3/14/2022

Software Requirement Specification
SRS (FYP-2022)

Sohail Ahmed Malik
SUKKUR IBA UNIVETSITY KANDHKOT CAMPUS

Final Year Project
Software Requirement Specification
For
COVID-19 Detection from X-Ray Images Using Self Attention Mechanism
(Bachelor of Sciences in Computer Sciences)

S#	Name	CMS ID	Mobile #	E-Mail
1	Sohail Ahmed Malik	023-18-0108	0304-2913054	Sohail.bcskkf18@iba-suk.edu.pk

Supervised by:
Mr. Syed Muzamil Hussain Shah _____

Co-Supervised by:
Mr. Noor Nabi Dahar _____

SUKKUR IBA UNIVERSITY
COMPUTER SCIENCE

Department of Computer Science
Sukkur IBA University Kandhkot Campus

Table of Contents

1. INTRODUCTION

1.1 Purpose

"**COVID-19 Detector**" is a web based system to get help in detecting COVID-19 by analyzing the X-ray images and using different deep learning models. It helps save time and save the cost of test kits, that are being too expensive. People are afraid of the going to test centers because of different kinds of rumors, so this platform can be a digital solution to this problem. Also to avoid different queues for hospital centers, this can be used to check whether the person is COVID Positive or negative, and then according to the results people should take preventive measures.

1.2 Document Conventions

Short Form	Complete Form	Description
COVID-19 Detector		Applications name
User		A person used the application
Responder		Our model which will tell the result of the Chest X-Ray.
WEB	Website	
Website		A website that operating by any operating system.
Developers		A person who build something
DB	Database	
DFD	Data flow diagram	
OS	Operating system	

1.2 Intended Audience and Reading Suggestions

The document is intended for doctors, researchers and students. Peoples just have to visit the portal and just have to create an account. The account will consist of different required fields such as name, address, location, contact number and email id and these accounts of the users will be visible to all the visitors of portal. On other hand patients can easily see all the profiles of doctors and can get consultation by contacting them through the contact number and email provided by doctors.

1.3 Product Scope

This product is useful for the patients, doctors also for the researchers, they can use it to get the results for the COVID-19 positives or negatives, and also the product will provide them a platform where they can contact with the registered doctors in order to get consultations regarding the COVID-19. Our product will also show the statistics of the COVID-19 tested using the web, this will also provide the preventive measures to the clients that are tested positive for COVID-19.

2. OVERALL DESCRIPTION

2.1 Project Background

COVID-19 has become a global pandemic issue, it has bad effects on the health, economy and population of the world. As Chest X-Ray is emerging as a valuable diagnostic tool for clinical management of COVID-19. Artificial intelligence (AI) has the power to detect and aid to the fastest evolution of X-Rays for detection of COVID-19 findings. So, we are working on the autonomous COVID-19 detection from X-Rays Images using well know models of deep learning, we will use multinational datasets and train those using different techniques and algorithms of deep learning. And we will implement this using the web Application. This will be helpful for the society, beneficial for the country and can reduce the cost of testing the people for COVID-19, because a simple detection application will tell the patients are either COVID-19 positive or not, also it will diagnose for lungs diseases. This is developed because almost 90% people are using mobile phones and they can easily access the web application and test themselves by giving an X-Ray image as input, the system will also detect other lungs diseases too.

AI has emerged in everyday life so that AI-based algorithms can readily identify X-Rays scans with COVID-19 associated any other diseases, as well as distinguish non-COVID-19 patients with high specificity in diverse patient populations.

2.2 Project Objectives

After the completion of this project we will be able to solve the problem of testing the COVID-19 Patients, this project also covers many other aspects including:

- Easy use of web application to test whether the person is COVID-19 positive or negative.
- Time saving, the most important feature of our project, because in the hospital centers there are many long queues that wait for their turn to go for the testing.
- Inexpensive web application compared to Test kit prices, our project can save the cost of test kits and can give the results accurately and comparatively better ones.
- 24/7 Availability anyone can use it anywhere.

2.3 System Constraints

a) Software Constraints

The major software constraints of project include time, cost, schedule, quality and scope.

b) Hardware Constraints

Processor cycles, to perform their tasks, actions/tasks must contend with other tasks and jobs in the system. Transactions give up their use of the processor at these points and must contend for use of the processor again when the activity has completed.

c) Cultural Constraints

Cultural Constraints of the project include language barriers or may be some users may be afraid of their privacy for sharing their X-Ray.

d) Legal Constraints

Legal constraints of the project include the use of the user's data in other aspect apart from the actual purpose of the data.

e) Environmental Constraints

Environment constraints include the gadgets on which this application can run and also the availability of internet service.

f) User Constraints

User constraints include the user's information about the project and training of using the web application, non-technical or lay men may face the issues.

2.4 Product Perspective

A distributed COVID-19 Detector database system stores the following information

2.4.1 User details

The user would be the doctors and patients they can communicate through this website. This website will store the information of doctors and patients.

2.4.2 Product Functions

Our product will perform the essential functions regarding the testing of COVID-19 from X-Ray images that includes login or sign up and other functions listed below:
The system performs the following functions.

- Creating an account at first log into system (Sign up)
- Log in
- Upload Image
- Select the Model
- Classify the Images
- Show Comparative Results
- Suggest Preventive Measures
- Log out

2.5 User Classes and Characteristics

2.5.1 Admin

Admin will have the all the authority of project. Admin will manage the all the module of application first admin will make sure the other user of the website. Admin will allow to update the system and can delete unnecessary things. Admin will add the model details etc.

.

2.5.2 Doctor

Doctors can create their account they will add the necessary information which will be provided from the admin side they can communicate with the patients (This is an additional module, if time remains I will add up this).

2.5.3 Patients

Patients can also sign up for their account creation and then they can perform the actions as per the module, like uploading the images and getting the results from specified classification algorithm.

2.6 Operating Environment

Operating environment for the patient and doctor is as listed below.

- Distributed database
- Client/server system
- Operating system: Any operating system.
- Database: My-Sql Database.

2.7 Design and Implementation Constraints

- SQL commands for above queries/application.
- Server Base use API to connect database with server.
- Centralized database system for website database queries.
-

2.8 User Documentation

COVID-19 Detector website can be use from any operating system user can simply go to any search engine and will enter the website address and then will use the website.

2.9 Assumptions and Dependencies

COVID-19 Detector have the following Assumptions and dependencies.

- Any device that could be mobile, PC and any tablet.
- Device that have any operating system.
- Device must have internet facility.

3. External Interface Requirements

These requirements include user interfaces (interaction logic between software and user), screen layouts, buttons, functions on every screen, hardware interfaces (here a team describes what devices the software is created for), and other relevant particularities.

3.1 User Interfaces

- Front-end software: Web browser
- Back-end software: Python (Django), My-Sql Database.

3.2 Hardware Interfaces

- Device (Support any Operating System).
- Device must be internet supported

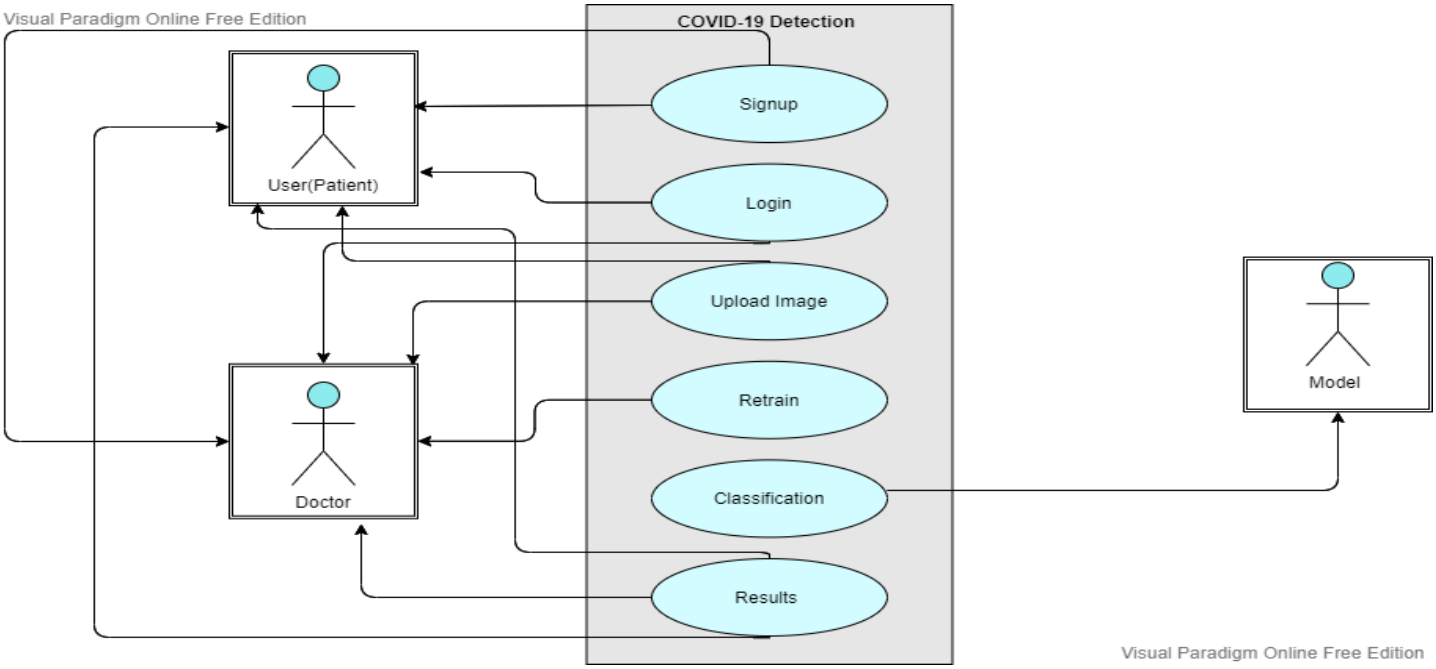
3.3 Software Interfaces

Following are the software used for the COVID-19 Detector Application

Software	Description
Operating system	Any operating system that could be windows. IOS, Android etc.
Database	To generate a record of user and responder and also the past incidents.

4.1 System Features

A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.



<Use case Id: Signup>		
Use case Id:	UC-01	
Actors:	User (Patient & Doctor)	
Feature:	User will click on the signup Button	
Pre-condition:	User will have to create an account	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Must fill with correct data	Verify username and password from Django user forms
2.	User click on the signup button	Allow user for account creation
Alternate Scenarios: Not Applicable		
Not Applicable		
Post Conditions		
Step#	**Description**	
1	After signup, user can be redirected to login page	
2	From there he can access different features like, uploading image and classifying cancer type. User can also save the image and load the saved images from the database or Django file storage system.	
Use Case Cross referenced	None	

<Use case Id: Login>		
Use case Id:	UC-02	
Actors:	User (Patient & Doctor)	
Feature:	User will click on the Login Button	
Pre-condition:	User must have an account	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Must fill with correct data	Verify username and password from Django user forms
2.	User click on the login button	Allow user for login
Alternate Scenarios: Not Applicable		
Not Applicable		
Post Conditions		
Step#	**Description**	
1	After login, user can be redirected to dashboard	
2	From there he can access different features like, uploading image and classifying cancer type. User can also save the image and load the saved images from the database or Django file storage system.	
Use Case Cross referenced	None	

<Use case Id: Upload Image>		
Use case Id:	UC-03	
Actors:	User (Patient & Doctor)	
Feature:	User will click on the Upload Button	
Pre-condition:	User must be logged into an account	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Upload the required image format	Verify username and password from Django user forms
2.	User click on the submit button	Allow user for uploading image
Alternate Scenarios: Not Applicable		
Not Applicable		
Post Conditions		
Step#	**Description**	
1	After uploading image, user can be redirected to display board for results.	
Use Case Cross referenced	None	

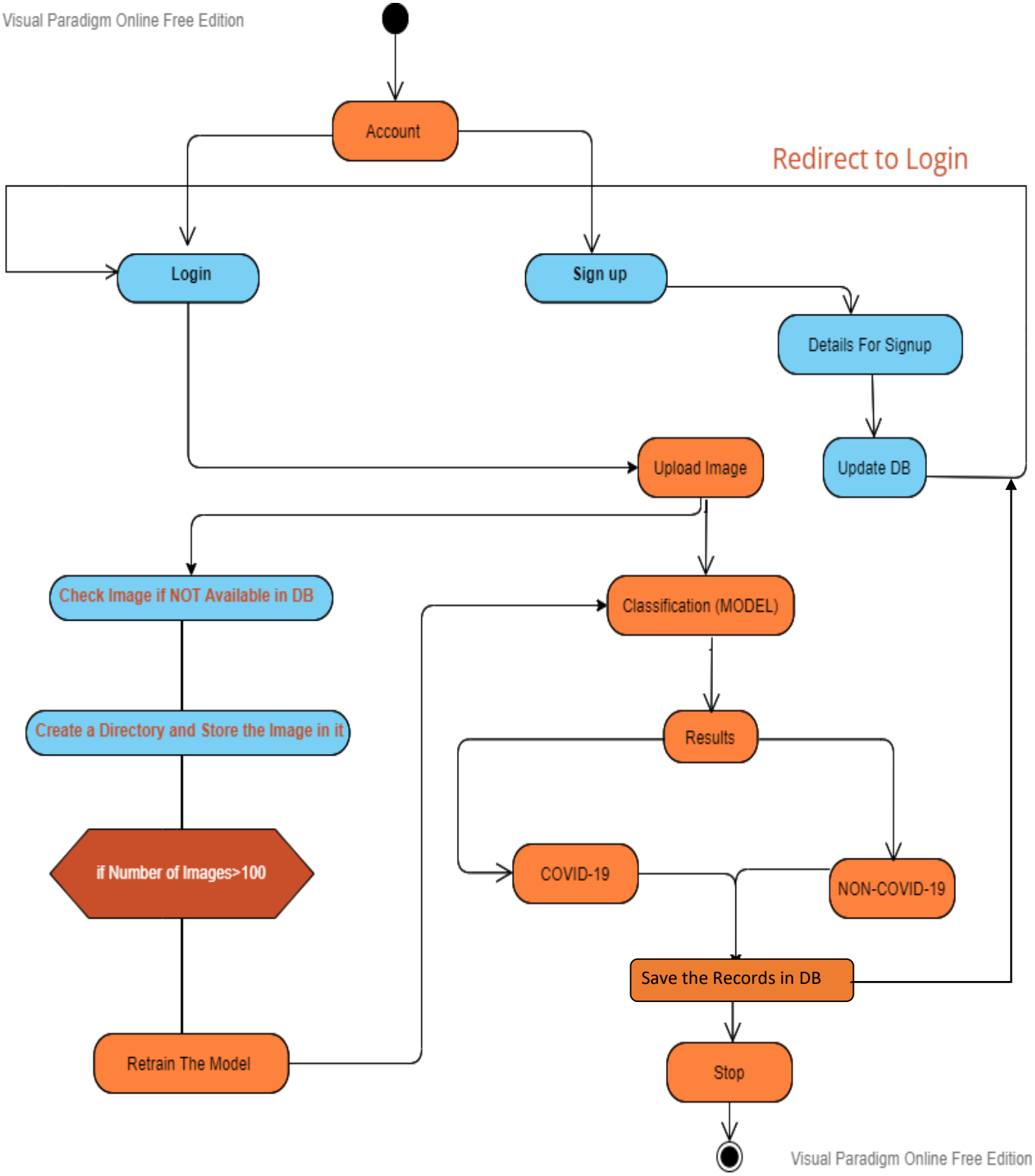
<Use case Id: Retrain>		
Use case Id:	UC-04	
Actors:	User (Patient & Doctor)	
Feature:	Limited users will be able to click on the button Retrain	
Pre-condition:	User must upload the image	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Upload the required image format	Verify image availability and formats rom DB.
2.	User click on the retrain button	Allow user for uploading image
Alternate Scenarios: Not Applicable		
Not Applicable		
Post Conditions		
Step#	**Description**	
1	After uploading image, user can be redirected to display board for results.	
Use Case Cross referenced	None	

<Use case Id: Classification>		
Use case Id:	UC-05	
Actors:	Model	
Feature:	Model will classify the image as per the training	
Pre-condition:	Model must be trained in order to classify the image	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Connect the model with the Module weights.	Not Applicable
2.	User click on the classify button	
Alternate Scenarios: Not Applicable		
Not Applicable		
Post Conditions		
Step#	**Description**	
1	After classification, user can be redirected to display board for results.	
Use Case Cross referenced	None	

<Use case Id: Results>		
Use case Id:	UC-06	
Actors:	User (Patient & Doctor)	
Feature:	User will click on the Upload Button	
Pre-condition:	User must be logged into an account User must have uploaded the image	
Scenarios		
Step#	**Action**	**Software Reaction**
1.	Upload the required image format	Show the results as per the model classification
2.	User click on the submit button	Allow user for uploading image
Alternate Scenarios: Not Applicable		
Not Applicable		
Post Conditions		
Step#	**Description**	
1	After uploading image, user can be redirected to display board for results.	
Use Case Cross referenced	None	

4.3 Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram. Activities modeled can be sequential and concurrent.



4.1.1 Description and Priority

User will sign up to the website and then can login to the system after login the system they will make their profile (In case of Doctors) and they can also respond to the patients. The other user will be the patient or normal user they can test themselves by giving an input of the chest x-ray image.

4.1.2 Stimulus/Response Sequences

- User will open website.
- Then they will have to login/signup for the account.
- Then they have to upload the image and wait for seconds to see their results.
- Then the user will be shown the preventive measures if they are COVID-19 positive.

4.4.3 Functional Requirements

- The system allows the Patient/Doctor to get register through this portal.
- Users can easily be get registered by creating their account.
- The system allows the users to get register through sending basic details.
- The system shows the status of user. (COVID-Tested or Not)
- System allows the patients and doctors to visit the web application anytime from anywhere without any kind of restriction.
- System allows the patients to see the preventive measures if they are COVID-19 positive.
- System provides the information of visitors of portal.

5. Other Nonfunctional Requirements

5.1 Some typical non-functional requirements are:

1. **Performance** – for example Response Time, Throughput, Utilization, Static Volumetric

   The quality and efficiency at which a Web application functions. Many factors can impact application performance, including bandwidth capacity, the number of users on a network, application protocols and coding, and attacks that exploit specific application vulnerabilities.

2. **Scalability**

   A scalable web application is a website that is able to handle an increase in users and load, whether in terms of a gradual or abrupt surge, without disrupting end-users' activities

3. **Capacity**

   Units such as bandwidth, database size, memory, disk usage, and server load all play a big role when it comes to capacity planning.

4. **Recoverability**

   How fault tolerance is the web application, this comes in affect when there is a huge number of the traffic and a sudden security threat.

5. **Maintainability**

   Maintainability is most commonly referred to as “the ease in which a system (web application) can be modified or extended

6. **Serviceability**

   When it comes to websites, serviceability, aka maintainability, implies their ability to be quickly, easily and safely:

   - Modified.

- Extended.
- Updated.
- Fixed.

5.2 Performance Requirements

System requires an operating system to perform efficiently. System is designed in a way that any key process will not delay the system. Application is considered to perform in real time.

5.3 Safety Requirements

Application ensures that notification and other data must not be touched. No modification in data can be changed. More over function of the system is to perform in every situation. In any error the system should response and give appropriate message

5.4 Security Requirements

The user information is stored in the system database. Database information cannot be achieved or accessed by the other user and not be threatened by any external interface. The user must have identity level.

5.5 Software Quality Attributes

Following are software quality attribute

- **Reliability**

  Reliability of an application is its probability of failure-free software operation for a specified period of time in a specified environment.

- **Usability**

  In Websites and software applications, usability has been defined as the ease at which an average person can use the software or website to achieve specific goals. Usability is comprised of learnability, memorability, efficiency, satisfaction and errors

- **Security**

  A web application firewall or WAF helps protect a web application against malicious HTTP traffic. By placing a filtration barrier between the targeted server and the attacker, the WAF is able to protect against attacks like cross site forgery, cross site scripting and SQL injection.

References

1. https://www.kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets
2. https://data.mendeley.com/datasets/8h65ywd2jr/3
3. https://www.hindawi.com/journals/mpe/2021/3281135/
4. https://covid19.who.int/.
5. https://www.who.int/news-room/fact-sheets/detail/pneumonia.