

PDSA system

SRS documentation



June 1, 2019

FUCLTY OF ENGINEERING

HELWAN

**Software Requirements Specification**

**For**

**Patient Doctor Smart Assistance**

Prepared by:

* **Basma Hassan Hamed**
* **Diaa Mohamed Sayed**
* **Eriny Tadros Hanna**
* **Menna ali Saeed**
* **Paula Ehab Alfy**

Faculty of Engineering, Helwan University

1-6-2019

Contents

[1. Preface: 4](#_Toc6071721)

[1.1 Document Purpose: 4](#_Toc6071722)

[1.2 Target Users: 4](#_Toc6071723)

[1.3 Revision History: 4](#_Toc6071724)

[2. Introduction: 4](#_Toc6071725)

[2.1 Purpose: 4](#_Toc6071726)

[2.2 Scope: 5](#_Toc6071727)

[2.3 Overview: 5](#_Toc6071728)

[3. Glossary: 5](#_Toc6071729)

[3.1 Acronyms, definitions, and abbreviations: 5](#_Toc6071730)

[4. System Users: 6](#_Toc6071731)

[4.1 System stakeholders: 6](#_Toc6071732)

[4.2 Users objectives : 7](#_Toc6071733)

[5. User Requirements definitions: 7](#_Toc6071734)

[5.1 System Functions: 7](#_Toc6071735)

[5.2 Constraints: 8](#_Toc6071736)

[6. System Architecture: 8](#_Toc6071737)

[7. System Functional requirements: 8](#_Toc6071738)

[7.1 Registration: 8](#_Toc6071739)

[7.2 Add patient’s information: 8](#_Toc6071740)

[7.3 Add medical report: 9](#_Toc6071741)

[7.4 View medical report: 9](#_Toc6071742)

[7.5 Delete medical reports: 9](#_Toc6071743)

[7.6 Update medical reports: 9](#_Toc6071744)

[7.7 Send notifications: 9](#_Toc6071745)

[7.8 View patient’s profile: 9](#_Toc6071746)

[7.9 Write prescription for patient: 9](#_Toc6071747)

[7.10 Set next appointment date: 9](#_Toc6071748)

[7.11 Add test result: 9](#_Toc6071749)

[7.12 Set surgery test: 9](#_Toc6071750)

[7.13 Get results from machine learning app about his X-ray: 9](#_Toc6071751)

[7.14 Add surgery result: 9](#_Toc6071752)

[8. Interface requirements: 10](#_Toc6071753)

[8.1 User interfaces: 10](#_Toc6071754)

[8.2 Software Interfaces: 10](#_Toc6071755)

[8.3 Communications Interfaces: 10](#_Toc6071756)

[9. Non-functional requirements: 10](#_Toc6071757)

[9.1 Availability: 10](#_Toc6071758)

[9.2 Security: 10](#_Toc6071759)

[9.3 Automation testing: 10](#_Toc6071760)

[10. System Models and Diagrams: 11](#_Toc6071761)

[10.1 Use case Diagram: 11](#_Toc6071762)

[10.2 Class Diagram: 11](#_Toc6071763)

[10.3 Sequence Diagram: 11](#_Toc6071764)

[10.4 State Diagram: 11](#_Toc6071765)

[11. System Evolution: 11](#_Toc6071766)

[12. Time Plan 11](#_Toc6071767)

[12.1 Work Breakdown Structure 11](#_Toc6071768)

[12.2 Gantt chart: 12](#_Toc6071769)

# Preface:

## Document Purpose:

The purpose of this document is to provide a detailed and complete PDSA system.

The document will provide an overview of the system in the first section; then each part will be

Explained in detail in the second and third sections.

## Target Users:

This document is composed by system engineers based on the requirements gathered from hospitals, doctors and patients.

This document is intended to be approved by manger and the staff working on developing the system.

## Revision History:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Description** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction:

## Purpose:

PDSA system aim to the communication between patients and their doctor easily, to help achieve the following:

* Doctors can keep track the medical reports of their patients easily.
* Patients can save their previous medical reports.
* Patients can know more information about their diseases.
* Doctors can know the previous medical reports for their patients.
* Patients are alarmed with the next Doctor's appointment, surgery time or Schedule of medicines.
* Patients can know recommended doctors or hospitals.
* Medical laboratory can communicate easily with patients.

## Scope:

PDSA (patient and doctor smart assistant) is system for managing patient’s medical reports which allow doctors to get the patient’s medical reports and track their medical history.

System functionalities:

* System can detect diseases from patient’s medical test through machine learning algorithm.
* Doctors can get patient medical reports by patient’s id.
* Access patient's information.
* Know the previous doctors who examined the patient.
* Know patient’s previous prescription which they had taken.
* Know the previous medical examination and medical reports of patient.
* The system remind the patient with the next required examination.
* The system remind the patient to do medical tests requested by the doctor.
* Medical laboratory can send test result to the patient.
* Hospital’s doctors can remind the patient with surgery time.

## Overview:

This document is organized as follows: first, an Overview description of PDSA systemand its high-level functions are presented (section 2.1 and 2.2). Section 4 states types of users who can use PDSA system. Then a list of general constraints that should be followed, assumptions and dependencies are presented in sections 5. Section 7 in the document provides a detailed description of the system functions and requirements. Section 10 presents some helping information and diagrams that will facilitate the understanding of this document.

Finally, Section 11 shows future work that should be done.

# Glossary:

## Acronyms, definitions, and abbreviations:

PDSA system: patient and doctor smart assistant system.

MySQL: My Structured Query Language: -

***MySQL is a database management system.***

PDO: PHP Data Objects: -

PDO has a much nicer interface, you will end up being more productive, and write safer and cleaner code. Instead of concatenating escaped strings into SQL, in PDO you bind parameters which is an easier and cleaner way of securing queries.

HTTP: Hypertext Transfer Protocol:-

The primary technology protocol on the web that allows linking and browsing this is the technology used to communicate between web servers and web users.

# System Users:

## System stakeholders:

* System Engineer :
* Responsible for requirements gathering.
* Responsible for development.
* Responsible for deployment and support.
* Patient:
* Write his\ her personal information
* Upload his\ her medical reports
* View , add, delete , update his report
* Get results from machine learning app about his X-ray
* View his prescription with its scheduled time
* get notification with the next Doctor's appointment
* Doctor:
* View patient’s profile
* Write prescription for patient
* Set next appointment date
* Medical laboratory:
* view patient’s profile
* Add test results
* Hospital’s doctor:
* View patient’s profile
* Write prescription for patient
* Set next appointment date
* Set surgery(operation) appointment
* Set surgery tests
* Add surgery results

## Users objectives :

* System Engineer:
* Gain Experience in software engineering and development.
* Patients:
* Save their previous and next medical report
* Can communicate easily with their doctors and Medical laboratory.
* Can know more information about their diseases.
* Get reminder for their next Doctor's appointment and Schedule of medicines.
* Get test results easily.
* Doctors :
* Know more information about their patients and keep track of their Therapeutic Period.
* Medical laboratory:
* Communicate easily with patients.
* Hospital’s doctor:
* Communicate easily with patients.
* Send reports and surgery’s reports to patients easily.

# User Requirements definitions:

## System Functions:

1. Registration.
2. Add patient’s information.
3. Add medical reports.
4. View medical reports.
5. Update medical reports.
6. Send notifications.
7. View patient’s profile.
8. Write prescription for patient
9. Set next appointment date
10. Add test result.
11. Set surgery test.
12. Get results from machine learning app about his X-ray.
13. Add surgery result.

## Constraints:

* Speed of server is low because it is free.
* Limited resources like ram and processor.
* Domain extension is not famous one like.org & .com but is .ml because it is free.
* http is not suitable for webcam using.

# System Architecture:

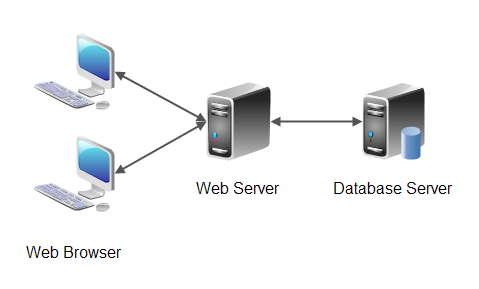


Figure 1 System Architecture

# System Functional requirements:

## Registration:

The patient make a registration with (username, name, age, birthday, email, password ) and the system generate 3ids for each of laboratory , doctor and hospital doctor.

## Add medical report:

Doctor, laboratory and hospital’s doctor they can add medical report for patient and also the patient himself can add his own old medical reports

## View medical report:

Doctors, laboratory and hospital’s doctor can view patient’s medical reports with entering the patient’s code that corresponding to each one of them and also the patient can view all his medical reports whatever it had been uploaded by who

## Update medical reports:

Only the patient can update his\ her medical report.

## Send notifications:

The patient receive notification if the doctor send for him note, message or alarming for the time of medicine, doctors’ appointment or surgery.

## View patient’s profile:

Doctors and medical lap can view patient’s information by entering patient’s username and id.

## Write prescription for patient:

Doctors can write prescription for patient by sending message or as a note for him/ her with entering the patient username and id that he/ she give them it.

## Set next appointment date:

Doctors can set next appointment date at date icon and patient can receive it as a note then he \ she get notification with the time.

## Add test result:

Medical laboratory sends attached to the patient as note or message by entering patient’s username and id then the patient upload it and add it to his\her medical reports.

## Set surgery test:

Doctors write the needed test for surgery to patient by sending message or as a note for him/ her, then he\she will get reminder for it.

## Get results from machine learning app about his X-ray:

Patient add his X-ray to the app then he\she get the result.

## Add surgery result:

Hospital’s doctors send attached to the patient as note then the patient upload it and add it to his\her medical reports.

# Interface requirements:

## User interfaces:

## Software Interfaces:

* Database access will be hosted by our free hosting server that using MySQL. Software that mange the database.

## Communications Interfaces:

* Users will access the website through http.
* Hosting server.

# Non-functional requirements:

## Availability:

* Traffic bandwidth unlimited.
* Our server UP 24/7.

## Security:

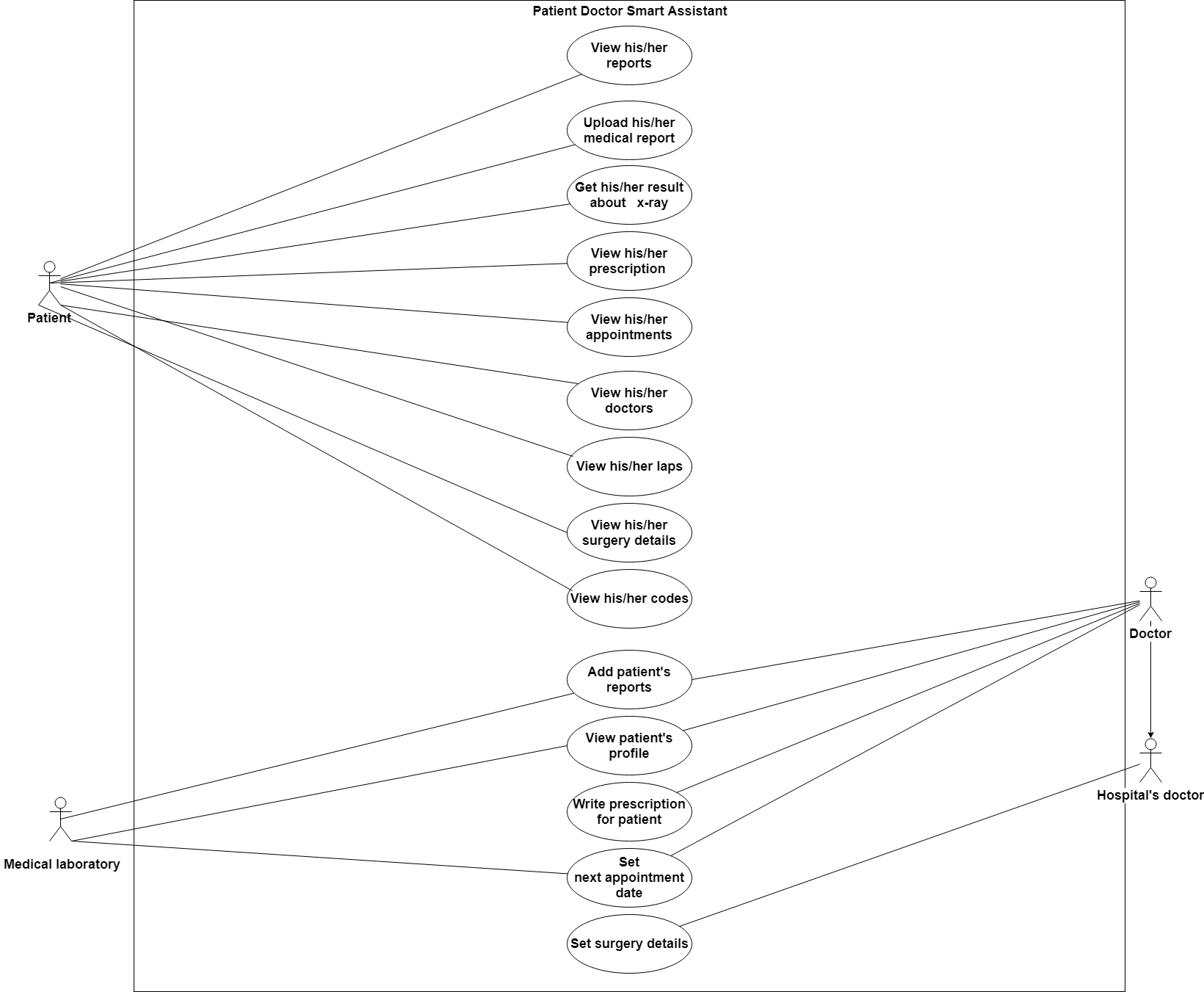
* The personal details can’t be access without account that be given by the patient.
* The password will be encrypted in the database.
* The input in the site was secured, filtered and validated from any malicious codes (SQL injection).
* Using PDO queries instead of MySQL queries.

## Automation testing:

* We used IMACROS addon handled by JAVASCRIPT to test all our site's pages to make sure that every entity do it's function well.
* It provide us with screenshots for whole site's pages that documented in SRS.

# System Models and Diagrams:

## Use case Diagram:



## Use Case Scenarios

1. Use Case: view his/ her reports(UC\_ID1)

* Participating Actors: Patient
* Entry Conditions : Patient is logged into the website
* Exit Conditions : Patient has received an acknowledgement from the system that the selected transaction is complete, or if not complete, a message explaining the failure
* Quality Requirements: (Performance) Patient receives a response from the system in less than 3 seconds
* Typical flow of events:

1. Patient selects View my reports

- System invokes View my reports use case and retrieve all the reports belong to this patient from the database

1. Use Case: upload his/ her reports(UC\_ID2)

* Participating Actors: Patient
* Entry Conditions : Patient is logged into the website
* Exit Conditions : Patient has received an acknowledgement from the system that the selected transaction is complete, or if not complete, a message explaining the failure
* Quality Requirements: (Performance) Patient receives a response from the system in less than 3 seconds
* Typical flow of events:

1. Patient selects Upload my reports

- System invokes upload my reports use case

1. Patient upload the report

- System validate the uploaded file

- For a valid response, system upload the file to the database and sends an acknowledgment back to the patient

* Exceptions :

1. Patient upload the report

* Invalid response received, so system reports failure with a message indicating invalid uploaded file

1. Use Case:(UC\_ID3)
2. Use Case: view his/ her prescription (UC\_ID4)

* Participating Actors: Patient.
* Entry Conditions: Patient is logged into the website.
* Exit Conditions: Patient has received an acknowledgement from the system that the selected transaction is complete, or if not complete, a message explaining the failure.
* Quality Requirements: (Performance) Patient receives a response from the system in less than 3 seconds.
* Typical flow of events:
  + - 1. Patient selects View my prescription

- System invokes View my prescription use case and retrieve all the prescription belong to this patient from the database.

1. Use Case: view his/ her appointment (UC\_ID5)

* Participating Actors: Patient.
* Entry Conditions: Patient is logged into the website.
* Exit Conditions: Patient has received an acknowledgement from the system that the selected transaction is complete, or if not complete, a message explaining the failure.
* Quality Requirements: (Performance) Patient receives a response from the system in less than 3 seconds.
* Typical flow of events:

Patient selects View my appointment

- System invokes View my appointment use case and retrieve all the appointment of this patient from the database.

1. Use Case: view his/ her doctors (UC\_ID6)

* Participating Actors: Patient.
* Entry Conditions: Patient is logged into the website.
* Exit Conditions: Patient has received an acknowledgement from the system that the selected transaction is complete, or if not complete, a message explaining the failure.
* Quality Requirements: (Performance) Patient receives a response from the system in less than 3 seconds.
* Typical flow of events:
  + - 1. Patient selects View my doctors

- System invokes View my doctors use case and retrieve all doctors whose deals with this patient from the database.

1. Use Case: view his/ her labs’s (UC\_ID7)

* Participating Actors: Patient.
* Entry Conditions: Patient is logged into the website.
* Exit Conditions: Patient has received an acknowledgement from the system that the selected transaction is complete, or if not complete, a message explaining the failure.
* Quality Requirements: (Performance) Patient receives a response from the system in less than 3 seconds.
* Typical flow of events
* 1. Patient selects View my lab’s

- System invokes View my lab’s use case and retrieve all lab’s that belongs to this patient from the database.

1. Use Case: view his/ her surgery details (UC\_ID8)

* Participating Actors: Patient.
* Entry Conditions: Patient is logged into the website.
* Exit Conditions: Patient has received an acknowledgement from the system that the selected transaction is complete, or if not complete, a message explaining the failure.
* Quality Requirements: (Performance) Patient receives a response from the system in less than 3 seconds.
* Typical flow of events:
  + - 1. Patient selects View my surgery details

- System invokes View my surgery details use case and retrieve all surgery details that belongs to this patient from the database which the hospital doctor recorded them.

1. Use Case: (UC\_ID9)

* Participating Actors: Patient.
* Entry Conditions: Patient is logged into the website.
* Exit Conditions: Patient has received an acknowledgement from the system that the selected transaction is complete, or if not complete, a message explaining the failure.
* Quality Requirements: (Performance) Patient receives a response from the system in less than 3 seconds.
* Typical flow of events
* 1. Patient selects View my lab’s
* System invokes View my lab’s use case and retrieve all lab’s that belongs to this patient from the database.

1. Use Case: (UC\_ID10)
2. Use Case: (UC\_ID11)
3. Use Case: write prescription for patient (UC\_ID12)
4. Use Case: (UC\_ID13)
5. Use Case: (UC\_ID14)

## Class Diagram:

## Sequence Diagram:

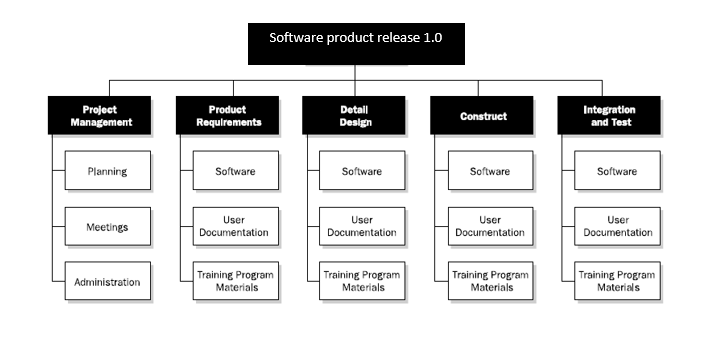
## State Diagram:

# System Evolution:

* The system should be able to work on different operating systems.
* It should work properly on devices with low specifications. Only the server on which it’s installed will be powerful.

# Time Plan

## Work Breakdown Structure



## Gantt chart:

