

**Group 10**

**Project Master Clinic SDS 1.0CM\_Id 04,01,2018**

**Master Clinic** **Statement of Work1.0\_mc****\_04 01, 2018**

**Master Clinic**

**Software Test Plan (STP)**

**Team 10 Master Clinic 1.1**

**Id Date Version: 1.0**

**CM Identifier: Master Clinic1.0\_mc\_04 01, 2018**

**Revision History**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sl. No.** | **Prepared/**  **Modified by** | **E-mail** | **Version** | **Date** | **Approved by** | **Descriptions/**  **Remarks** |
| 1. | Aisha Mousa | aishamousa632@yahoo.com | 1.0 | 3.29,2018 | Hady Maher | TC#1,TC#4,TC#5,TC#10,TC#11,TC#12,TC#13 |
| 2. | Reham Hamdy | rehamhy6@gmail.com | 1.0 | 3.29,2018 | Hady Maher | TC#2,TC#3,TC#6,TC#14,TC#15,TC#16,TC#17,TC#18,TC#20,TC#21,TC#23,TC#24, |
| 3. | Hady Maher | hadymaher311@gmail.com | 1.0 | 4.01,2018 | Aisha Mousa  Reham Hamdy | TC#19,TC#22,TC#25 |
| 4. | Aisha Mousa | aishamousa632@yahoo.com | 1.0 | 4.01,2018 | Hady Maher | TC#7,TC#8,TC#9 |
| 5. | Hady Maher | hadymaher311@gmail.com | 1.0 | 4.05,2018 | Aisha Mousa  Reham Hamdy | TC#26 till TC#41 |
| 6. | Hady Maher | hadymaher311@gmail.com | 1.0 | 4.07,2018 | Aisha Mousa | -Overview  -Testing Environment |
| 7. | Hady Maher | hadymaher311@gmail.com | 1.0 | 4.08,2018 | Aisha Mousa | -Non-functional testing  -Glossary |
| 8. | Aisha Mousa | aishamousa632@yahoo.com | 1.0 | 4.08,2018 | Hady Maher | -Interface testing |

**Distribution list**

|  |  |  |
| --- | --- | --- |
| **Name** | **E-mail** | **Notes** |
| Dr: Ahmed Hamdy |  |  |
| TA: Ali El-Sedeek | alielseddeek@gmail.com |  |
| TA: Dina El Reedy | dinaelreedy@gmail.com |  |

**Table of Contents**

**1. Overview**

**2. Test Methodology**

2.1 Testing Environment

2.2 Functional Testing

2.3 Non-functional Testing

2.3.1 Performance Testing

2.3.2 Quality Testing

2.3.3 Interface Testing

2.3.4 Other Testing

**3. Glossary**

**Software Test Plan**

# **Overview**

In this phase we are going to do the functional testing manually by some test criteria such as entering some valid and invalid data and see what will be the response of the application then evaluate its behavior according to the expected behavior.As for non-functional testing,in some parts we are going to use Google audit tool.

# **Test Methodology**

## **Testing Environment**

To test our system in this phase all we need is to start the application and test every single functionality in its own by submitting multiple kinds of data to this functionality and record the behavior of the application with every kind of entered data.

## **Functional Testing**

This section is included in the Test cases document which attached to this document.

## **Non-functional Testing**

### *Performance Testing*

We test 2 kinds of performance testing:

1. Speed test:

This type of testing is examining the response of the application but this testing is dependent on

the server we are deploying the system on and the ping of the hosting and other things belongs to

real hosting not our local host.

But according to google audit tool on our local host it gives these results in 3 different pages:

1. Patient Login page:

Total performance test gives 81% with first meaningful paint after 3.370 ms and first

interactive after 3.880 ms.

1. Admin profile page:

Total performance test gives 65% with first meaningful paint after 5.020 ms and first

interactive after 5.240 ms.

1. Admin views patients table page:

Total performance test gives 41% with first meaningful paint after 6.350 ms and first

interactive after 8.990 ms.

1. Security test:

This type of testing is examining the security methods of the application such as password

encryption and defending against some attacks like sql injection, cross site scripting, csrf attack…

etc.

1. Passwords encryption:

We are using the latest encryption in PHP 7.2 (bcrypt).

1. Defense against sql injection:

Sql injection is a method of attacking the database of the application if it’s constructed of

one of the sql databases like Mysql we are using.

All sql injection in all forms of the application is failed to inject our database.

1. Defense against cross site scripting:

Cross site scripting is to submit a malicious script in the request to change some behavior

of the application.

By making some cross site scripting attacks on the application the results shows that

some browsers is defending against cross site scripting like Google Chrome and Opera by

killing the request which has any script in it but some other browsers like firefox doesn’t

do that but our website can filter these scripts and deal with them as ordinary strings not

scripting languages.

1. Defense against CSRF attack:

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute

unwanted actions on a web application in which they're currently authenticated.

In the framework we are using for developing this website it has a defense method of this

type of attacks which making a token attached to every request to validate that the request

isn’t automated.

### *Quality Testing*

This type of testing can be examined by showing the response time of the application but this

testing is dependent on the server we are deploying the system on and the ping of the hosting and

other things belongs to real hosting not our local host.

This section is included above in performance speed testing.

### *Interface Testing*

⇒Application Server and Database Server interface:

* Servers are executed properly.
* Errors are handled properly and return an error message for any query made by application.
* Queries sent to the database give expected results.
* When connection between database and application can’t be established an appropriate message is shown to the end user.

### *Other Testing*

1. Testing throttle requests in login system:

This test is examining the too many requests on the login page with invalid data.

If the user try to login with invalid data more than 5 times the system will stop him for 60 seconds

to let him request again to login.

1. Testing data separation with 3 types of users:

Our system has 3 different types of users with different data and different accessibility to the

application because of that we implement a guard system to separate the data and the access to

functionality and pages but there is a kind of admins is called super admin (the owner of the

clinics) he has more policies than other admins and this part will be implemented in the next

phase.

# **Glossary**

Super admin ⇒ The owner of the clinic.

Local host ⇒ local service on our PCs.