Online Health Monitoring System

Test Results

COP4331, Fall, 2014

Modification history:

Version	Date	Who	Comment
v0.0	8/15/00	G. H. Walton	Template
v1.0	11/22/14	C. Chaffin	First Draft
v2.0	11/24/14	C. N. McCue	Added new defects and formatting

Team Name: Team 14

Team Members:

- Jon Carelli - email - web page
- Chris McCue - email - web page
- Chris Chaffin - email - web page
- Ethan Pitts - email - web page
- David Gundler - email - web page
- James Luke - email - web page

Contents of this Document

1. Introduction:

Overall Objective for Software Test Activity

Reference Documents

2. Description of Test Environment
3. Overall Stopping Criteria
4. Description of Individual Test Cases
5. Integration Testing
6. Conclusion

***We have inserted the original Deliverables I Test Plan for reference. After that, are sections for Integration Testing and Conclusions*

SECTION 1: Introduction

- Overall Objective for Software Test Activity:

  The objective of the test plan is to identify activities that will help produce an application with quality performance, usability, and functionality, by way of creating test cases and identifying bugs.

  The software test will ensure that our PHP functions are working as documented, our server connectivity is sufficient, our video chat service can support our target load, and that our web pages are displayed properly.

Reference Documents:

- Concept of Operations
- Software Requirement Specification
- Project Management Plan

SECTION 2: Description of Test Environment

We will run our tests on a Virtual Private Server, running a light LAMP stack. This is the most cost efficient and powerful combination for running our application on the web. The application requires at least 512MB of RAM, 1 GHz single core processor, and the ability to connect to the Internet. This test environment would be the same environment that the software would run in after deployment.

Developers will do initial testing, as development progresses users will be asked to test software and provide feedback. If a user finds a bug it will be recorded (create issue ticket with note that this was found by a user and not a team member) and sent to the development team for a solution. At this point, Chris Chaffin will be the tester. This, however, is subject to change.

SECTION 3: Stopping Criteria

The criteria for a stop will be rated according to the importance of the error.

Bugs will be classified on a scale of 1-4, one being the most critical and four representing a problem with a workaround.

- For a bug to be rated 1 there must be a critical error in the application. For example: on launching the application the program crashes before fully loading. Bugs with a rating of 1 will be the most important and of the highest priority.
- A rating of 2 is representative of a bug that needs to be addressed quickly and sometimes crashes the application.
- A rating of 3 is considered to be medium priority, but still needs to be addressed. This type of bug does not crash the system, but prevents some sort of work from being done.
- A rating of 4 is of the lowest priority. These types of bugs are changes that we would like to implement, but are not necessary for the end goal.

If errors are found during testing, the above criteria will be utilized to determine the seriousness of the bug. Bugs of 1 and 2 ratings will require immediate attention. If bugs of ratings 3 or 4 are encountered, we will continue the series of tests as far as possible, recording all bugs that appear. The results of the unit tests will indicate clearly which functions are problematic.

If no errors are found, it will be assumed that the software is working as intended, as long as the unit tests are thorough enough.

If the application is working as documented, meets specifications, and does not contain errors reachable by end users, it will be declared "good enough to deliver".

SECTION 4: Description of Individual Test Cases

- **Test Objective: Create a user from the public web form and activate it**
  - Test Description 1: We will test adding a user to the database by a web form, using the following data:
    - Email: any email reachable by the tester
    - Username: test_user
    - Password: UCF!Test

    Then, open the activation email and click on the activation URL

  - Test Conditions: We will run this test whenever the user signup form changes, and also if any database settings were changed.
  - Expected Results: You will be brought to a page that asks you to look for a confirmation email. An email with an activation URL will be sent to the tester's email address. The user will be added to the database.

- **Test Objective 2: Create and activate a doctor user from the public web form and activate it**
    - Test Description: We will test adding a doctor user to the database by a web form, using the following data:
        - Email: any email reachable by the tester
        - Username: test_doctor
        - Password: UCF!Test

      Then, open the activation email and click on the activation URL

    - Test Conditions: We will run this test whenever the user signup form changes, and also if any database settings were changed.
    - Expected Results: You will be brought to a page that asks you to look for a confirmation email. An email with an activation URL will be sent to the tester's email address. The user will be added to the database.
- **Test Objective 3: User(patient) logs in**
    - Test Description: We will test logging in as a user, using the following credentials:
        - Username: test_user
        - Password: UCF!Test
    - Test Conditions: We will run this test whenever the user login form changes, and also if any database settings were changed.
    - Expected Results: You will be brought to the patient-specific dashboard page
- **Test Objective 4: User(patient) requests a video chat**
    - Test Description: First complete test #3. Navigate to the chat request form. Request a chat from *test_doctor* for the next available time slot.
    - Test Conditions: We will run this test whenever the video chat request form changes.
    - Expected Results: test_doctor will be notified that the time slot has been filled. The timeslot table in the database will be updated.
- **Test Objective 5: Doctor logs in**
    - Test Description: We will test logging in as a doctor, using the following credentials:
        - Username: test_doctor
        - Password: UCF!Test
    - Test Conditions: We will run this test whenever the user login form changes, and also if any database settings were changed.
    - Expected Results: You will be brought to the doctor-specific dashboard page
- **Test Objective 6: Test video chat session**
    - Test Description: Log into the test_user and test_doctor accounts on separate computers. At the time scheduled in test #4, navigate to the video chat page on both machines.
    - Test Conditions: We will run this test whenever the video chat page changes.
    - Expected Results: The two users will be connected in a video chat session at the time scheduled in Objective 4.
- **Test Objective 7: Assigning medicine**
    - Test Description: Log into the test_doctor account. Navigate to the schedule reminder page and select the test_user account. Schedule test_user to take TEST_MEDICINE once daily.
    - Test Conditions: We will run this test whenever the schedule page changes.
    - Expected Results: The reminder will be added to the database, and the test_user account will get a notification to take TEST_MEDICINE.
- **Test Objective 8: Viewing reminders**
    - Test Description: Log into the test_user. Navigate to the reminder calendar.
    - Test Conditions: We will run this test whenever the reminder calendar changes.
    - Expected Results: A calendar should display all the reminders for test_user.
- **Test Objective 9: Manage account**

  - Test Description: After logging into test user, navigate to the manage account page. Enter the following information to update the account.
    - Name: John Doe
    - Phone Number: 555-555-5555
    - Email: any email reachable by the tester
    - Video Chat credentials: a valid username and password to log in to the video chat service
  - Test Conditions: We will run this test whenever the user account system changes.
  - Expected Results: The results of the search should include the test_doctor account.
- **Test Objective 10: Sending a message to the patient**
  - Test Description: Log in as test_doctor. Navigate to the messaging page and select test_user from the drop down list. Enter "Test Message" into the text field and hit the "Send Message" button.
  - Test Conditions: We will run this test whenever the messaging system changes.
  - Expected Results: test_user will see a popup on their page, displaying the test message.
- **Test Objective 11: Logging out of a session**
  - Test Description: After logging into test user, log out of the account. Attempt to access the reminder calendar page. You should get a page that requests a log in.
  - Test Conditions: We will run this test whenever the user account system changes.
  - Expected Results: You should not be able to access any information about the test_user account after logging out.

SECTION 5: Integration Testing

Throughout the programming phase several new test cases were developed for unforeseen bugs. Below you will find a list of tests added during the integration phase.

Test Case: Select Patient
Objective: When clicked, should display any pill schedules and appointments for selected patient.

Add some pill schedules and appointments for patient. These should be stored in database. To verify these are stored in the database, reload the page and, then, click "Select Patient".   The page reloads directly from the database. The page should reflect everything that was added. Controls to select patient are now disabled.

Test Case: Add Appointment
Objective: When clicked, should add an appointment to "Appointments" list and database.

Select a date and time for the appointment. Click "Add Appointment". The appointment should now be displayed in the "Appointments" list. Reload the page to make sure the appointment was stored in the database. The page reloads directly from the database.

Test Case: Invalid Appointment Date
Objective: Appointments with invalid dates will not be added to database or displayed in the "Appointments" list. Instead, a status message will alert the user as to which error has occurred.

Select “4/31/2014” and a time. Click “Add Appointment”. The status bar will display the message, “Please enter a valid date. This month only has 30 days.” No appointment should be added to the “Appointments” list. Reload the page to make sure the appointment was not stored in the database. The page reloads directly from the database.

Select “2/30/2014” and a time. Click “Add Appointment”. The status bar will display the message, “Please enter a valid date. February only has 28-29 days.” No appointment should be added to the “Appointments” list. Reload the page to make sure the appointment was not stored in the database. The page reloads directly from the database.

Select “2/29/2014” and a time. Click “Add Appointment”. The status bar will display the message, “Please enter a valid date. Non-leap years only have 28 days.” No appointment should be added to the “Appointments” list. Reload the page to make sure the appointment was not stored in the database. The page reloads directly from the database.

Test Case: Add Pill Schedule

Objective: When clicked, should add a pill schedule to “Pill Schedules” list and database.

Enter medication name. Select dosage, start date, duration, start time, times/day and end time. Click “Add Pill Schedule”. The pill schedule should now be displayed in the “Pill Schedules” list. Reload the page to make sure the appointment was stored in the database. The page reloads directly from the database.

Test Case: Invalid Pill Schedule Date

Objective: Pill schedules with invalid dates will not be added to database or displayed in the “Pill Schedules” list. Instead, a status message will alert the user as to which error has occurred.

Select “4/31/2014” and select other parameters. Click “Add Pill Schedule”. The status bar will display the message, “Please enter a valid date. This month only has 30 days.” No pill schedule should be added to the “Pill Schedules” list. Reload the page to make sure the pill schedule was not stored in the database. The page reloads directly from the database.

Select “2/30/2014” and select other parameters. Click “Add Pill Schedule”. The status bar will display the message, “Please enter a valid date. February only has 28-29 days.” No pill schedule should be added to the “Pill Schedules” list. Reload the page to make sure the pill schedule was not stored in the database. The page reloads directly from the database.

Select “2/29/2014” and select other parameters. Click “Add Pill Schedule”. The status bar will display the message, “Please enter a valid date. Non-leap years only have 28 days.” No pill schedule should be added to the “Pill Schedules” list. Reload the page to make sure the pill schedule was not stored in the database. The page reloads directly from the database.

Test Case: Non-entered Pill Name

Objective: Pill schedules without a name will not be added to database or displayed in the “Pill Schedules” list. Instead, a status message will alert the user that there is no pill name.

Enter all parameters except the pill name. Click “Add Pill Schedule”. The status bar will display the message, “Please enter a pill name”. No pill schedule should be added to the “Pill Schedules” list. Reload the page to make sure the pill schedule was not stored in the database. The page reloads directly from the database.

Test Case: Pill Schedule End Time Before Start Time

Objective: A start time must come before an end time. Pill schedules with this error will not be added to database or displayed in the “Pill Schedules” list. Instead, a status message will alert the user that this error has occurred.

Enter all parameters. Select an end time that comes before the start time. Click “Add Pill Schedule”. The status bar will display the message, “Please make sure that the start time is less than the end time”. No pill schedule should be added to the “Pill Schedules” list. Reload the page to make sure the pill schedule was not stored in the database. The page reloads directly from the database.

Test Case: Duplicate Pill Schedule Name
Objective: Pill schedules cannot have duplicate names. Pill schedules with duplicate names will not be added to database or displayed in the “Pill Schedules” list. Instead, a status message will alert the user that the duplicate cannot be accepted.

Enter a duplicate pill name and all other parameters. Click “Add Pill Schedule”. The status bar will display the message, “A pill schedule with this name already exists”. No pill schedule should be added to the “Pill Schedules” list. Reload the page to make sure the pill schedule was not stored in the database. The page reloads directly from the database.

Test Case: Duplicate Appointment Date-Time
Objective: Appointments cannot have duplicate date-times. Appointments with duplicate date-times will not be added to database or displayed in the “Appointments” list. Instead, a status message will alert the user that the duplicate cannot be accepted.

Enter a duplicate date and time parameters for an appointment. Click “Add Appointment”. The status bar will display the message, “An appointment is already scheduled at the same time”. No appointment should be added to the “Appointments” list. Reload the page to make sure the appointment was not stored in the database. The page reloads directly from the database.

Test Case: Delete Pill Schedule
Objective: When clicked, should delete a pill schedule from the “Pill Schedules” list and database.

Select pill schedule to delete from the “Pill Schedules” list. Click “Delete Pill Schedule”. The pill schedule should now be deleted from the “Pill Schedules” list. Reload the page to make sure the pill schedule was deleted from the database. The page reloads information directly from the database.

Test Case: No Pill Schedule Selected Delete
Objective: If no pill schedule is selected, no pill schedule can be deleted from the “Pill Schedules” list and database.

Click “Delete Pill Schedule” without selecting an item from the “Pill Schedules” list. The message “No pill schedules have been selected to delete.” should appear in the status bar. No pill schedules will be deleted. Reload the page to make sure the pill schedule was not deleted from the database, since the page reloads information directly from the database.

Test Case: Delete Appointment
Objective: When clicked, should delete an appointment from the “Appointments” list and database.

Select appointment to delete from the “Appointments” list. Click “Delete Appointment”. The appointment should now be deleted from the “Appointments” list. Reload the page to make sure the appointment was deleted from the database. The page reloads information directly from the database.

Test Case: No Appointment Selected Delete

Objective: If no appointment is selected, no appointment can be deleted from the “Appointments” list and database.

Click “Delete Appointment” without selecting an item from the “Appointments” list. The message “No appointments have been selected to delete.” should appear in the status bar. No appointments will be deleted. Reload the page to make sure the appointment was not deleted from the database, since the page reloads information directly from the database.

Test Case: Incorrect Captcha during Registration
Objective: Should give error when user enters wrong Captcha text

Load the registration form and add any dummy text to the fields. Be sure to enter the Captcha information incorrectly at the bottom. When the submit button is clicked, the next page will generate an error explaining that the Captcha was incorrect.

Test Case: Duplicate Email Change in Settings
Objective: Get an error message when a user attempts to change their email to another user’s address

Log into test_doctor (password is “password”). Go to the settings panel. Change the email address to test@test.test, which is the email address of another test account. Attempt to save the settings. You should get a duplicate error message at the bottom of the page.

Test Case: Appointment Send
Objective: When run send out appointment

When the timer hits the right time will pull information from the database and sent it out to the right recipient.

Test Case: Pill Send
Objective: When run send out pill

When the timer hits the right time will pull information from the database and sent it out to the right recipient.

Test Case: Correct Time to Send
Objective: setting boundaries to set the messages out by.

Setting an upper and lower bound so the correct messages are set with in a certain time frame.

Test Case: The Send
Objective: The ability to send.

Setting up the server with the capability to send the messages in a SMS manner.

Test Case: ON OFF
Objective: The ability to set on and off.

Setting up separate code that will able to turn on and off the SMS program.

Test Case: Updates to the Doctor
Objective: Sending updates to the doctor.

Sending the status of the SMS program such as if it has turn off and any fail logs.

Test Case: Long run.
Objective: The ability of extensively long running times.

Avoiding the time out of PHP.

Test Case: Advanced timer
Objective: A more controlled sleep timer.

Sleep timer with control to go off at intervals of X time in minutes.

Test Case: Correct Icon Loaded
Objective: The icon on the second tab should be different for doctors and patients.

Log into the system as a doctor. The icon on the second tab should depict a watch to represent the scheduling module.

Log into the system as a patient. The icon on the second tab should depict a calendar to represent the calendar module.

Test Case: Navigate as Doctor
Objective: A doctor should be able to navigate to each subpage, using the tabs on the left.

Log into the system as a doctor. Click on the first tab. The messages module should be loaded to the right.

Click on the second tab. The scheduling module should be loaded to the right.

Click on the third tab. The doctor settings module should be loaded to the right.

Test Case: Navigate as Patient
Objective: A patient should be able to navigate to each subpage, using the tabs on the left.

Log into the system as a patient. Click on the first tab. The messages module should be loaded to the right.

Click on the second tab. The calendar module should be loaded to the right.

Click on the third tab. The patient settings module should be loaded to the right.

Test Case: Navigate Calendar
Objective: A patient should be able to click through weeks, months, and years to view their schedule.

Log into the system as a user and select the calendar icon. Try selecting the month, year, weekly, arrows, and today buttons.

Test Case: Add Two Events on the Same Day
Objective: A patient should be able to have an appointment and take their medicine on the same day.

Log into the system as a doctor and enter medication name. Select dosage, start date, duration, start time, times/day and end time. Click “Add Pill Schedule”. The pill schedule should now be displayed in the “Pill Schedules” list. Reload the page to make sure the appointment was stored in the database. The page reloads directly from the database. Now, set up and appointment by selecting a date and time for the appointment. Click “Add Appointment”. The appointment should now be displayed in the “Appointments” list. Reload the page to make sure the appointment was stored in the database. The page reloads directly from the database. Navigate to the calendar on the selected day to view the results.

Test Case: Take Three Pills in One Day
Objective: A patient should be able to see how many times a day they should take their medicine.

Log into the system as a doctor and follow the previous test case “Add Pill Schedule” and select the number of times per day to three.

Test Case: Log in Two Users
Objective: Two or more different patients should be able to log in and view separate calendars.

Log into the system with two users at the same time and view the calendar simultaneously.

TEST RESULTS

Because of time constraints, we had to skip the coding prototype/unit testing phase and go straight to coding integration/integration testing. Except for James’ fault caught in the unit testing stage, the faults below were caught in the integration testing stage.

UNIT/INTEGRATION TESTING					
Test Case	**Who ran**	**When run**	**What environment**	**Result**	**Comments (only necessary for “fail”)**
Select Patient	CNM	11/4/14	Firefox 33.1	Fail	Without major revision, the code will not integrate with the database. Looking to use forms with php.
Add Appointment	CNM	11/4/14	Firefox 33.1	Fail	Without major revision, the code will not integrate with the database. Looking to use forms with php.
Add Pill Schedule	CNM	11/4/14	Firefox 33.1	Fail	Without major revision, the code will not integrate with the database. Looking to use forms with php.
Delete Appointment	CNM	11/4/14	Firefox 33.1	Fail	Without major revision, the code will not integrate with the database. Looking to use forms with php.
Delete Pill Schedule	CNM	11/4/14	Firefox 33.1	Fail	Without major revision, the code will not integrate with the database. Looking to use forms with php.
Invalid Appointment Date	CNM	11/11/14	Firefox 33.1	Fail	No error check implemented yet.
Invalid Pill Schedule Date	CNM	11/11/14	Firefox 33.1	Fail	No error check implemented yet.
Non-entered Pill Name	CNM	11/11/14	Firefox 33.1	Fail	No error check implemented yet.
Pill Schedule End Time Before Start Time	CNM	11/11/14	Firefox 33.1	Fail	No error check implemented yet.

No Pill Schedule Selected Delete	CNM	11/11/14	Firefox 33.1	Fail	Tries to delete nothing from database.
No Appointment Selected Delete	CNM	11/11/14	Firefox 33.1	Fail	Tries to delete nothing from database.
Invalid Appointment Date	CNM	11/11/14	Firefox 33.1	Pass	Error-checking implemented.
Invalid Pill Schedule Date	CNM	11/11/14	Firefox 33.1	Pass	Error-checking implemented.
Non-entered Pill Name	CNM	11/11/14	Firefox 33.1	Pass	Error-checking implemented.
Pill Schedule End Time Before Start Time	CNM	11/11/14	Firefox 33.1	Pass	Error-checking implemented.
No Pill Schedule Selected Delete	CNM	11/11/14	Firefox 33.1	Pass	Error-checking implemented.
No Appointment Selected Delete	CNM	11/11/14	Firefox 33.1	Pass	Error-checking implemented.
Select Patient	CNM	11/15/14	Firefox 33.1	Fail	Only can use dummy doctor and patient at this time. Separate page from rest of pill/appointment manager. Cannot integrate with the Navigator.
Add Appointment	CNM	11/14/14	Firefox 33.1	Fail	Must use different pages to accomplish. Cannot integrate with the Navigator.
Add Pill Schedule	CNM	11/14/14	Firefox 33.1	Fail	Must use different pages to accomplish. Cannot integrate with the Navigator.
Delete Appointment	CNM	11/14/14	Firefox 33.1	Fail	Must use different pages to accomplish. Cannot integrate with the Navigator.
Delete Pill Schedule	CNM	11/14/14	Firefox 33.1	Fail	Must use different pages to accomplish. Cannot integrate with the Navigator.
Duplicate Pill Schedule Name	CNM	11/15/14	Firefox 33.1	Fail	Allows for duplicate pill schedules.
Duplicate Appointment Name	CNM	11/15/14	Firefox 33.1	Fail	Allows for duplicate appointments.
Select Patient	CNM	11/15/14	Firefox 33.1	Pass	On one page. Will integrate with Navigator. Disables after patient selected.
Add Appointment	CNM	11/15/14	Firefox 33.1	Pass	On one page. Will integrate with Navigator.
Invalid Appointment Date	CNM	11/15/14	Firefox 33.1	Pass	Displays error in status bar now.
Add Pill Schedule	CNM	11/15/14	Firefox 33.1	Pass	On one page. Will integrate with Navigator.
Invalid Pill Schedule Date	CNM	11/15/14	Firefox 33.1	Pass	Displays error in status bar now.
Non-entered Pill Name	CNM	11/15/14	Firefox 33.1	Pass	Displays error in status bar now.
Pill Schedule End Time Before Start Time	CNM	11/15/14	Firefox 33.1	Pass	Displays error in status bar now.
Duplicate Pill Schedule Name	CNM	11/15/14	Firefox 33.1	Pass	Displays error in status bar now.
Duplicate Appointment Date-Time	CNM	11/15/14	Firefox 33.1	Pass	Displays error in status bar now.
Delete Pill Schedule	CNM	11/15/14	Firefox 33.1	Pass	On one page. Will integrate with Navigator.
No Pill Schedule Selected Delete	CNM	11/15/14	Firefox 33.1	Pass	Displays error in status bar now.
Delete Appointment	CNM	11/15/14	Firefox 33.1	Pass	On one page. Will integrate with Navigator.
No Appointment Selected Delete	CNM	11/15/14	Firefox 33.1	Pass	Displays error in status bar now.
User Login	JL	10/16/14	Firefox 33.1	Pass	
User Register	JL	10/16/14	Firefox 33.1	Pass	
Doctor Register	JL	10/16/14	Firefox 33.1	Pass	
Duplicate Email	JL	10/16/14	Firefox 33.1	Fail	Email allowed to be a duplicate for now.
Duplicate Email	JL	10/18/14	Firefox 33.1	Pass	
Update Settings	JL	10/18/14	Firefox 33.1	Pass	
Incorrect Captcha	JL	11/15/14	Firefox 33.1	Pass	

Send Ability	DJG	11/10/14	XAMPP	Fail	The send function on the new server is not set up for mailing.
Long run time	DJG	11/2/14	XAMPP/PHP	Fail	30 Sec max runtime
Long run time	DJG	11/3/14	XAMPP/PHP	Pass	set_time_limit(0); to allow the time to run.
Correct time to send out	DJG	11/3/14	XAMPP/PHP	Fail	Working but not to an acceptable state. Has problems with minutes to hours to days ect.
Advanced timer	DJG	11/3/14	XAMPP/PHP	Fail	Working with sleep(900); but not acceptable to static.
The send out	DJG	11/4/14	XAMPP/PHP	Fail	Working but still has bugs with possible fall out massages to going.
pill and appointments send out	DJG	11/5/14	XAMPP/PHP	Fail	Working but still needs formatting for the massages.
pill and appointments send out	DIG	11/6/14	XAMPP/PHP	Fail	No database to pull from yet.
Send Ability	DJG	11/12/14	XAMPP/PHP	Pass	Set up the server with phpMailer and to the necessary function.
Pill and Appointments send out	DJG	11/12/14	XAMPP/PHP	Fail	Set up a Gmail account and set the Sever to it still no database information.
Updates to the doctor	DJG	11/13/14	XAMPP/PHP	Fail	Set up the fail logs for the doctor but still need to set up a way to run it all.
Pill and Appointments send out	DJG	11/14/14	XAMPP/PHP	Fail	More advanced send but possible fall through of messages.
Pill and Appointments send out	DJG	11/15/2014	SERVER/PHP	Pass	Working with full information from the database.
Correct time to send out	DJG	11/16/2014	SERVER/PHP	Pass	New algorithm for sending out the massages that looks +15 minutes forward and -15 back.
Advanced timer	DJG	11/16/2014	SERVER/PHP	Pass	New algorithm for calculating the time to the next run.
ON OFF	DJG	11/16/2014	SERVER/PHP	Pass	New code to start and stop the program.
Updates to the Doctor	DJG	11/16/2014	SERVER/PHP	Pass	Now can tell doctor if the SMS is turned off.
Full integration in to the database.	DJG/ JL	11/17/2014	SERVER/PHP	Pass	Fully pulling and send the data to the patient.
Correct Icon Loaded	EMP	11/17/14	Safari 8.0	Pass	
Navigate as Doctor	EMP	11/17/14	Safari 8.0	Pass	
Navigate as Patient	EMP	11/17/14	Safari 8.0	Pass	
Navigate Calendar	CLC	11/10/14	Chrome 38.0	Pass	
Add Two Events on the Same Day	CLC	11/17/14	Chrome 38.0	Pass	
Take Three Pills in One Day	CLC	11/17/14	Chrome 38.0	Pass	The pills should have an extended duration time for the day rather than the normal 30 minute block.
Login as Two Users	CLC	11/17/14	Chrome 38.0	Fail	The calendar currently reads from one file causing loss of data.
Default date will not change	JC	11/15/14	Firefox 33.1	Fail	Fixed
Incorrectly formatted day coming in from the database into the	JC	11/15/14	Firefox 33.1	Fail	Fixed

JSON document, this was causing all of the appointments and medicines to be landing on Jan 1st 2014/2015.					
The calendar was named json.php.	JC	11/15/14	Firefox 33.1	Fail	Fixed
Would not send message when striking enter.	JC	11/15/14	Firefox 33.1	Fail	Fixed
Page would not refresh history on load.	JC	11/15/14	Firefox 33.1	Fail	Fixed
History would not refresh when a new message was received.	JC	11/15/14	Firefox 33.1	Fail	Fixed
JSON array was throwing an error when trying to print it.	JC	11/15/14	Firefox 33.1	Fail	Fixed
Change onload from HTML to JavaScript call	CNM	11/18/14	Firefox 33.1	Fail	Fixed
Change CSS style to make pill/appointment manager presentable	CNM	11/18/14	Firefox 33.1	Fail	Fixed

As we tested the application, errors were found and corrected.

SYSTEM TESTING					
Test Case	**Who Ran**	**When Ran**	**Environment**	**Result**	**Comments (only necessary for “fail”)**
Military time issues with noon and midnight	DG discovered	11/17/14	Firefox 33.1	Fail	Fixed by CNM
No doctor’s name for appointments and pill schedules	DG discovered	11/19/14	Firefox 33.1	Fail	Fixed by CNM
Small time step causes infinite loop	DG discovered	11/21/14	Firefox 33.1	Fail	Fixed by CNM
Calendar not showing times	CNM discovered	11/21/14	Firefox 33.1	Fail	Fixed by JC
Chat time off	CNM discovered	11/21/14	Firefox 33.1	Fail	Fixed by JC

SECTION 6: Conclusion

All major test objectives were passed during the integration phase with the exception of video chat, which was an optional feature. Therefore test objective four must be disregarded for the end results.

Throughout the integration phase we found several new test cases, which the majority of were handled. A few of the test cases slipped through due to time constraints, but overall the system is working according to the SRS.

Template created by G. Walton (GWalton@mail.ucf.edu) on March 28, 1999 and last modified on August 15, 2000.

This page last modified by Chris McCue (christopher.mccue@knights.ucf.edu) on 11/24/14