Software Requirements Specification

Version 1.0

<<Annotated Version>>

September 11, 2021

**Online Hospital Management System**

|  |  |
| --- | --- |
| Name | ID |
| Md.Tahmid Alam | 20-42061-1 |
| Saiful Islam | 20-42585-1 |
| Md. Hafizur Rahman Nayan | 20-41845-1 |

**Submitted in Partial Portion of the Requirements of Software Engineering**

1. Introduction  
   1.1. Purpose

The purpose of this document is to present a detailed description of Online hospital management system. It will explain the purpose & features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the World Health Organization for its approval.

1.2. Scope of Project

This program will be an online hospital administration system. This system will be developed to optimize the user's convenience by giving tools to help automate the operation for obtaining medical facilities, which would otherwise have to be done manually. The system will fulfill the demands of the users and will be simple to learn and operate.

More specifically, this system is specially developed to allow a user to manage and communicate with a group of administrators. E-mail communication between users, physicians, and administrators will be facilitated by the program. Preformatted reply forms are used to offer a specific procedure at each level of the treatment's progression through the system; the position of these forms is changeable via the application's maintenance settings. A relational database with a list of users, physicians, and administrators is also included in the system.

1.3. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Active Information | The document that is tracked by the system; it is a narrative that is planned to be posted to the public website. |
| User | Person submitting information to be reviewed. In case of multiple users, this term refers to the *principal user*, with whom all communication is made. |
| Database | Collection of all the information monitored by this system. |
| Admin | Person who receives the information, sends information for review, and makes final judgments for treatment. |
| Field | A cell within a form. |
| Historical Society Database | The existing membership database (also HS database). |
| Doctor | A person that examines information and has the ability to recommend approval of the treatment for admin or to request that changes be made in the information. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Patient, Doctor an Admin. |

## 1.4. References

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

## 1.5. Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# 2.0. Overall Description

## 2.1 System Environment

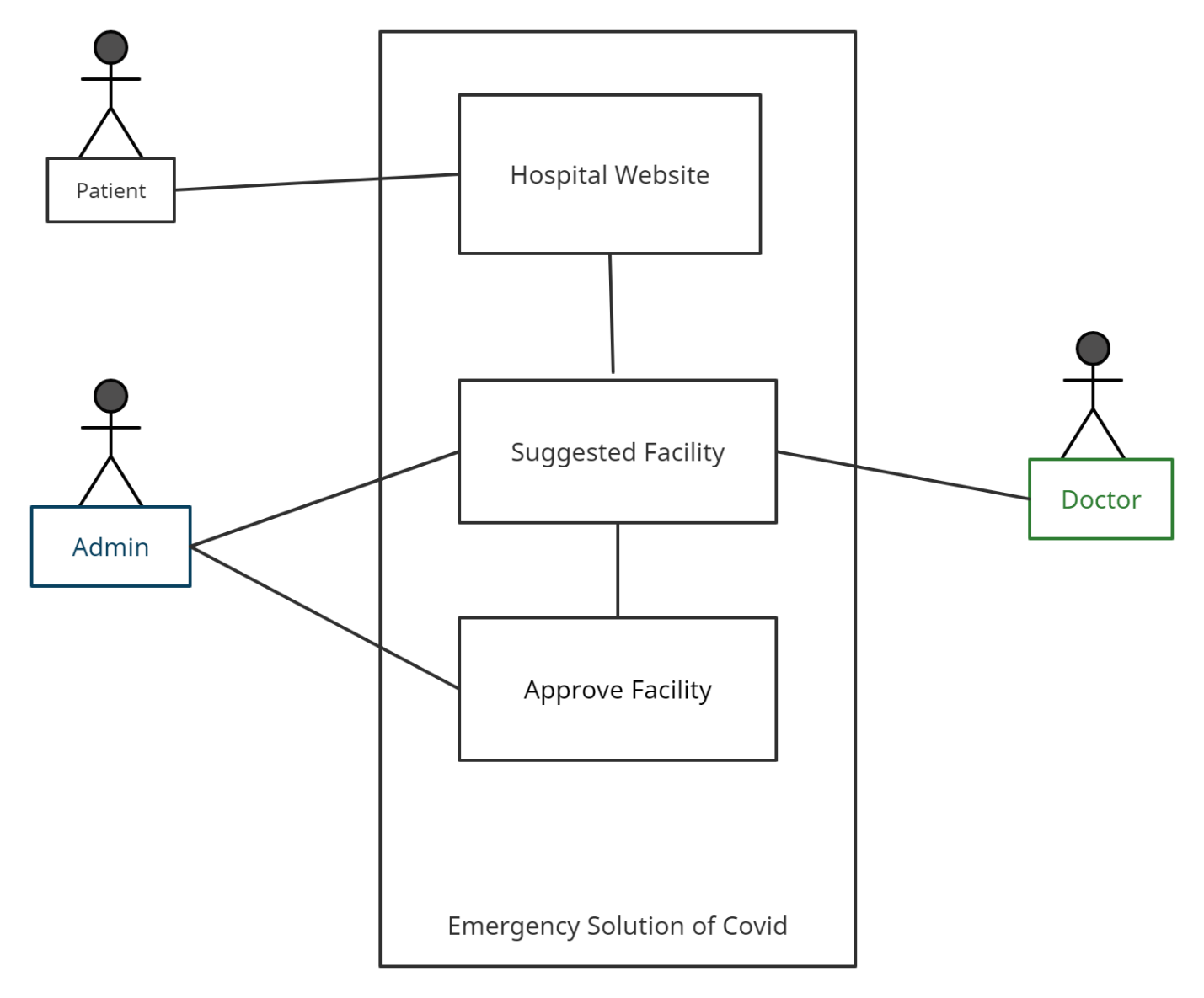


Figure 1 - System Environment

Emergency covid solution app has three active actors and one cooperating system.

The users, doctors and admins access the Online system through the Internet. Any users and doctors’ communication with the system is through email. The administrator accesses the entire system directly. There is a link to the Historical Society.

<< The division of the Emergency covid solution app into two component parts, the Online system and the doctors is an example of using domain classes to make an explanation clearer. >>

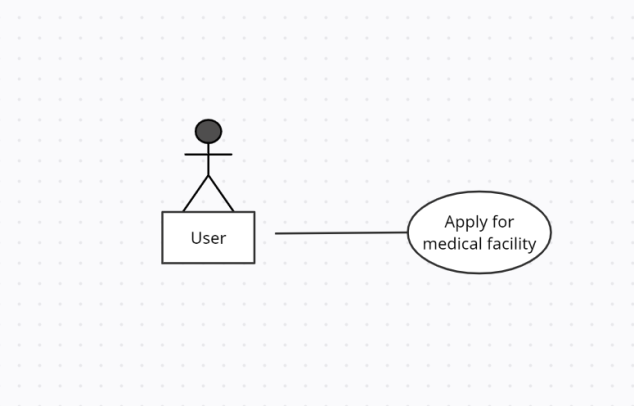
## 2.2 Functional Requirements Specification

This section outlines the use cases for each of the active readers separately. The users and doctors have only one-use case apiece while the admin is main actor in this system.

### 2.2.1 User Use Case

#### **Use case:** Apply for medical facility

**Diagram:**



**Brief Description**

The user accesses the Emergency covid solution apply for medical facilities.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already accessed the Emergency covid solution app.

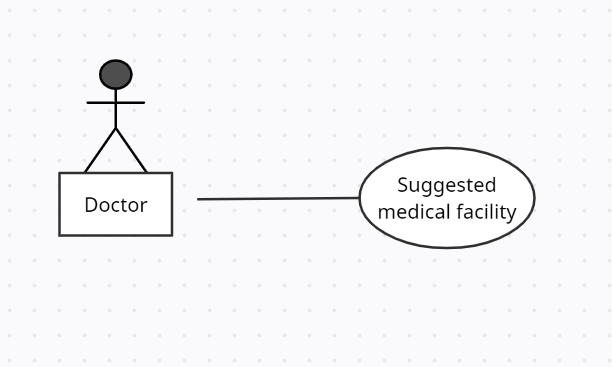
1. The user chooses to search by hospital name, category, or keyword.
2. The system displays the choices to the Reader.
3. The user selects the hospital desired.
4. The system presents the details of the hospital to the user.
5. The user chooses to download the details.
6. The system provides the requested hospital.

### 2.2.2 Author Use Case

In case of multiple doctors, this term refers to the *admin*, with whom all communication is made.

#### **Use case: Submit Article**

**Diagram:**



**Brief Description**

The doctor suggest the medical facility.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Doctor has already connected to the Emergency covid solution app.

The Doctor chooses the *Email Editor* button.

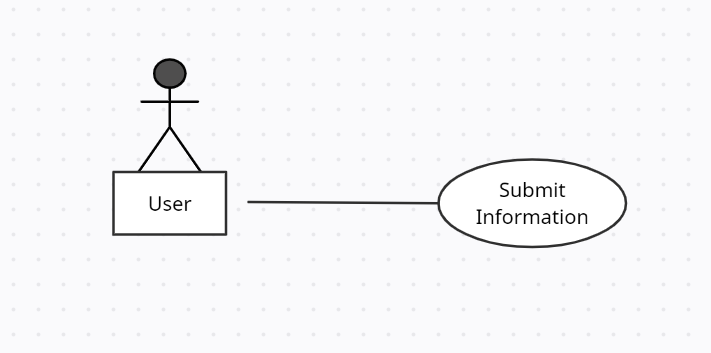
1. The System uses the *sendto* HTML tag to bring up the user’s email system.
2. The Doctor fills in the Subject line and attaches the files as directed and emails them.
3. The System generates and sends an email acknowledgement.

### 2.2.3 User Use Case

In case of multiple users, this term refers to the *principal user*, with whom all communication is made.

#### **Use case: Submit Article**

**Diagram:**



**Brief Description**

The user either submits Information or resubmits an edited information.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already connected to the Emergency covid solution app.

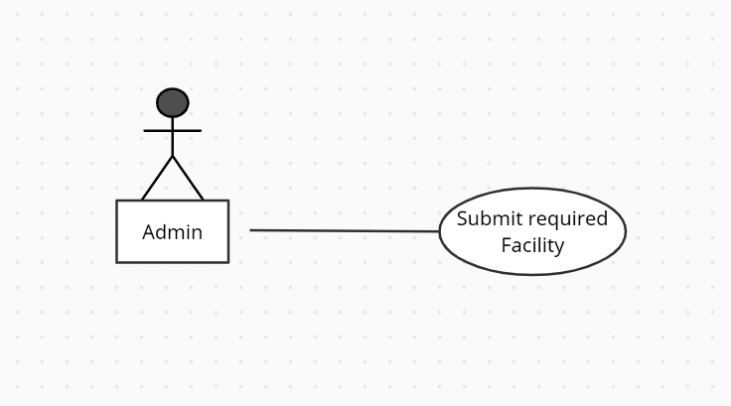
The Author chooses the *Email Editor* button.

1. The System uses the *sendto* HTML tag to bring up the user’s email system.
2. The user fills in the Subject line and attaches the files as directed and emails them.
3. The System generates and sends an email acknowledgement.

### 2.2.4 Admin Use Case

#### **Use case: Submit Required facility**

**Diagram:**



**Brief Description**

The admin submits a review of the information.

**Initial Step-By-Step Description**

Before this use case can be initiated, the admin has already connected to the Emergency covid solution app.

1. The admin chooses the *Email Editor* button.
2. The System uses the *sendto* HTML tag to bring up the user’s email system.
3. The admin fills in the Subject line and attaches the file as directed and emails it.
4. The System generates and sends an email acknowledgement.

### 2.2.5 Editor Use Cases

The Editor has the following sets of use cases:

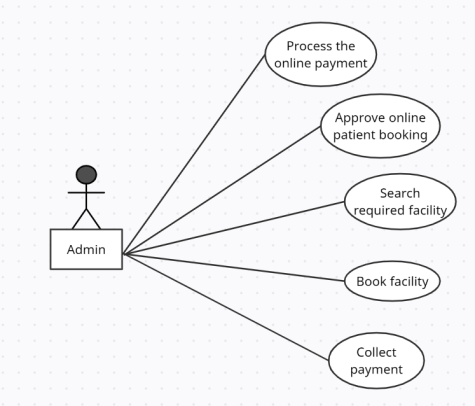
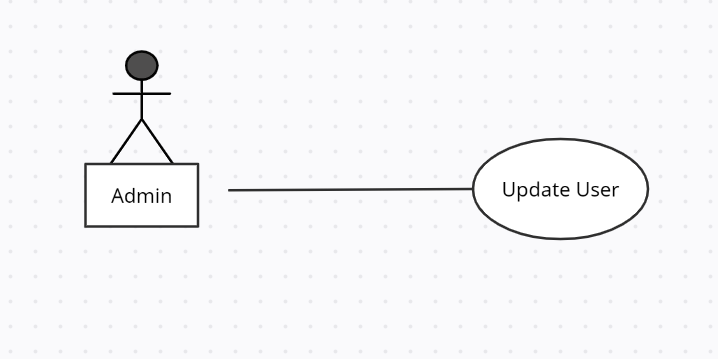


Figure 3 - Editor Use Cases

**Update Information use cases**

#### **Use case: Update Author**

**Diagram:**



**Brief Description**

The admin enters a new user or updates information about a current user.

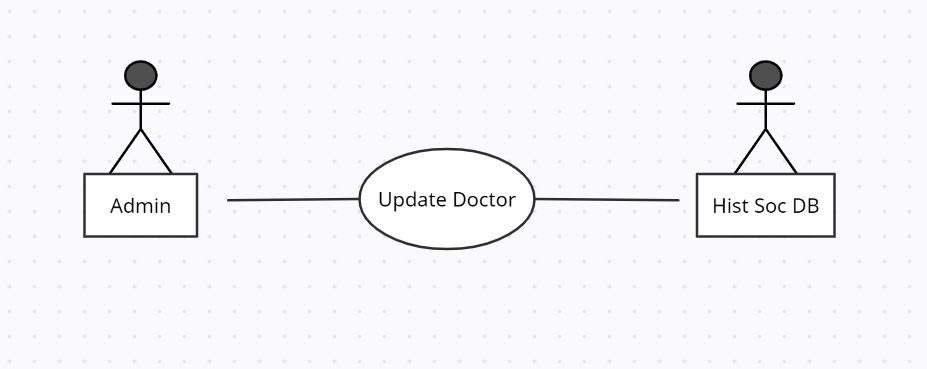
**Initial Step-By-Step Description**

Before this use case can be initiated, the admin has already accessed the main page of the Article Manager.

1. The admin selects to *Add/Update Author*.
2. The system presents a choice of adding or updating.
3. The admin chooses to add or to update.
4. If the admin is updating a user, the system presents a list of users to choose from and presents a grid filling in with the information; else the system presents a blank grid.
5. The admin fills in the information and submits the form.
6. The system verifies the information and returns the admin to the main page.

#### **Use case: Update doctor**

**Diagram:**



**Brief Description**

The admin enters a new doctor or updates information about a current doctor.

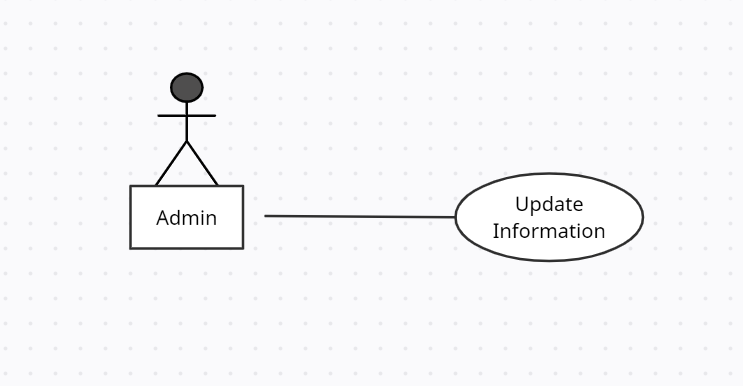
**Initial Step-By-Step Description**

Before this use case can be initiated, the Editor has already accessed the main page.

1. The admin selects to *Add/Update doctor*.
2. The system presents a choice of adding or updating.
3. The doctor chooses to add or to update.
4. The system links to the Historical Society Database.
5. If the doctor is updating a doctor, the system and presents a grid with the information about the doctor; else the system presents list of members for the editor to select a doctor and presents a grid for the person selected.
6. The admin fills in the information and submits the form.
7. The system verifies the information and returns the admin to the main page.

#### **Use case: Update Information**

**Diagram:**



**Brief Description**

The admin enters information about an existing user.

**Initial Step-By-Step Description**

Before this use case can be initiated, the admin has already accessed the main page

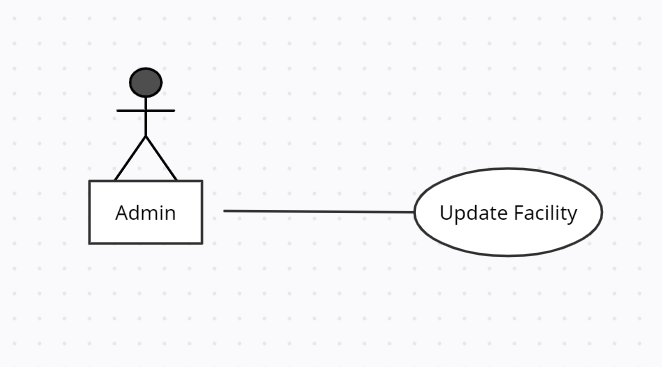
The Editor selects to *Update Article*.

1. The system presents s list of active users.
2. The system presents the information about the chosen users.
3. The admin updates and submits the form.
4. The system verifies the information and returns the admin to the main page.

#### **Use case: Update Facility**

This use case extends the *Update facility* use case.

**Diagram:**



**Brief Description**

The admin enters a facility into the system.

**Initial Step-By-Step Description**

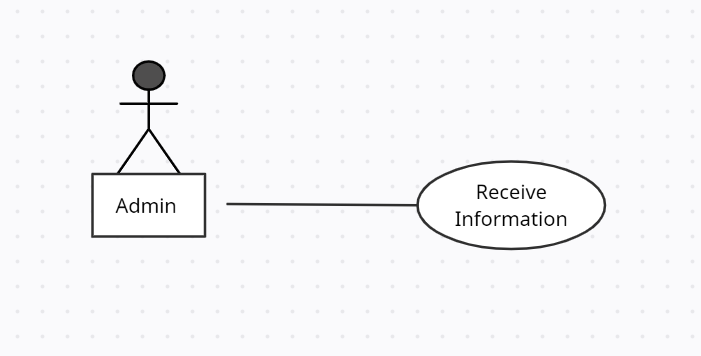
Before this use case can be initiated, the admin has already accessed the facility page using the *Update facility* use case.

1. The admin selects to *Receive facility*.
2. The system presents a grid for filling with the information.
3. The admin fills in the information and submits the form.
4. The system verifies the information and returns the admin to the main page.

**Handle Article use cases**

#### **Use case: Receive Information**

**Diagram:**



**Brief Description**

The admin enters a new or revised information into the system.

**Initial Step-By-Step Description**

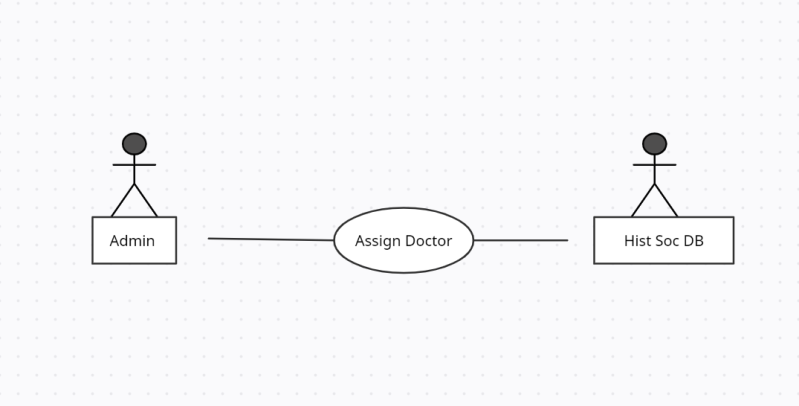
Before this use case can be initiated, the admin has already accessed the main page.

1. The admin selects to *Receive information*.
2. The system presents a choice of entering a new information or updating the existing information.
3. The admin chooses to add or to update.
4. If the admin is updating the information, the system presents a list of information to choose from and presents a grid for filling with the information; else the system presents a blank grid.
5. The admin fills in the information and submits the form.
6. The system verifies the information and returns the admin to the Main page.

#### **Use case: Assign Doctor**

This use case extends the *Update information* use case.

**Diagram:**



**Brief Description**

The admin assigns one or more doctors to a user.

**Initial Step-By-Step Description**

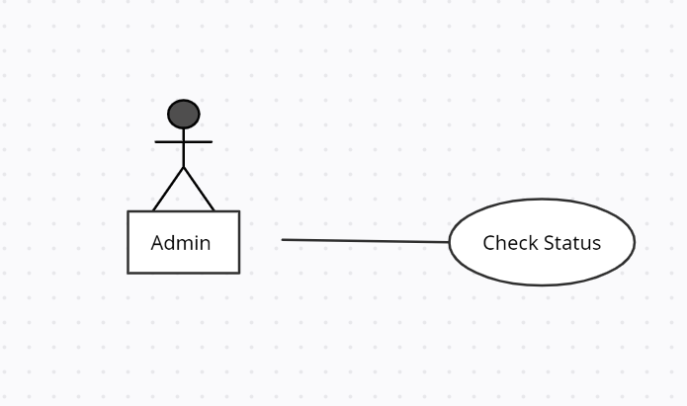
Before this use case can be initiated, the admin has already accessed the user using the *Update information* use case.

1. The admin selects to *Assign doctor*.
2. The system presents a list of doctors with their status.
3. The admin selects a doctor.
4. The system verifies that the person is still an active member using the Historical Society Database.
5. The admin repeats steps 3 and 4 until sufficient doctors are assigned.
6. The system emails the doctors, attaching the article and requesting that they do the review.
7. The system returns the admin to the *Update information* use case.

**Check Status use case:**

#### **Use case: Check Status**

**Diagram:**



**Brief Description**

The admin checks the status of all active users.

**Initial Step-By-Step Description**

Before this use case can be initiated, the admin has already accessed the main page.

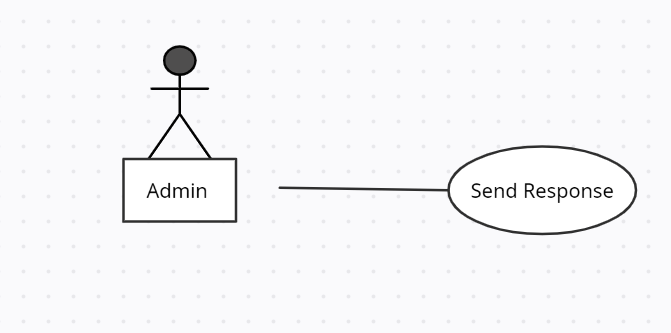
1. The admin selects to *Check Status*.
2. The system returns a scrollable list of all active users with their status.
3. The system returns the admin to the main page.

**Send Recommendation use cases:**

#### **Use case: Send Response**

This use case extends the *Update users* use case.

**Diagram:**



**Brief Description**

The admin sends a response to a user.

**Initial Step-By-Step Description**

Before this use case can be initiated, the admin has already accessed the user using the *Update user* use case.

1. The admin selects to *Send Response*.
2. The system calls the email system and puts the user’s email address in the Recipient line and the name of the article on the subject line.
3. The admin fills out the email text and sends the message.
4. The system returns the to the main page.

<< Since three of the actors only have one use case each, the summary diagram only involves the Editor. Adapt the rules to the needs of the document rather than adapt the document to fit the rules. >>

## 2.3 User Characteristics

The user is expected to be Internet literate and be able to use a search engine. The main screen of the emergency covid solution app will have the search function and a link to “Doctor/Hospital Information.”

The doctors are expected to be Internet literate and to be able to use email with attachments.

The admin is expected to be Windows literate and to be able to use button, pull-down menus, and similar tools.

The detailed look of these pages is discussed in section 3.2 below.

## 2.4 Non-Functional Requirements

The covid emergency solution app will be on a server with high-speed Internet capability. The physical machine to be used will be determined by the Historical Society. The software developed here assumes the use of a tool such as Tomcat for connection between the Web pages and the database. The speed of the user’s connection will depend on the hardware used rather than characteristics of this system.

# 3.0. Requirements Specification

## 3.1 External Interface Requirements

The only link to an external system is the link to the Historical Society (HS) Database to verify the membership of a user. The admin believes that a society member is much more likely to be an effective reviewer and has imposed a membership requirement for a doctor. The HS Database fields of interest to the Web Publishing Systems are member’s name, membership (ID) number, and email address (an optional field for the HS Database).

The *Assign doctor* use case sends the doctor ID to the HS Database and a Boolean is returned denoting membership status. The *Update doctor* use case requests a list of member names, membership numbers and (optional) email addresses when adding a new doctor. It returns a Boolean for membership status when updating a doctor.