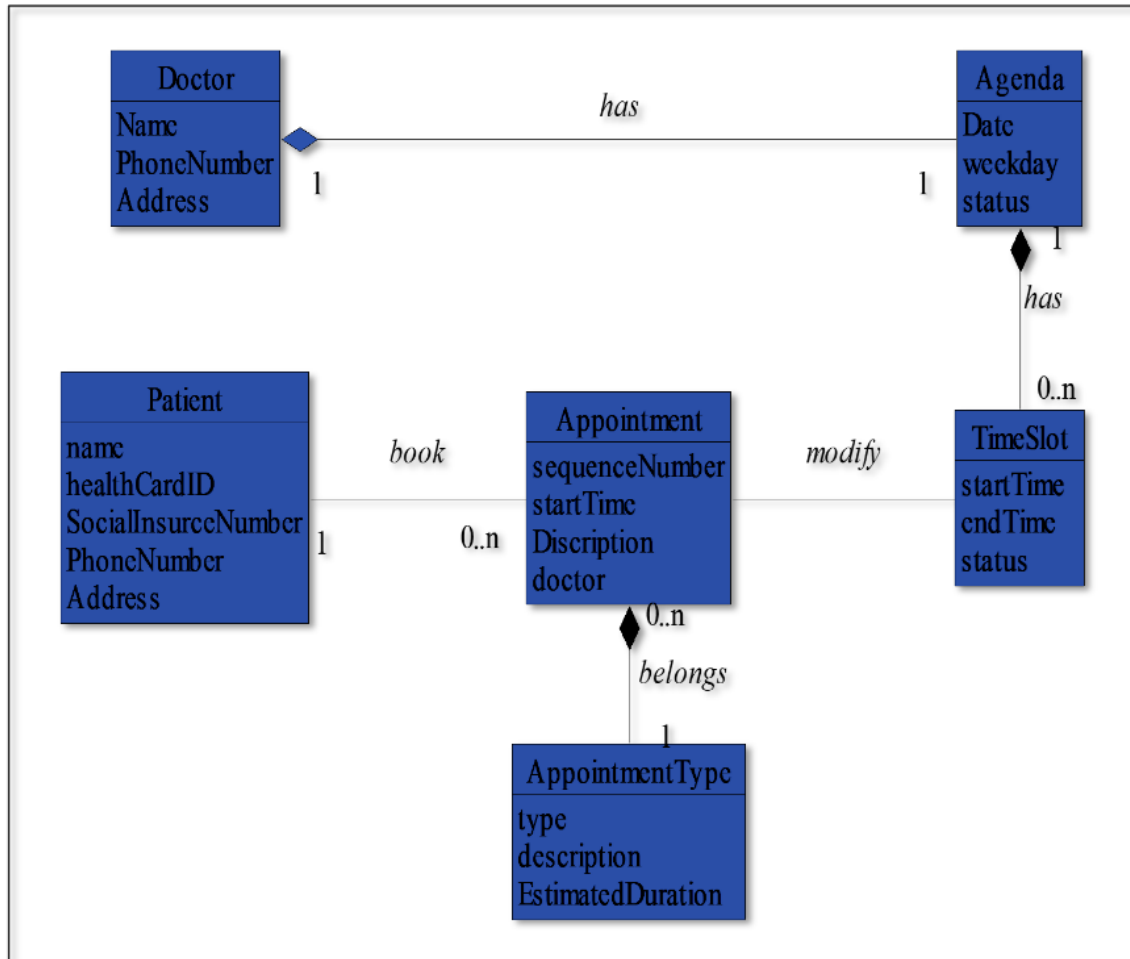


Figure 2-CLASS DIAGRAM OF O.H.A.S.S.

## A.2 BEHAVIORAL Model

### A.2.1 USE-CASE Model

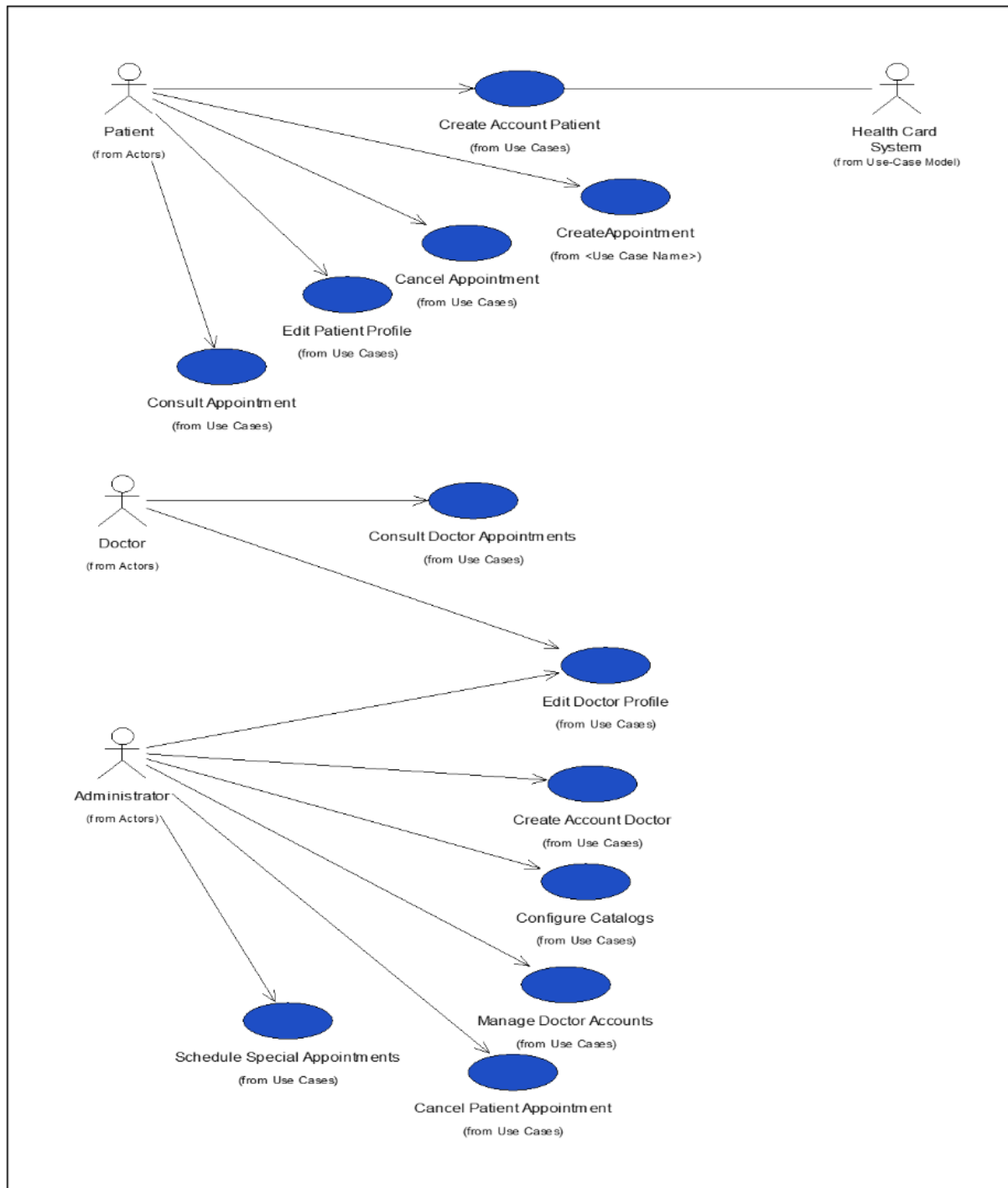


Figure 3-USE-CASE DIAGRAM

## ORIGINALITY REPORT

48%

SIMILARITY INDEX

42%

INTERNET SOURCES

5%

PUBLICATIONS

37%

STUDENT PAPERS

## PRIMARY SOURCES

1	Submitted to Universiti Malaysia Sarawak Student Paper	18%
2	juliepetrusa.com Internet Source	12%
3	Submitted to Franklin University Student Paper	6%
4	Submitted to Laureate Higher Education Group Student Paper	3%
5	Submitted to American Public University System Student Paper	2%
6	bart.sm.luth.se Internet Source	1%
7	mafiadoc.com Internet Source	1%
8	Submitted to Higher Education Commission Pakistan Student Paper	1%
9	www.coursehero.com	

Internet Source

1%

10

[link.springer.com](https://link.springer.com)

Internet Source

1%

11

[www.conceptdraw.com](https://www.conceptdraw.com)

Internet Source

1%

12

[jeff.fivefourteen.net](https://jeff.fivefourteen.net)

Internet Source

1%

13

Submitted to Oxford Brookes University

Student Paper

1%

Exclude quotes On

Exclude matches Off

Exclude bibliography On