

---

**Software Test Plan  
EKG USING CLOUD SERVICES**

**Version 1.0**

**10/22/2013**

**Prepared by Team 3**

**Jabili Vakkalanka**

**Sharifa Minkabo**

**Zheng Rong**

**Jai Krishna Pyneni**

**Sharath Reddy**

## Table of Contents

<b>1</b>	<b>Introduction</b> .....	Error! Bookmark not defined.
<b>2</b>	<b>Test Objective</b> .....	<b>3</b>
<b>3</b>	<b>Types of Testing</b> .....	Error! Bookmark not defined.
<b>4</b>	<b>Unit Testing</b> .....	<b>3</b>
	4.1 Test case 1.....	4
	4.2 Test case 2.....	5
	4.3 Test case 3.....	6
	4.4 Test case	
	4.....	<b>Error!</b>
	<b>Bookmark not defined.</b>	
	4.5 Test case 5.....	8
	4.6 Test case 6.....	9
	4.7 Test case 7.....	10
	4.8 Test case 8.....	11
<b>5</b>	<b>Integration Testing</b> .....	<b>11</b>
	5.1 Test case 9.....	12
	5.2 Test case 10.....	13
<b>6</b>	<b>Syatem Testing</b> .....	<b>13</b>
	6.1 Test case 11.....	14
<b>7</b>	<b>Performance Testing</b> .....	<b>15</b>
	7.1 Test case 12.....	15
<b>8</b>	<b>Acceptance Testing</b> .....	<b>16</b>
	8.1 Test case 13.....	16
<b>9</b>	<b>Bug Tracking</b> .....	<b>16</b>
<b>10</b>	<b>Issues</b> .....	<b>17</b>

## Revision History

Name	Date	Reason For Changes	Version
SRS	10/20/2013	Changes according to requirements	V1.0.1

## **1 Introduction**

This document is a demonstrating our testing strategy for the EKG USING CLOUD SERVICES. This document introduces the test cases, entry-exit criteria and standards that will apply to the unit, integration and system testing. We will utilize testing criteria under the white box, black box, and system-testing paradigm.

## **2 Test Objective**

The objective of our test plan is to find and report as many bugs as possible to improve the quality of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

## **3 Types of Testing**

The tests we will perform are going to fall under the following categories:

1. Unit Test.
2. Integration Test.
3. System Test.
4. Performance Test.
5. Acceptance Test.

All test categories fall under broader testing class of either:

1. White box testing, or
2. Black box testing.

We will consider both:

1. Sunny day scenarios, and
2. Rainy day scenarios.

Each test case will have:

1. Entry criteria.
2. Exit criteria.

## **4 Unit Testing**

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness.

The following test cases we will perform under unit testing.

#### 4.1 Test Case 1

<b>Test Type</b>	Unit Test, Sunny Day, Black Box	
<b>Test Case Number</b>	1	
<b>Test Case Name</b>	Admin login	
<b>Test Case Description</b>	Administrator input the username and password to login to the cloud system.	
<b>Item(s) to be tested</b>		
1	Login process	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
Username password		Login is successful.
<b>Procedural Steps</b>		
1	Fill in the username and password	
2	Click Login button	
3	Validate the input	
4	Get into system interface	

**Entry criteria:**

1. Enter the IP address into Web Explorer.

**Exit criteria:**

1. Login successful.

## 4.2 Test Case 2

<b>Test Type</b>	Unit Test, Rainy Day, Black Box	
<b>Test Case Number</b>	2	
<b>Test Case Name</b>	Exceptions in login.	
<b>Test Case Description</b>	Different exceptions in login process will be considered.	
<b>Item(s) to be tested</b>		
1	Invalid information.	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
1. Missing information (e.g., username is missing).		1. Show message "Username or Password can't be empty".
2. Invalid information (e.g., wrong UID or PWD)		2. Show message "Invalid Username or Password".
<b>Procedural Steps</b>		
1	No input username or password into the text field.	
2	Wrong username or password	
3	Check the error message.	

**Entry criteria:**

1. Sunny day scenarios are successful for admin to login.

**Exit criteria:**

1. Exceptions are successfully handled.

### 4.3 Test Case 3

<b>Test Type</b>		Unit Test, Sunny Day, Black Box
<b>Test Case Number</b>		3
<b>Test Case Name</b>		Admin Add New Users(Doctors or Patients)
<b>Test Case Description</b>		Test Add function of system
<b>Item(s) to be tested</b>		
1	Add new user	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
<ol style="list-style-type: none"> <li>1. Get into user control panel</li> <li>2. Select Add option</li> <li>3. Choose type of user</li> <li>4. Input the info of user</li> <li>5. Confirm</li> </ol>		<ol style="list-style-type: none"> <li>1. Show the user control panel interface</li> <li>2. Show the table for admin to fill if he wants to add a new user.</li> <li>3. Add a new user is finished and show successful message to the interface.</li> </ol>
<b>Procedural Steps</b>		
1	Click on the Setting button and User Control button	
2	Click Add New User button	
3	Click Patient or Doctor for user's type	
4	Input the information of user like Username, Password, Name ,etc.	
5	Click Confirm button	

**Entry criteria:**

1. Admin has accessed the system.
2. Cookies is alive.

**Exit criteria:**

1. Validation works correctly.
2. Click other buttons like Cancel, Exit, Modify, etc.
3. Close the browser.
4. Time out.

#### 4.4 Test Case 4

<b>Test Type</b>		Unit Test , Rainy Day, Black Box
<b>Test Case Number</b>		4
<b>Test Case Name</b>		Exceptions in Add New Users
<b>Test Case Description</b>		Test the Add New User process using exceptional cases.
<b>Item(s) to be tested</b>		
1	Add new user.	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
1. Missing required information (e.g., username or password is missing).		1. Show message “Incomplete Information”.
2. Invalid information (e.g., wrong DOB format)		2. Show message “Invalid information”.
3. Username existed		3. Show message “User already Exist”.
4. Do nothing for 5 minutes		4. Show message “Time out, login again.” And turn to the login page.
<b>Procedural Steps</b>		
1	Click on the Add New Users.	
2	Intentionally miss some field.	
3	Intentionally give invalid information.	
4	Intentionally give same required fields information.	
5	Intentionally wait for long time no input.	
6	Check the error message.	

**Entry criteria:**

1. Sunny day scenarios are successful for getting into control panel.

**Exit criteria:**

1. Exceptions are handled successfully.

#### 4.5 Test Case 5

<b>Test Type</b>		Unit Test , Sunny Day, White Box
<b>Test Case Number</b>		5
<b>Test Case Name</b>		Receive data from patients' device
<b>Test Case Description</b>		Test whether the data could be accepted by cloud system.
<b>Item(s) to be tested</b>		
1	Data acceptance	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
Patients' devices send a data by ftp server for every minute.		All data are accepted by cloud system, and correctly save in the file system.
<b>Procedural Steps</b>		
1	Use an EKG simulator to emulate the patient device, and send data for every minute.	
2	Check whether UID and Password is accepted.	
3	Check data are correct or not.	

**Entry criteria:**

1. Send data from an EKG simulator.

**Exit criteria:**

2. Data acceptations are handled successfully.

#### 4.6 Test Case 6

<b>Test Type</b>	Unit Test , Rainy Day, White Box	
<b>Test Case Number</b>	6	
<b>Test Case Name</b>	Exceptions in Data acceptations	
<b>Test Case Description</b>	Test whether wrong device could send data to cloud system.	
<b>Item(s) to be tested</b>		
1	Exceptions in Data acceptations	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
Patients' devices send a data by ftp server for every minute but with wrong UID and Password.		Invalid UID and Password.
<b>Procedural Steps</b>		
1	Use an EKG simulator to emulate the patient device, and send data for every minute but with wrong UID and Password.	
2	Check whether UID and Password is accepted.	
3	Check whether the data are saved in file system.	

**Entry criteria:**

1. Sunny day scenario for data acceptations are working perfect.

**Exit criteria:**

1. Exceptions are handled successfully.

#### 4.7 Test Case 7

<b>Test Type</b>	Unit Test , Sunny Day, White Box	
<b>Test Case Number</b>	7	
<b>Test Case Name</b>	Visualization of data	
<b>Test Case Description</b>	Test whether the data visualization is working correctly and showing perfect result or not.	
<b>Item(s) to be tested</b>		
1	Data visualization.	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
Patient's ID and other information (e.g., date range etc.)		EKG data summary and plots.
<b>Procedural Steps</b>		
1	Login as a doctor.	
2	Give patient's ID and date range.	
3	Click Submit button.	
4	Check visualized information (summary report or plots) are correct or not.	

**Entry criteria:**

1. Patient's EKG data exist in the database.

**Exit criteria:**

1. Report and plots are appearing.
2. Data in report and plots are correct.

#### 4.8 Test Case 8

<b>Test Type</b>	Unit Test, Rainy Day, White Box	
<b>Test Case Number</b>	8	
<b>Test Case Name</b>	Exceptions in visualization of data	
<b>Test Case Description</b>	Test whether the data visualization is responding the exceptional case of invalid patient ID, date range and so on.	
<b>Item(s) to be tested</b>		
1	Data visualization.	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
1. Patient's ID is invalid. 2. Date range is invalid.		"Invalid Entry" message.
<b>Procedural Steps</b>		
1	Login as a doctor.	
2	Give a random patient ID and date range.	
3	Click Submit button.	
4	Check whether the error message is appearing or not.	

**Entry criteria:**

2. Sunny day scenario for data visualization is working perfect.

**Exit criteria:**

2. Exceptions are handled successfully.

#### 4.9 Test Case 9

<b>Test Type</b>		Unit Test, Sunny Day, Black Box
<b>Test Case Number</b>		9
<b>Test Case Name</b>		Write notes for patient's data
<b>Test Case Description</b>		Test whether the doctor could write note for patient's data.
<b>Item(s) to be tested</b>		
1	Note writing	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
<ol style="list-style-type: none"> <li>1. Create a new TXT file</li> <li>2. Write note in this file.</li> </ol>		<ol style="list-style-type: none"> <li>1. Create successful</li> <li>2. Save note successful</li> </ol>
<b>Procedural Steps</b>		
1	Login as a doctor.	
2	Click Patient's EKG File button	
3	Enter in Patient's folder	
4	Click Create Notes	
5	Input file name, contend and save it.	

**Entry criteria:**

1. Login as a doctor.

**Exit criteria:**

1. Writing note is handled successfully.

### 5 Integration Testing

There are three modules that will need to be integrated: the User Interface module and the EKG Data Receiving and Database module. The following describes these. We will be employing an incremental testing strategy to complete the integration.

**Module 1 - User Interface (GUI) Module**

This module provides a simple GUI where the actors can perform the different actions. The testing will be performed to ensure two-way data transfer accuracy between user interface and data repository.

## Module 2 – EKG Data Receiving

This module transfers the EKG sensor data to repository. Testing will be performed to ensure one-way data transfer accuracy from EKG simulator to data repository.

## Module 3 –Database module

This module covers the whole EKG Cloud Sever system working well. To perform data integrity, we will conduct incremental testing.

### 5.1 Test Case 10

<b>Test Type</b>	Integration Test, White Box	
<b>Test Case Number</b>	10	
<b>Test Case Name</b>	Module 1 and Module 3 integration test	
<b>Test Case Description</b>	Test whether data between two modules are being transferred or not.	
<b>Item(s) to be tested</b>		
1	MySQL database.	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
Enter, change/update data from module 1.		See the expected change of data in the MySQL tables.
<b>Procedural Steps</b>		
1	Login as a admin.	
2	Click on Modify User link.	
3	Update information.	
4	Submit.	
5	Click on Delete User link	
6	Confirm	
7	Submit.	
8	Go to MySQL tables and check the corresponding information.	

**Entry criteria:**

1. MySQL database is successfully connected with Module 1.

**Exit criteria:**

1. Data stored in the database are correct.

**5.2 Test Case 11**

<b>Test Type</b>		Integration Test, White Box.
<b>Test Case Number</b>		10
<b>Test Case Name</b>		EKG simulator transforms data to Cloud Server. (Module 2 and Module 3)
<b>Test Case Description</b>		Test the EKG simulator whether it is successfully transferring the data to the system or not.
<b>Item(s) to be tested</b>		
1	Transformation.	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
1. EKG sensor data		1. Data is successfully uploaded to system.
<b>Procedural Steps</b>		
1	Open the EKG simulator.	
2	Click on the Send Data/Upload Data button.	
3	Check the data in the repository.	

**Entry criteria:**

1. Connection between database and EKG simulator is established.

**Exit criteria:**

1. Data uploaded in the database is correct.

## 6 System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. In our case we will focus only on function validation in the Amazon EC2 cloud environment.

### 6.1 Test Case 11

<b>Test Type</b>		System Test, Black Box
<b>Test Case Number</b>		11
<b>Test Case Name</b>		System Test
<b>Test Case Description</b>		Test runs the project inside Amazon EC2 cloud services.
<b>Item(s) to be tested</b>		
1	Entire project inside EC2.	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
Perform all actions in the project.		All outputs are stable as it is in offline.
<b>Procedural Steps</b>		
1	Create a micro instance in Amazon EC2.	
2	Start Apache web server.	
3	Upload the project (the DB, user interface etc.)	
4	Run the project.	
5	Perform all operation in incremental fashion.	
6	Check the output.	

**Entry criteria:**

1. Amazon EC2 instance has been created successfully.
2. The project passed all the offline test cases.
3. Uploading the project is successful.

**Exit criteria:**

1. The outputs for all actions are identical as when tested offline.

## 7 Performance Testing

### 7.1 Test Case 12

<b>Test Type</b>	Performance Test, Rainy Day, Black Box.	
<b>Test Case Number</b>	12	
<b>Test Case Name</b>	Load, validation, response time test.	
<b>Test Case Description</b>	Test whether user's queries/problem reports are successfully uploaded or not.	
<b>Item(s) to be tested</b>		
1	Load test.	
<b>Specifications</b>		
<b>Input</b>		<b>Expected Output/Result</b>
Invalid entry in updated information.		Show message "Invalid Entry" message.
<b>Procedural Steps</b>		
1	Create hundreds of user's accounts.	
2	Make service request for hundreds of users at the same time.	
3	See the response time and validity of the system.	

**Entry criteria:**

1. System passed all units, system, integration tests.

**Exit criteria:**

1. System responds successfully.

## 8 Acceptance Testing

### 8.1 Test Case 13

<b>Test Type</b>	Acceptance Test
<b>Test Case Number</b>	13
<b>Test Case Name</b>	User acceptance test
<b>Test Case Description</b>	Test whether users' demands and expectations are fulfilled or not.
<b>Item(s) to be tested</b>	
1	All features.
<b>Specifications</b>	
<b>Input</b>	<b>Expected Output/Result</b>
Perform all actions.	User is satisfied with the outcomes.
<b>Procedural Steps</b>	
1	Meet the users.
2	Ask them to perform all the actions they want to do.
3	Show them the outputs.

**Entry criteria:**

1. System passed all unit, system, integration tests.

**Exit criteria:**

1. System responds successfully and customers or users are satisfied with the outcomes.

## **9 Bug Tracking**

During testing, the testing team members encounter behavior that goes against a specified or implied design requirement in the product, that we call bugs. All the bugs will be reported and documented. When this happens, we will reproduce the bugs for the developers. The testing team will keep track of the bug states as well.

## **10 Issues**

1. All tests will be performed manually. We are not using any testing tools.
2. Performance test will be limited due to resource limitations for “free” EC2 users.