



# MedLife

## Test Plan

### **GROUP 8:**

Bipul Kumar (2014CS50282)

Kartar Singh (2014CS50286)

Prachi Singh (2014CS50289)

Gulshan Jangid (2014CS50285)



# Table of Contents

<b>Table of Contents</b>	<b>2</b>
<b>1. Introduction:</b>	<b>3</b>
1.1. Purpose:	3
1.2. Project Overview:	3
<b>2. Test Strategy:</b>	<b>4</b>
2.1. Test Objectives:	4
2.2. Test Assumptions:	4
2.2. Test Principles:	4
<b>3. Execution Strategy:</b>	<b>6</b>
3.1. Entry and Exit Criteria:	6
3.2. Test Cycles:	6
3.3. Test Metrics:	7
3.4. Defect reporting & tracking:	7
<b>4. Testing:</b>	<b>9</b>
4.1. Test Environment:	9
4.2. Test Modules:	9
4.3. Test Cases:	9



# 1. Introduction:

## 1.1. Purpose:

This test plan describes the testing approach and overall framework that will drive the testing of MedLife. This document introduces:

- **Test Strategy:** It will lay rules which form the test and will be based on, including the givens of the project (e.g.: objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy).
- **Execution Strategy:** Describes how the test will be performed and process to identify and report defects, and to fix and implement fixes.
- **Test Management:** Process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, team roster)

## 1.2. Project Overview:

MedLife is intended to be a cloud based and scalable free-to-use application that will be an easy to use framework, that allows the patient to maintain his entire medical history, a common platform between Labs, Patients and Doctors to facilitate the exchange of test reports and a messaging system between Doctors and Patients so that patient can use this portal ask relevant doubts to his doctor. Main aim is to coordinate the interaction between Pathology labs, doctors and patients. There is a host of features planned to streamline and improve user experience, including messaging, real-time notifications, and segregated profiles.



## 2. Test Strategy:

### 2.1. Test Objectives:

The objective of the test is to verify that the functionality of Medlife works according to the specifications. The test will execute and verify the test scripts, identify, fix and retest all high and medium severity defects per the entrance criteria, prioritize lower severity defects for future fixing via CR.

The final product of the test is twofold:

- A usable website
- A set of stable test scripts that can be reused for test execution

### 2.2. Test Assumptions:

- Exploratory Testing would be carried out once the build is ready for testing.
- Test environment and preparation activities will be owned by Development Team.
- Development team will provide Defect fix plans based on the Defect meetings during each cycle to plan.
- The same will be informed to Test team prior to start of Defect fix cycles.
- Project team has the knowledge and experience necessary, or has received adequate training in the system, the project and the testing processes.
- The system will be treated as a black box; if the information shows correctly online and in the reports, it will be assumed that the database is working properly.

### 2.2. Test Principles:

- Testing will be focused on meeting the objectives, cost efficiency, and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.

- Testing environment and data will emulate a production environment as much as possible.
- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals. There will be entrance and exit criteria.



## 3. Execution Strategy:

### 3.1. Entry and Exit Criteria:

#### 3.1.1 Unit Testing

**Entry:**

- 1) Requirement, Design or other relevant documents are reviewed, analysed and approved
- 2) Developed code for new units is available
- 3) Test environment is available
- 4) New requirements are defined and approved

**Exit:**

- 1) Testing phase finished successfully.
- 2) Detected bugs have been fixed and closed.

#### 3.1.2 Integration Testing

**Entry:**

- 1) Unit Testing phase is finished.
- 2) Bugs found in unit testing phase have been fixed.
- 3) Integration testing plan and the testing environment is ready.
- 4) Modules to be integrated have successfully completed the unit testing phase.

**Exit:**

- 1) Testing phase finished successfully.
- 2) Detected bugs have been fixed and closed.

#### 3.1.3 System Testing

**Entry:**

- 1) Test cases and environment for system testing is available.
- 2) Integration testing phase finished successfully.
- 3) Bugs found in integration testing phase have been fixed.

**Exit:**

- 1) Testing phase finished successfully.
- 2) Detected bugs have been fixed and closed.
- 3) All requirements have been met.

### 3.2. Test Cycles:

Follow three test cycles:

- 1) Test Cycle 1 : In this cycle, try to detect most of the defects by executing all the test cases.
- 2) Test Cycle 2 : The fixes for the defects found in the first test cycle are verified in the second test cycle.
- 3) Test Cycle 3 : In the third and the final test cycle, a selected subset of test cases of the original test suite is executed to ensure that the software is

stable before it is released to the customer. The objective of this test cycle is to execute all the selected test cases against a single software image.

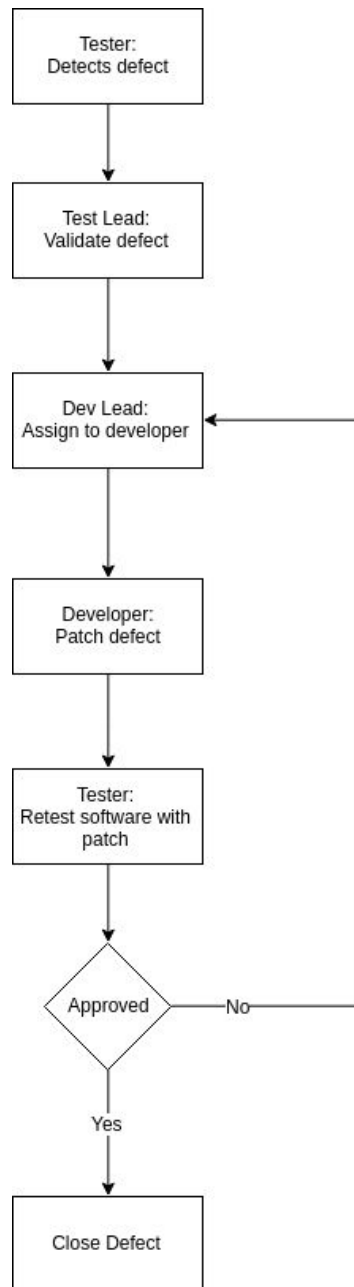
### 3.3. Test Metrics:

Term	Definition
Open Defects	Number of defects/bugs identified in testing phase and are yet to be fixed.
Fixed Defects	Number of defects/bugs that were open in past and have now been fixed.
Test Execution Status	Provide pass, fail and total defects in the given testing phase.

### 3.4. Defect reporting & tracking:

What we want to achieve here is to be able to handle reported bugs/defects and then assign them to developers and track the bug/defect.

Flow of Defect reporting and tracking:







## 4. Testing:

### 4.1. Test Environment:

- OS: Windows 10
- RAM: 8GB
- PHP 7.3.4
- Browsers: Chrome
- Network: speed of at least 1 Mbps

### 4.2. Test Modules:

- Sign Up
- Login
- Patient Access List
- Patient Timeline
- Upload Report

### 4.3. Test Cases:

#### 4.3.1. Sign Up

TEST ID	TO BE TESTED	EXECUTION STEPS	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
SIGNUP_01	Validate username	Check in the database if the username exists for that role	Username	The username already exists. Please try another username.	The username already exists. Please try another username.	Success
		Check username format	Username	Username must start with a small letter	Username must start with a small letter	Success
SIGNUP_02	Validate password	Check length of the password	Password	Password must be atleast 6 characters long.	Password must be atleast 6 characters long.	Success
SIGNUP_03	First Name mandatory field	Filled	NA	Able to continue further		Success

		Not filled	NA	Please fill out this field.	Please fill out this field.	Success
--	--	------------	----	-----------------------------	-----------------------------	---------

#### 4.3.2. Login

TEST ID	TO BE TESTED	EXECUTION STEPS	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
LOGIN_01	Username, password, role mandatory field	Username is blank and password is entered	Username: NULL Password: value	Please fill out this field.	Please fill out this field.	Success
		Username is entered and password is blank	Username: value Password: NULL	Please fill out this field.	Please fill out this field.	Success
		Username is blank and password is blank	Username: NULL Password: NULL	Please fill out this field.	Please fill out this field.	Success
LOGIN_02	Validate user	Check in the database if the username-password-role combination matches	Username, Password, Role	Control should be passed to the homepage of corresponding user		Success
		Username password & role does not match	Username, Password, Role	Username/Password/Role do not match.	Username/Password/Role do not match.	Success

#### 4.3.3. Patient Access List

TEST ID	TO BE TESTED	EXECUTION STEPS	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
ACCESS_01	Username, role mandatory field	Username is blank and role is blank	Username: NULL Role: NULL	Username and Role field cannot be empty	Username and Role field cannot be empty	Success
		Username is entered and role is blank	Username: value Role: NULL	Role field cannot be empty	Role field cannot be empty	Success
		Username is blank and role is entered	Username: NULL Role: value	Username field cannot be empty	Username field cannot be empty	Success
ACCESS_02	Add user	Check in the database if the	Username, Role: Lab	Add given user to Access list of the	Add given user to Access list of	Success

		username-role combination matches		patient & add patient to Access list of the lab.	the patient & add patient to Access list of the lab.	
		Check in the database if the username-role combination matches	Username, Role: Doctor	Add given user to Access list of patient & add patient to Access list of the doctor.	Add given user to Access list of patient & add patient to Access list of the doctor.	Success
		Username & role does not match	Username, Role	The user does not exist.	The user does not exist.	Success
ACCESS_03	Remove user	Select user to remove from Access List	Click on Remove	User removed	User removed	Success

#### 4.3.4. Patient Timeline

TEST ID	TO BE TESTED	EXECUTION STEPS	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TIMELINE_01	Edit Tags	Patient views his Report Timeline	Click on Edit Tags	Tags replaced in the database	Tags replaced in the database	Success
		Doctor views patient's Report Timeline	Click on Edit Tags for posts editable by the doctor	Tags replaced in the database	Tags replaced in the database	Success
TIMELINE_02	Add comment	Doctor views patient's Report Timeline	Click on Edit Tags for posts editable by the doctor	Comment replaced in the database	Comment replaced in the database	Success
TIMELINE_03	Search Tag	User searches for posts with a given tag	Input tag	Posts only with given tags	Posts only with given tags	Success

#### 4.3.5. Upload Report

TEST ID	TO BE TESTED	EXECUTION STEPS	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
UPLOAD_01	Patient and Doctor username mandatory field	Patient username is blank and Doctor username is blank	Patient username: NULL Doctor username: NULL	Patient and doctor username cannot be empty	Patient and doctor username cannot be empty	Success

		Patient username is entered and Doctor username is blank	Patient username: value Doctor username: NULL	Doctor username cannot be empty	Doctor username cannot be empty	Success
		Patient username is blank and Doctor username is entered	Patient username: NULL Doctor username: value	Patient username cannot be empty	Patient username cannot be empty	Success
UPLOAD_02	Upload Report	Check in the database if lab exists in patient's access list	Patient Username	Report gets uploaded in patient's blob	Report gets uploaded in patient's blob	Success
		Lab does not exist in patient's access list	Patient Username	Patient has not given you access	Patient has not given you access	Success