
Testing & Integration Plan

FH Mobile Application

Version 1.0

Prepared by

Omar Rivera
Andrew Poirier
Daven Amin
Rick Rejeleene

Table of Contents

1.0 Introduction	3
2.0 Modular.....	3
2.1 Validation Requirements -OX.....	3
2.1.1 Validation Requirements	3
2.2 User Interface Requirements -OX	4
2.2.1- User Interface – GUI provided	4
2.2.2 User Interface – Allow authorizing individuals to apply penalty or credit	5
2.3 Performance Requirements.....	6
2.3.1 – Performance Requirements – Booting up/ Authenticate login	6
2.3.2- Performance Requirements – Retrieving Data and Application Reporting Back	7
2.4 User Platform Requirements	7
2.4.1 Test User Platform Requirements.....	7
2.5 User Account Requirements	8
2.5.1 Test User Account Requirements - full name	8
2.5.2 Test User Account Requirements - Login name.....	9
2.5.3 Test User Account Requirements - Password	10
2.5.4 Test User Account Requirements - Account Balance Information	10
2.6 Account Requirements.....	11
2.6.1 Test account login	11
2.6.2 Test account registration	12
2.6.3 Test account persistence	12
2.6.4 Test account updates	13
2.7 Workout schedule requirements	13
2.7.1 Test account registration	13
2.7.2 Test user authentication preferences.....	14
2.8 Database Requirements.....	15
2.8.1– User information fields test scenario.....	15
2.8.2-Trainer can input information test scenario.....	16
2.8.3-Trainer shall have access to penalty/credit & activity goals.....	16
2.8.4-Users can select a trainer	17
2.8.5-Having a penalty & credit option.....	17

2.8.6-Audit Trail	18
2.8.7-System will have limited access to authorized users.....	19
2.8.8-Trainer only has access to user's schedule	20
2.8.9-Trainer can update penalty amount	20
2.8.10-Trainer can control the actual amount of activity to calculate penalty	21
3.0 System Integration	22
3.1.1– Integration Phase -1 – Testing system components independently	22
3.2.1– Integration Phase -2 – Data Base and Server Communications, Functionality and Performance	23
3.3.1– Integration Phase -3 –Client and Server Communications, Functionality and Performance ..	24
3.4.1– Integration Phase -4 –Complete system Integration Communications, Functionality and Performance	25
4.0 Testing Schedule	26

1.0 Introduction

The main purpose of the system integration and testing is to validate the software is ready to be release to the public. The integration procedure includes a test case for all the testable requirements and a step by step guile on how to test each case. The modular approach allow us to confirm all the requirements are being implemented and the products have the expected quality.

The integration testing routines follow the “White Box” approach to test and integrate the software. This looking inside the box approach, provide us with more flexibility in terms of testing each one of the modules functionality, software/hardware interfaces, and database structures. The integration testing follows a modular order and tries to test each one of the system components in different phases.

2.0 Modular

2.1 Validation Requirements

2.1.1 Validation Requirements

Requirement Number	3.2.4
Requirement Name	Validation Requirements
Requirement Description	<ul style="list-style-type: none"> • Allow a user with <i>authenticator</i> privileges to view <i>workout schedules</i> which they have been authorized to <i>authenticate</i> that <i>workout metrics</i> have been met. • Allow a user with <i>authenticator</i> privileges to mark <i>workout metrics</i> in authorized <i>workout schedules</i> as <i>accomplished</i>.
Pre-conditions	The user must have authenticator privileges to mark metrics for another users
Procedures	<ol style="list-style-type: none"> 1. Using the provided GUI navigate to the authenticate workout routine screen using the available menu/tabs (see Figure 2.4.1-1Gui Menu/Tabs). 2. Select the user that you will validate (If multiple users are available), and complete the metric information of the workout routine. This will consist of entering amount of repetitions, activities, etc. 3. After entering all the required fields press the summit button at the end of the screen (See figure 2.4.1.-2Summit button). 4. Wait for the server to confirm the workout validation.
Post Conditions	The system will return the message: “Schedule workout routine validated” informing to the user the validation is completed.
Test Results	PASS/FAIL

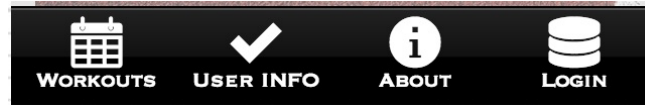


Figure 2.1.1-1 GUI Menu and Tabs

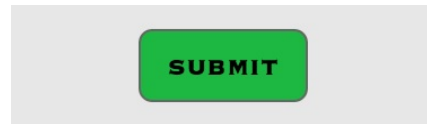


Figure 2.1.1-2-Submit Button

2.2 User Interface Requirements

2.2.1- User Interface – GUI provided

Requirement Number	4.1-UI-1
Requirement Name	User Interface – GUI provided
Requirement Description	All users shall be able to access the mobile application using a GUI provided by the application.
Pre-conditions	The user must have the application installed in the phone.
Procedures	<ol style="list-style-type: none"> 1. Using your Android phone, navigate through the multiple Android desktop to find the FH app Icon. 2. Once you find the FH Icon, selected and wait until the software loads. 3. If the application was installed correctly, the screen shall display a GUI interface for the user to interact with the app (See Figure 2.8.UI1 GUI).
Post Conditions	The GUI that can be controlled by the user will appear in the screen.
Test Results	PASS

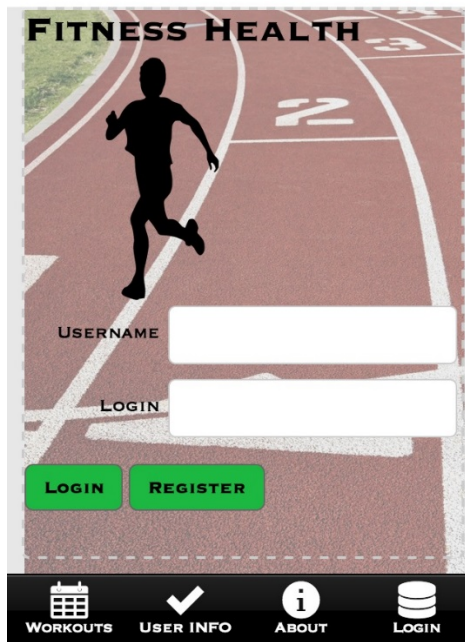


Figure 2.2.1 UI1 GUI

2.2.2 User Interface – Allow authorizing individuals to apply penalty or credit

Requirement Number	4.1-UI-3
Requirement Name	User Interface
Requirement Description	Allow authorizing individuals to apply penalty or credit
Pre-conditions	The user must be logged into the system.
Procedures	<ol style="list-style-type: none"> 1. Using the GUI tabs, Navigate to the user information screen. (See Figure 4.1-UI-3) 2. In the Trainer Field add the name of the person that you would like to have as a trainer. 3. Wait until the system confirms the trainer it's associated with your account.
Post Conditions	The system will display a message that confirms the trainer it's now associated with your account.
Test Results	PASS

USER INFORMATION

FIRST NAME

LAST NAME

USER TYPE

USER ID

PASSWORD

TRAINER

DOB

EMAIL

WORKOUTS USER INFO ABOUT LOGIN

Figure 4.1-UI-3

2.3 Performance Requirements

2.3.1 - Performance Requirements – Booting up/ Authenticate login

Requirement Number	5.1-1
Requirement Name	Performance Requirements – Booting up/ Authenticate login
Requirement Description	Users shall be able to log in to the application and received confirmation in less than 5 seconds (< 5 sec).
Pre-conditions	User must have an account in the system and the application installed. Have a time measurements device available. (chronometer)
Procedures	<ol style="list-style-type: none"> 1. Open the application in your Android phone. 2. The first screen is the login screen, fill the user and password fields with your account information (NOTE: DO NOT PRESS LOGIN BUTTON YET) 3. Using your “Chronometer” select the mode that provides the best time resolution to measure seconds. 4. Press the chronometer start button and the application login button simultaneously to measure the authentication time. 5. Wait until the system logs you in and stop the chronometer <p>P.S: The system should log you in < 5 secs.</p>
Post Conditions	The system will log you in and change to a screen different screen
Test Results	PASS

2.3.2- Performance Requirements – Retrieving Data and Application Reporting Back

Requirement Number	5.1-2 , 5.1-3
Requirement Name	Performance Requirements – Retrieving Data and Application Reporting Back
Requirement Description	<ul style="list-style-type: none"> Trainer shall be able to upload workout results and received a confirmation in less than 5 seconds (< 5 sec). Application shall confirm data being updated in the database in less than 5 seconds (< 5 sec).
Pre-conditions	User must have an account in the system and the application installed. Have a time measurements device available. (chronometer)
Procedures	<ol style="list-style-type: none"> Open the application in your Android phone. In the user information screen fill the trainer field with valid information (NOTE: DO NOT PRESS SUMMIT BUTTON YET) Using your “Chronometer” select the mode that provides the best time resolution to measure seconds. Press the chronometer start button and the application submit button simultaneously to measure the system response time. Wait until the system displays the message that confirms your information have been updated and stop the chronometer <p>P.S: The system should report back in < 5 secs.</p>
Post Conditions	The system will log you in and change to a screen different screen
Test Results	PASS

2.4 User Platform Requirements

2.4.1 Test User Platform Requirements

Requirement Number	3.2.1
Requirement Name	User Platform Requirements
Requirement Description	The User platform is to be tested by using a device with an Android Operating System.
Pre-conditions	<p>The User needs a Device with an Android Operating System with Version 4.0 or higher.</p> <p>The User also needs to have Intel XDK software.</p>
Procedures	<ol style="list-style-type: none"> Find an android device. Verify that it is running Android version 4.0 or higher. Connect the Device to the System. Open Intel XDK Install the application by clicking Build using Intel XDK. Open the Application on your device named Intel XDK.
Post Conditions	<ul style="list-style-type: none"> If the Application is successfully installed, it will open. If the Application is not successfully installed, it will prompt the user “Application installation failed”
Test Results	Pass if all the Post Conditions are met.

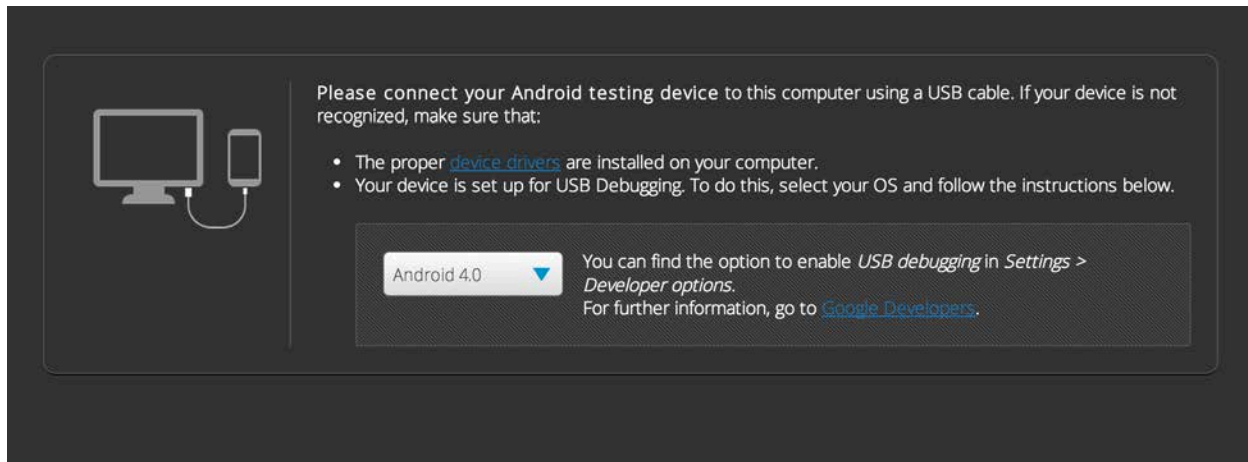


Figure for 2.4.1

2.5 User Account Requirements

2.5.1 Test User Account Requirements - full name

Requirement Number	3.2.7
Requirement Name	User Account Requirements – full name
Requirement Description	The User account contains a form that needs to have valid information for testing, here the first name and the last name is tested.
Pre conditions	The User needs to be in the User Information Page.
Procedures	<ol style="list-style-type: none"> 1. Enter the First Name. 2. Enter the Last Name. 3. Enter the valid information for the rest of the form. 4. Click Submit after entering the above information.
Post Conditions	<ul style="list-style-type: none"> • If the First name and Last name is empty, the user information page will not submit. • If the First name and Last name are alphabets, the page will successfully pass through. • If the First name and Last name contain numbers, the page will not submit.
Test Results	<ul style="list-style-type: none"> • Pass if all the Post Conditions are True.

The screenshot shows a web application interface with a blue header bar containing the text "FITNESS HEALTH". Below the header, the section is titled "USER INFORMATION". There are two input fields: the first is labeled "FIRST NAME" and contains the placeholder text "FIRST NAME"; the second is labeled "LAST NAME" and contains the placeholder text "LAST_NAME".

Figure 2.5.1

2.5.2 Test User Account Requirements - Login name

Requirement Number	3.2.7
Requirement Name	User Account Requirements – Login name
Requirement Description	The User account form needs to be verified for testing, the login name is tested here.
Pre conditions	The User needs to be in the User Information Page.
Procedures	<ol style="list-style-type: none"> 1. Enter the Login ID 2. Enter the valid information for the rest of the form. 3. Click Submit after entering all the above Information
Post Conditions	<ul style="list-style-type: none"> • If the Login ID is empty, the user information page will not submit • If the Login ID is greater than 10 characters, the user information page will not submit.
Test Results	<ul style="list-style-type: none"> • Pass if all the Post conditions are True

The screenshot shows a login form with two input fields. The first field is labeled "LOGIN ID" and is empty. The second field is labeled "PASSWORD" and contains the placeholder text "PASSWORD".

Figure for 2.5.2, 2.5.3

2.5.3 Test User Account Requirements - Password

Requirement Number	3.2.7
Requirement Name	User Account Requirements – Password
Requirement Description	The User account form contains a password and the password field is tested here.
Pre conditions	The User needs to be in the User Information Page.
Procedures	<ol style="list-style-type: none"> 1. Enter the Password 2. Enter the valid information for the rest of the form. 3. Click Submit after entering all the above information.
Post Conditions	<ul style="list-style-type: none"> • The Page will not submit if the Password is empty • The Page will submit if the Password is not empty.
Test Results	<ul style="list-style-type: none"> • Pass if all the Post conditions are True

2.5.4 Test User Account Requirements - Account Balance Information

Requirement Number	3.2.7
Requirement Name	User Account Requirements – Balance Information
Requirement Description	The User account contains balance information for new user, the balance amount is set by default.
Pre conditions	The User needs to be in the User Information Page.
Procedures	<ol style="list-style-type: none"> 1. The User needs to fill valid information for the rest of the form. 2. Click Submit
Post Conditions	<ul style="list-style-type: none"> • The Default User balance is assigned for the user.
Test Results	<ul style="list-style-type: none"> • Pass if all the Post conditions are True

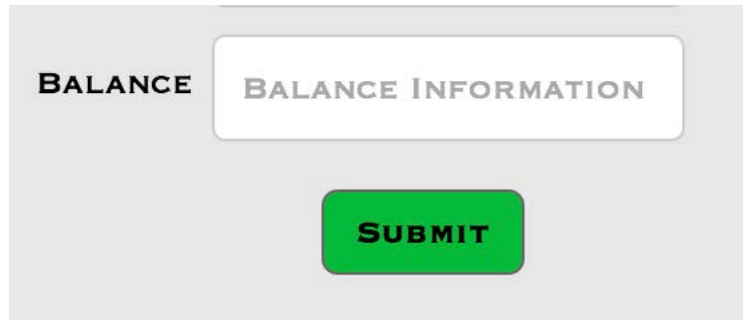
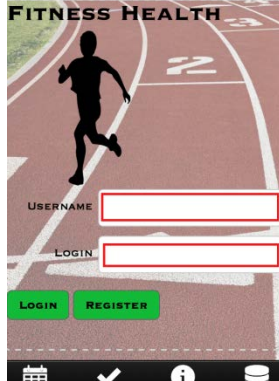


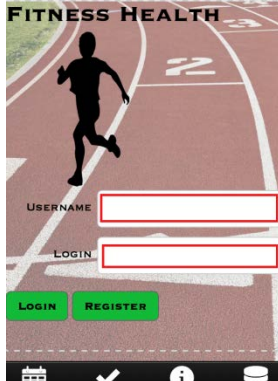
Figure 2.5.4

2.6 Account Requirements

2.6.1 Test account login

Requirement Number	3.2.2	
Requirement Name	Account Requirements	
Requirement Description	Allow the user to log into the service with account credentials	
Pre-conditions	User has a valid account (username and password) on the service	
Procedures		<ol style="list-style-type: none"> 1. Open the FH mobile application 2. Type in valid username into "username" field 3. Type in valid password into "password" field 4. Press "Login" button
Post Conditions	Mobile application proceeds to Workouts page	
Test Results	PASS if post conditions are true, FAIL if application hangs or returns an error/alert.	

2.6.2 Test account registration

Requirement Number	3.2.2
Requirement Name	Account Requirements
Requirements Description	Allow the user create a new account in the service
Pre-conditions	NONE
Procedures	 <ol style="list-style-type: none"> 1. Open the FH mobile application 2. Type in a desired username into "username" field 3. Type in a desired password into "password" field 4. Press "Register" button
Post Conditions	<p>If password does not meet the requirements (is not between 1 and 20 alphanumeric characters), an alert is displayed requesting the password be modified to meet the password requirements.</p> <p>If username does not meet the requirements (is not between 1 and 10 alphanumeric characters), an alert is displayed requesting the username be modified to meet the username requirements.</p> <p>If the username already exists in the system, an alert is displayed requesting another valid username be chosen.</p> <p>If username and passwords meet the requirements and username does not already exist in the system, an alert is displayed stating "success" and application proceeds to Workouts page</p>
Test Results	PASS only if all post conditions are true, FAIL otherwise.

2.6.3 Test account persistence

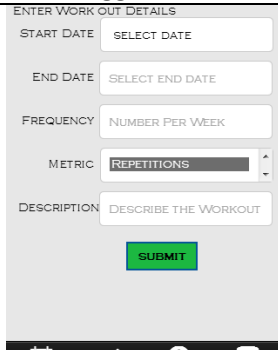
Requirement Number	3.2.2
Requirement Name	Account Requirements – Persist user account information when user is not logged into service.
Pre-conditions	All tests in section 2.9 – Database Requirements PASS
Procedures	<ul style="list-style-type: none"> • Verify that database tables persist between reboots of database machine
Post Conditions	Database tables remain consistent between machine reboots
Test Results	PASS if post conditions are true, FAIL database tables are dropped between reboots.

2.6.4 Test account updates

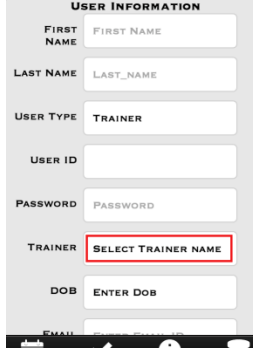
Requirement Number	3.2.2
Requirement Name	Account Requirements
Requirement Description	Update user account information when user is logged into service.
Pre-conditions	All tests in section 2.3 – Workout schedule requirements PASS All tests in section 2.4 – Validation requirements PASS All tests in section 2.5 – Server requirements PASS All tests in section 2.7 – User account requirements PASS
Procedures	N/A
Post Conditions	N/A
Test Results	PASS if all tests referenced in pre-conditions PASS, otherwise FAIL.

2.7 Workout schedule requirements

2.7.1 Test account registration

Requirement Number	3.2.3
Requirement Name	Workout Schedule Requirements
Requirement Description	<ul style="list-style-type: none"> Allow the user to create one or more workout schedules composed of a workout metric, a frequency per week, and a total number of weeks Allow the user to specify a workout metric as a number and a textual description
Pre-conditions	User is logged in and at the Add Workout screen
Procedures	<div>  </div> <ol style="list-style-type: none"> Select a start date Select an end date Type in a number of workouts per week (Frequency) Select a metric (Repetitions, Distance, or Time) Type a description Press "Submit"
Post Conditions	<p>If the Start Date is not a valid (a calendar date at the current date or later), an alert is displayed requesting the field be changed to indicate a valid date.</p> <p>If the End Date is not valid (a calendar date later than the Start Date), an alert is displayed requesting the field be changed to indicate a valid date.</p> <p>If the Frequency is not a positive integer number, an alert is displayed requesting the field be changed to a positive integer number.</p> <p>If Description is blank or contains non ASCII characters, an alert is displayed requesting the field be changed to contain at least one character and only valid ASCII characters.</p> <p>If all fields contain valid input, an alert is displayed stating "success", and the application proceeds to the Workouts page.</p>
Test Results	PASS only if all post conditions are true, FAIL otherwise.

2.7.2 Test user authentication preferences

Requirement Number	3.2.3
Requirement Name	Workout Schedule Requirements
Requirement Description	<ul style="list-style-type: none"> Allow the user to provide a user name to authenticate that workout metrics have been met for a particular workout schedule
Pre-conditions	<p>System contains more than one user account.</p> <p>User is logged in and at User Information page</p>
Procedures	<div data-bbox="548 520 799 865">  <p>USER INFORMATION</p> <p>FIRST NAME <input type="text"/></p> <p>LAST NAME <input type="text"/></p> <p>USER TYPE <input type="text" value="TRAINER"/></p> <p>USER ID <input type="text"/></p> <p>PASSWORD <input type="password"/></p> <p>TRAINER <input type="text" value="SELECT TRAINER NAME"/></p> <p>DOB <input type="text" value="ENTER DOB"/></p> <p>EMAIL <input type="text"/></p> </div> <ol style="list-style-type: none"> 1. Verify that “Trainer” dropdown field contains all other user account names. 2. Verify that selection of “Trainer” remains consistent after leaving and re-entering User Information page 3. Verify that selection of “Trainer” remains consistent after restarting application
Post Conditions	Trainer field contains all other user account names and field selection persists between application sessions and navigation
Test Results	PASS if post conditions are true, else FAIL

2.8 Database Requirements

2.8.1– User information fields test scenario

Reuirement Number	4.5.1
Requirement Name	Database Interface
Requirements Description	Users shall have access to input information into Use profile information fields
Pre conditions	Access to PHP my admin to run the query to login
Procedures	<ol style="list-style-type: none"> 1. Log into phpmy admin with credentials from server (root) 2. Navigate to the user table under the FH_DB database 3. Select SQL Button 4. Run The following SQL Code to insert the new user : INSERT INTO `fh_db`.`user` (`Balance`, `userID`, `DOB`, `Height`, `Weight`, `Password`, `EmailAddress`, `Name`, `Address`, `CCNum`, `Type`, `transID`, `LoginID`) VALUES ('33', '5678', '2014-02-11', '6', '180', 'password', 'someone@uri.edu', 'Joe Someone ', '50 lower college rd. Kingston RI 02882 ', '45989293', '1', '333', '34050'); 5. To update password or any other field, Run the SQL Query: UPDATE `user` SET `Password`=3333333 WHERE userID=1234 6. The user is now created or updated
Post Conditions	Both queries run successfully
Test Results	PASS

2.8.2-Trainer can input information test scenario

Reuirement Number	4.5.1
Requirement Name	Database Interface
Requirements Description	Trainers shall have access to input information into Trainer profile information fields.
Pre conditions	Access to PHP my admin to run the query to login
Procedures	<ol style="list-style-type: none"> 1. Log into phpmy admin with credentials from server (trainer account in this case) 2. Navigate to the activities table under the FH_DB database 3. Select SQL Button 4. Run The following SQL Code to validate the activity : 5. UPDATE `activities` SET `Validate` = 'Y' WHERE `userID`=125 6. The user's activity is now validated by the trainer
Post Conditions	Query run successfully
Test Results	PASS

2.8.3-Trainer shall have access to penalty/credit & activity goals

Reuirement Number	4.5.1
Requirement Name	Database Interface
Requirements Description	Trainers shall have access to Penalty, Credit and Goals options.
Pre conditions	Access to PHP my admin to run the query to login
Procedures	<ol style="list-style-type: none"> 1. Log into phpmy admin with credentials from server (trainer account in this case) 2. Navigate to the activities table under the FH_DB database 3. Select SQL Button 4. Run The following SQL Code to validate the activity : SELECT `PenAmt`,`Quantity` FROM `activities` WHERE `userID`=125 5. The user's goals and penalty amount is now displayed for the specific user (125 in this test case)
Post Conditions	Query run successfully
Test Results	PASS

2.8.4-Users can select a trainer

Reuirement Number	4.5.2
Requirement Name	Credit/Debit requirements
Requirement Description	Users can select a Trainer to associate with their account.
Pre conditions	Access to PHP my admin to run the query to login
Procedures	<ol style="list-style-type: none"> 1. Log into phpmy admin with credentials from server (trainer account in this case) 2. Navigate to the users table under the FH_DB database 3. Select SQL Button 4. Run The following SQL Code to find trainers : SELECT `userID`,`Name` FROM `user` WHERE `Type`='T' 5. The users that are trainers are now displayed with the userID and their name.
Post Conditions	Query run successfully
Test Results	PASS


2.8.5-Having a penalty & credit option

Reuirement Number	4.5.2
Requirement Name	Credit/Debit Requirements
Requirement Description	<p>Selecting Penalty option engages withdrawal workflow.</p> <p>Selecting Credit option engages credit workflow.</p>
Pre conditions	Access to PHP my admin to run the query to login
Procedures	<ol style="list-style-type: none"> 1. Log into phpmy admin with credentials from server (trainer account in this case) 2. Navigate to the activities table under the FH_DB database 3. Select SQL Button 4. Run The following SQL Code for the user to opt out of the penalty option UPDATE `activities` SET `PenAmt` = 0 WHERE `userID`=1234 5. This will now make the credit/debit amount set to 0 so any time the user misses or completes a workout there will be a balance of 0 added or deducted to the current credit.
Post Conditions	Query run successfully
Test Results	PASS

2.8.6-Audit Trail

Reuirement Number	4.5.3
Requirement Name	Compliance Requirements
Requirement Description	The database will have a functional audit trail
Pre conditions	Access to PHP my admin to run the query to login
Procedures	<ol style="list-style-type: none"> 1. Log into phpmy admin with credentials from server (trainer account in this case) 2. Navigate to the bank under the FH_DB database 3. Select SQL Button 4. Run The following SQL Code to get the current balance of all users audit : SELECT `userID`, `CurrentBalance` FROM `bank` WHERE 1 5. Select the Activities table under the FH_DB Database 6. Run The following SQL Code to get the un validated activities for all users audit: SELECT `userID`, `WorkoutID` FROM `activities` WHERE `Validate`='N' 7. There is now a list of the user's
Post Conditions	Query run successfully
Test Results	PASS

2.8.7-System will have limited access to authorized users

Reuirement Number	4.5.3
Requirement Name	Compliance Requirements
Requirement Description	The system will limit access to authorized users
Pre conditions	Access to PHP my admin to run the query to login
Procedures	<ol style="list-style-type: none"> 1. Log into phpmy admin with credentials from server (standard account in this case) 2. Navigate to the activities table under the FH_DB database 3. Select SQL Button 4. Run The following SQL Code to simulate the user trying to delete an activity DELETE FROM `activities` WHERE 1Select the Activities table under the FH_DB Database 5. PHPmyAdmin will then respond with the message below saying the user is working in the database and is currently logged in but does not have access to delete an activity 
Post Conditions	Query run successfully
Test Results	PASS

2.8.8-Trainer only has access to user's schedule

Reirement Number	4.5.4															
Requirement Name	Security Requirements															
Requirement Description	Trainers can enter their trainer profile, but can only see the users' schedule.															
Pre conditions	Access to PHP my admin to run the query to login															
Procedures	<div><div><div>1. Log into phpmy admin with credentials from server (trainer account in this case)</div><div>2. Navigate to the workout table under the FH_DB database</div><div>3. Select SQL Button</div><div>4. Run The following SQL Code will make the trainer only see the workout schedule for users SELECT `StartTime`,`EndTime`,`Date`,`WorkTime`,`userID` FROM `workout_db` WHERE 1</div><div>5. This will return the list of workout schedules for the users</div></div><div><table><tr><th>StartTime</th><th>EndTime</th><th>Date</th><th>WorkTime</th><th>userID</th></tr><tr><td>16:42:34</td><td>02:40:00</td><td>2014-03-26</td><td>00:30:00</td><td>12345</td></tr><tr><td>16:43:31</td><td>02:40:00</td><td>2014-03-26</td><td>00:30:00</td><td>12345</td></tr></table></div></div>	StartTime	EndTime	Date	WorkTime	userID	16:42:34	02:40:00	2014-03-26	00:30:00	12345	16:43:31	02:40:00	2014-03-26	00:30:00	12345
StartTime	EndTime	Date	WorkTime	userID												
16:42:34	02:40:00	2014-03-26	00:30:00	12345												
16:43:31	02:40:00	2014-03-26	00:30:00	12345												
Post Conditions	Query run successfully															
Test Results	PASS															

2.8.9-Trainer can update penalty amount

Reirement Number	4.5.4
Requirement Name	Security Requirements
Requirement Description	Trainers can control how penalties and credits are assessed to the user's account.
Pre conditions	Access to PHP my admin to run the query to login
Procedures	<ol style="list-style-type: none"> 1. Log into phpmy admin with credentials from server (trainer account in this case) 2. Navigate to the activities table under the FH_DB database 3. Select SQL Button 4. Run The following SQL code for the trainer to change the penalty amount for a selected user UPDATE `activities` SET `PenAmt`=20 WHERE `userID`=1234 5. This will change the penalty amount for a particular user.
Post Conditions	Query run successfully
Test Results	PASS

2.8.10-Trainer can control the actual amount of activity to calculate penalty

Reuirement Number	4.5.4
Requirement Name	Security Requirements
Requirement Description	Trainers control when the goals are met, which returns the penalties to the user account.
Pre conditions	Access to PHP my admin to run the query to login
Procedures	<ol style="list-style-type: none"> 1. Log into phpmy admin with credentials from server (trainer account in this case) 2. Navigate to the activities table under the FH_DB database 3. Select SQL Button 4. Run The following SQL code for the trainer to change the actual work out amount for a selected user so the penalty can be calculated. UPDATE `activities` SET `ActQuan`=20 WHERE `userID`=1234 5. This will put the amount in from the trainer for the penalty to be calculated off of.
Post Conditions	Query run successfully
Test Results	PASS

3.0 System Integration

3.1.1- Integration Phase -1 – Testing system components independently

The first step of system integration plan will consist of making sure each one of the system components are working according to the required specifications. In the FH application this can be defined as having the main system components fully operational and tested independently before trying to integrate or try to test any interaction with any of the other mayor system components. The phase-1 will validate the system have all the necessary components to start with a formal system integration. The integration phase will include, testing the client application, the server application and the database independently, see Figure 4- System Integration Phase -1 System Components.

Phase 1 Integration Testing:

- Testing the client app installation
- Testing GUI and classes that don't have dependencies with the Server or Database components.
- Testing database installation and making sure the tables and fields are created/available.
- Testing Apache server configuration and making sure server services are available (PHP/APACHE works)

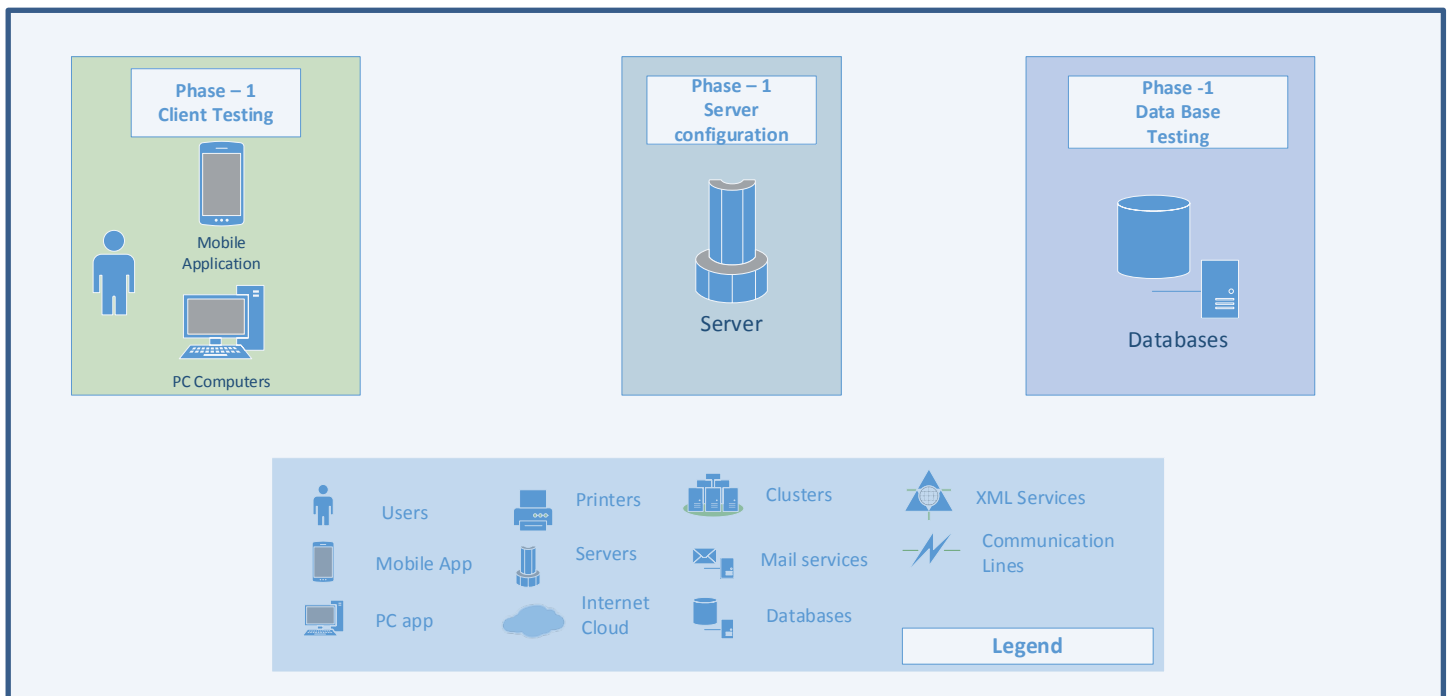


Figure 3- System Integration Phase -1 System components

3.2.1- Integration Phase -2 – Data Base and Server Communications, Functionality and Performance

The Phase 2 of the system integration will consist of integrating the database system with the server system. As part of this phase, we will test the communications interfaces (Ethernet/Wireless) between the Database and the Server system. In addition, this phase includes testing the functionality and the interaction between the server and database. This will be performed using a PHP script to send a set of requests to the MySQL database from the server and receiving back the expected information, with the right format in the server system.

Phase 2 Integration Testing:

- Testing Ethernet/Wireless protocol communication between the server and database
- Testing Server to Database Requests
- Testing Database to server Responses
- Testing Table and Fields formats after server request

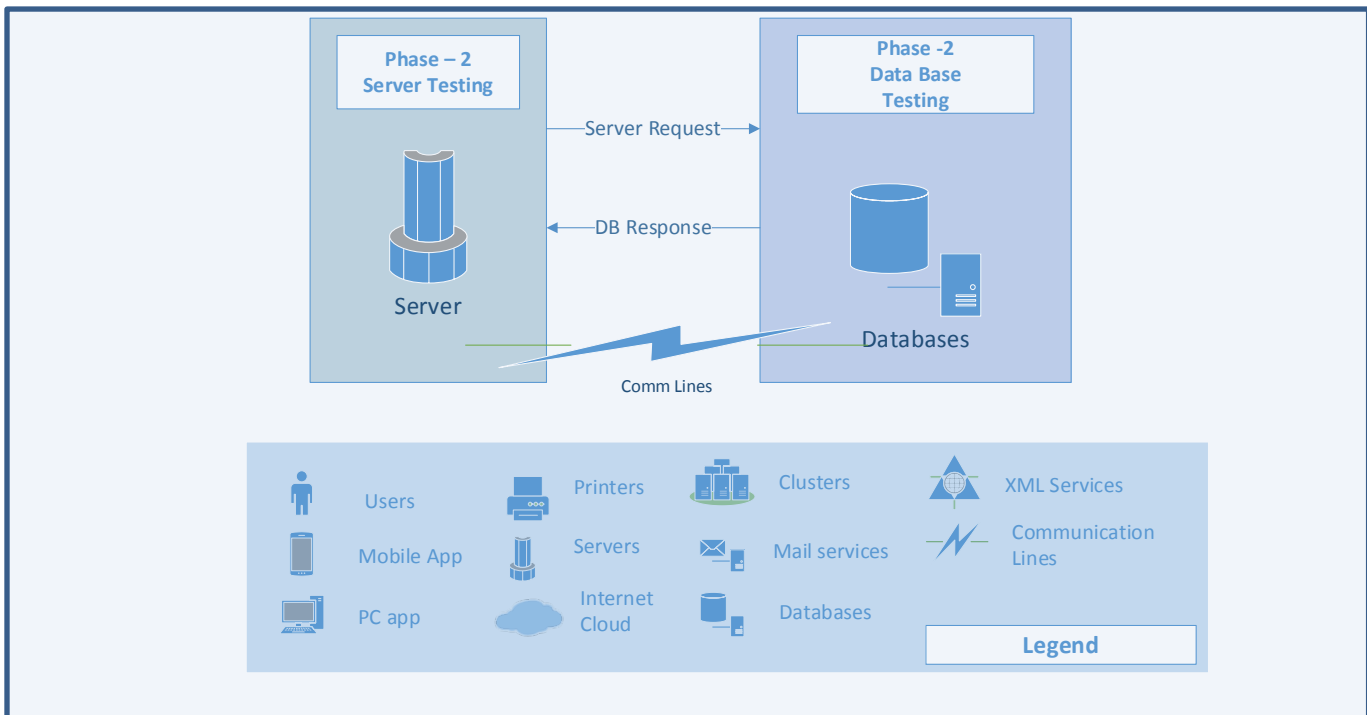


Figure 3.2.1- System Integration Phase -2 –Server and Database Integration

3.3.1- Integration Phase -3 –Client and Server Communications, Functionality and Performance

The Phase 3 of the system integration will consist of integrating the Android application with the server system. As part of this phase, we will test the communications interfaces (Ethernet/Wireless) between the Client Application (The Android operating system will provide the communication layer) and the Server system. In addition, this phase includes testing the functionality and the interaction between the client and server components. This phase will be tested using a client script application that will send and receive data objects as a JSON format from the server. The server must be able to receive data from the client to process it; this will be tested sending data from the client application, processing it in the server and displaying the results (e.g sending a set of numbers to by sum from the client and using the server to perform the calculations). In a summary, the client must be able to perform read and write operation on the server and the server must be able to process the client request. (See Figure 5 Client and Server Integration)

Phase 3 Integration Test

- Testing Ethernet/Wireless protocol communication between the client application and server
- Testing client to server requests
- Testing Server to client Responses
- Testing Data fields formats in the client application after sever request (Data consistency)

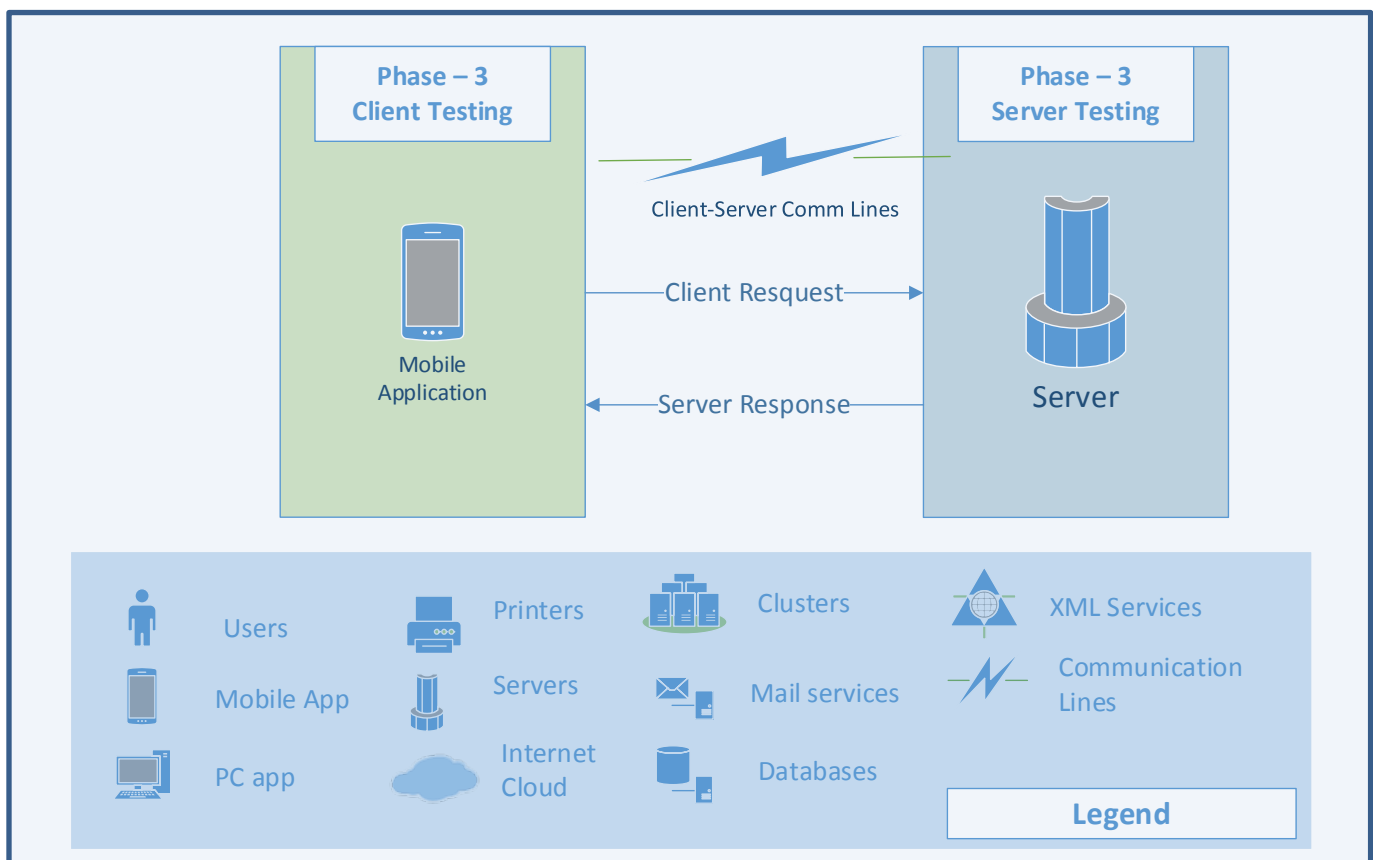


Figure 3.3.1-System Integration Phase 3 Client and Server Integration

3.4.1- Integration Phase -4 –Complete system Integration Communications, Functionality and Performance

The Phase 4 of the system integration will test and integrate all the system components as a one whole system. In theory, this phase of the integration should represent the whole system as a system of systems. In the FH application case this will encompass the client application, server scripting, database queries and responses and any additional interaction and functionality occurring in any other system operational layer. (See Figure 6 Complete System Integration)

Phase 4 Integration Test

- Testing Ethernet/Wireless protocol in between all system simultaneously (Client/Server and database)
- Testing client to server and server to database request
- Testing database to server and server to client responses
- Testing system performance requirements.

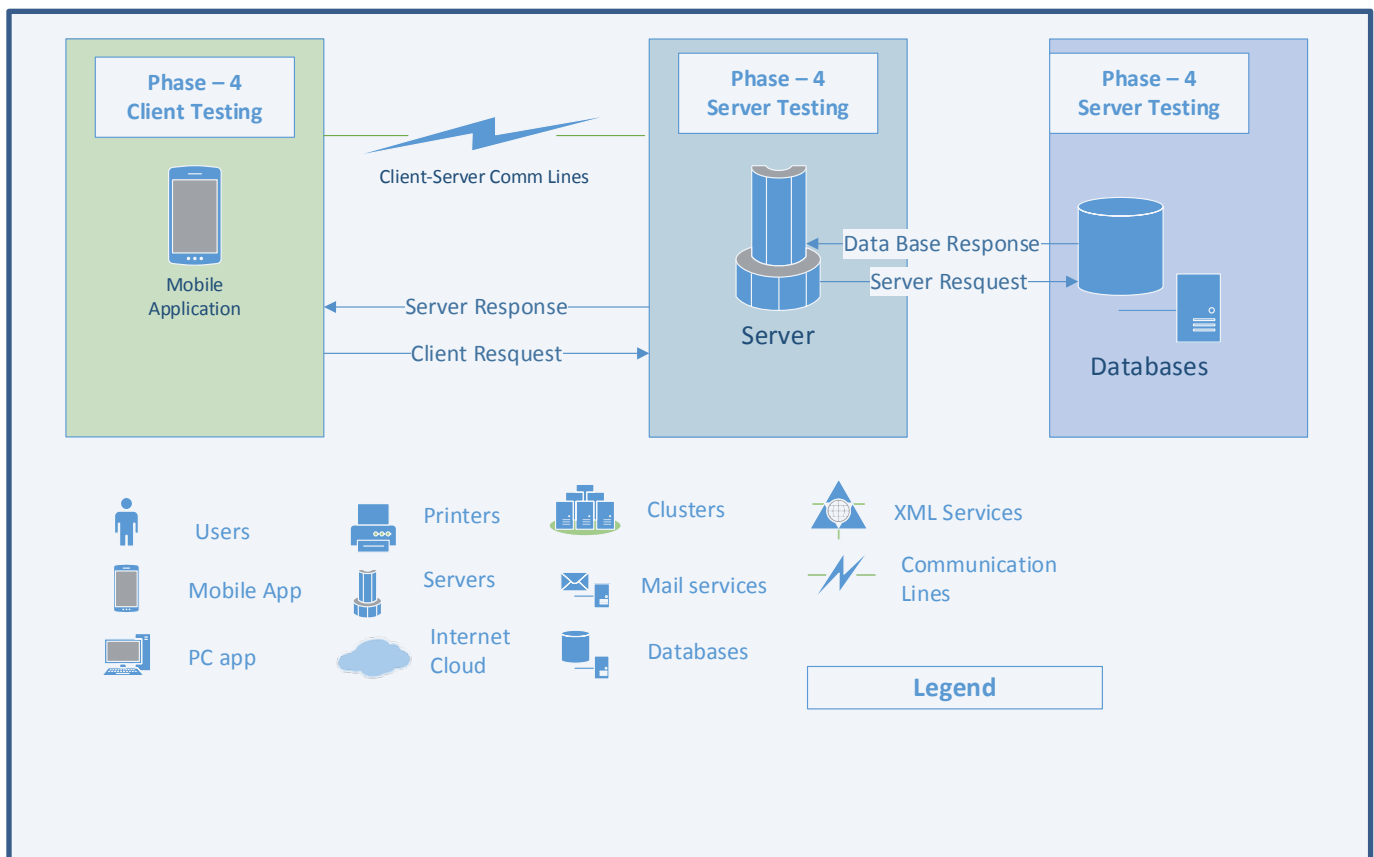


Figure 3.4.1--System Integration Phase 4 Complete System Integration

4.0 Testing Schedule

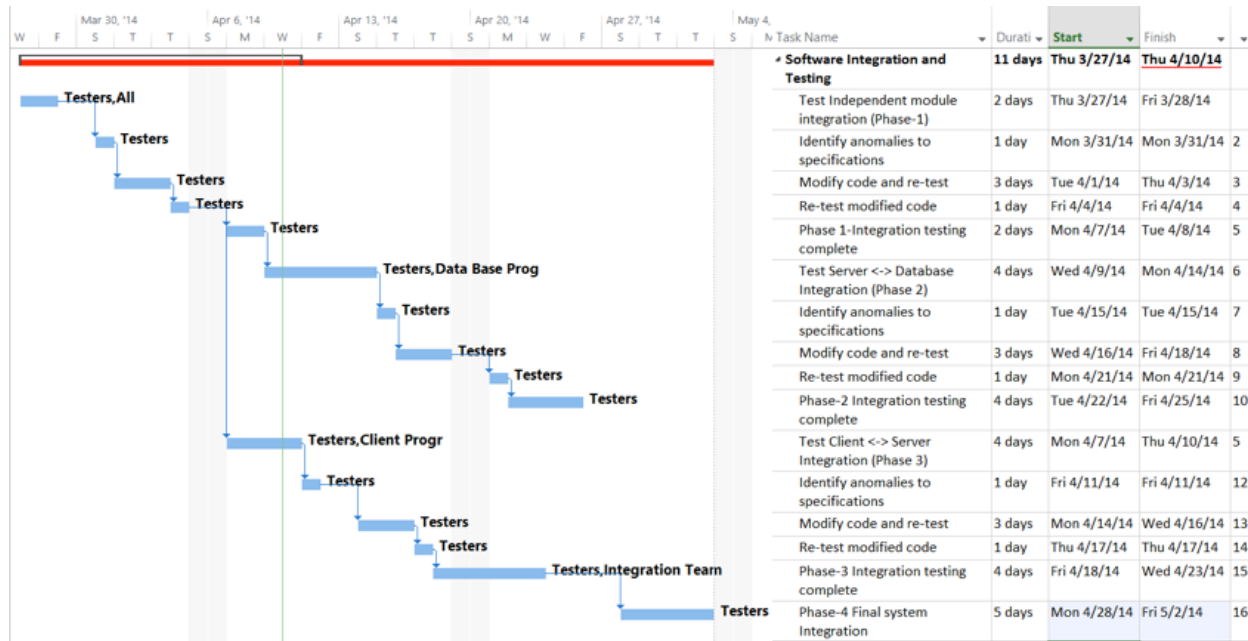


Figure 4.1 – Testing Schedule