Online Health Monitoring System

Deliverables III

COP 4331, Fall, 2014

Team Name: Team 14

Team Members:

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

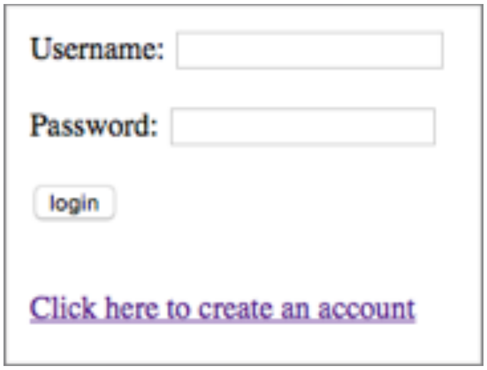
Contents of this Document

Deliverables III

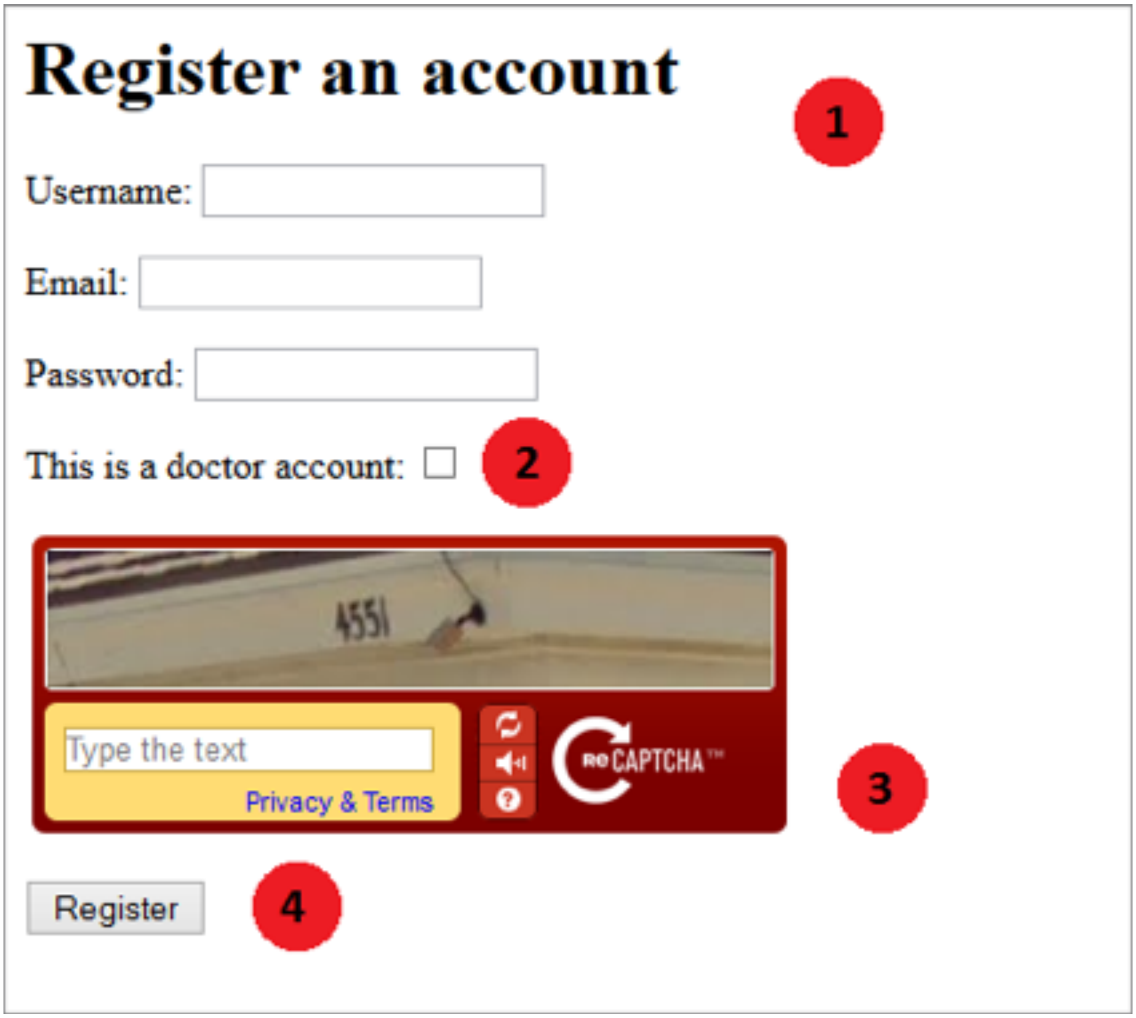
User Manual

I. Logging in

To start using the Online Health Monitoring System, you first must create an account. To do this from the login screen, just click on the “Click here to create an account” hyperlink, below the login credentials. If you already have an account, just enter your login credentials in the fields shown, and click “login.”



If you are creating a new account, you will see the registration screen. Fill out all the information on this page, and click “Register” to create your account. Note that you must have a unique email and have correctly entered the CAPTCHA to be able to register.

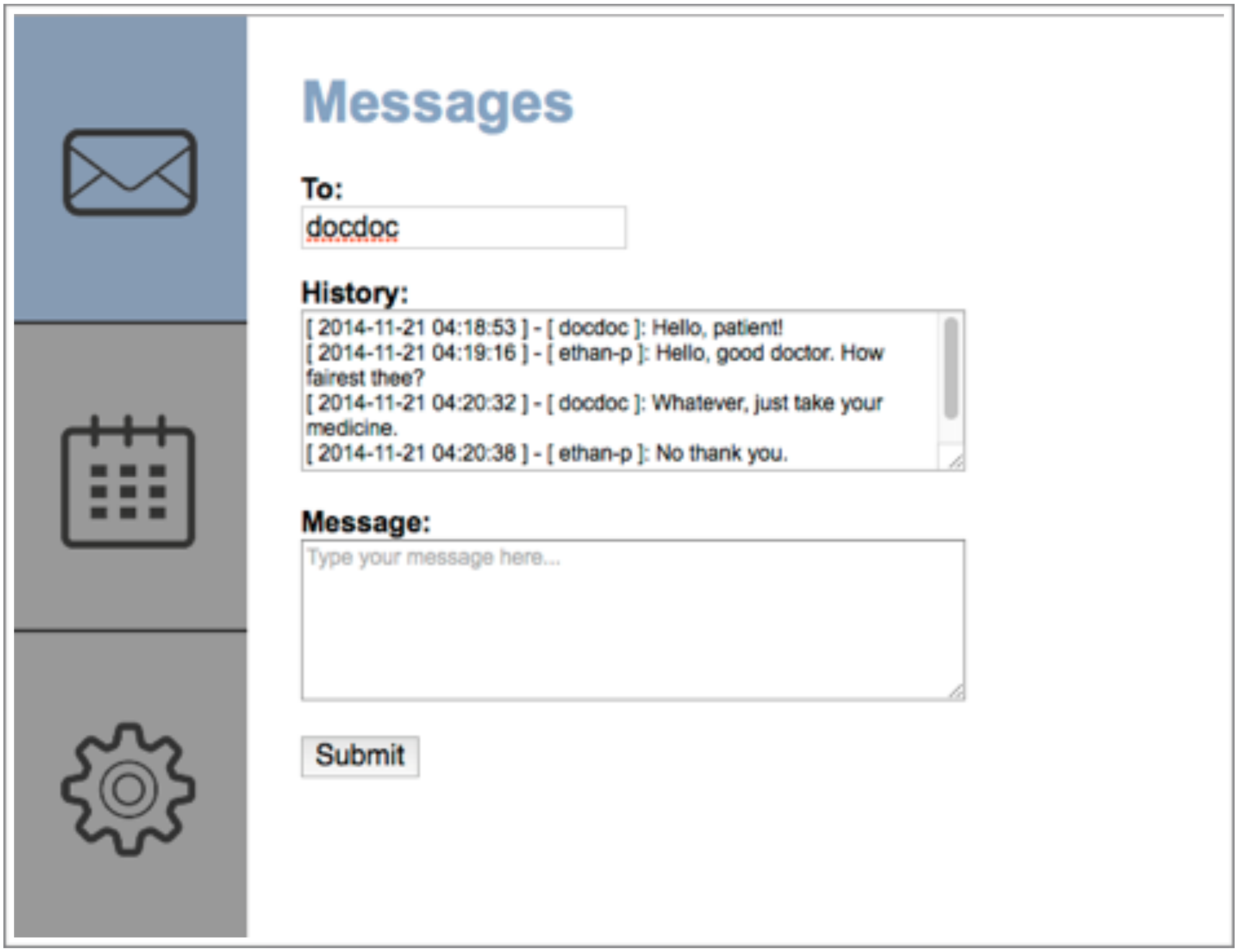


Each part of the registration form will be explained in detail here (refer to below image for number references):

1. Choose a unique username and an email address that has not been registered in the system yet.
2. Check this box if you are registering an account as a doctor.
3. Enter the text shown in the CAPTCHA image. This helps prevent bots from flooding our database with users.
4. Once every field is entered, click on the Register button to create your account.

II. Messages

When you first log in, you will by default be sent to the messages module. You can also navigate to the Messages module by clicking on the tab indicated with a mail icon.



Messages

To: docdoc

History:
[2014-11-21 04:18:53] - [docdoc]: Hello, patient!
[2014-11-21 04:19:16] - [ethan-p]: Hello, good doctor. How fairest thee?
[2014-11-21 04:20:32] - [docdoc]: Whatever, just take your medicine.
[2014-11-21 04:20:38] - [ethan-p]: No thank you.

Message: Type your message here...

To send a message to a user, just type the username in the “To:” field, type the message you would like to send in the “Message:” field, and click “Submit,” or press the enter key. Messages sent to you from a user will show up in the history when you type their name in the “To:” field.

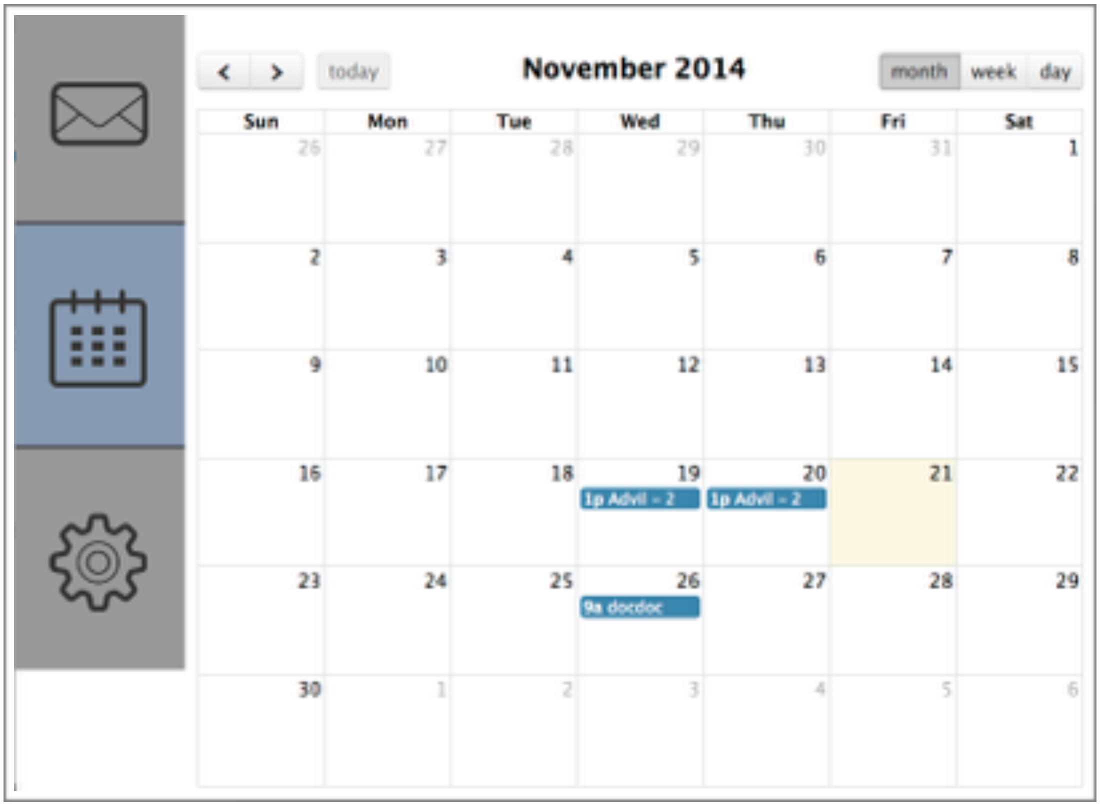
III. Calendar (Patients Only)

The calendar is used as a viewing tool for patients to see their upcoming appointments and medicine schedules. Navigate to the calendar by logging in as a patient, and clicking on the tab indicated by the calendar icon. Patients cannot manipulate the calendars events in any capacity. They can however, change viewing options to see previous or future month.

The Calendar can be arranged in three different viewing options, month, week, and day. Each one respectively will arrange the calendar’s view accordingly. Furthermore, there are two arrow keys that skip by month, week, or day depending on the selection. If you happen to get to far away from the current date, you can click the today button and you will be returned to the present date.

- Appointments will be displayed in the following way: 6:00a Doctor_Bob
- Medicine schedules will be displayed in the following way: 12:12 Tylenol - 2

Below is an example of a calendar with two medications and one appointment scheduled.

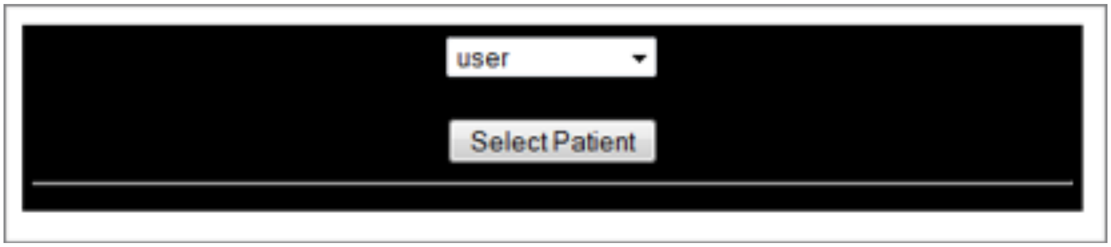


November 2014

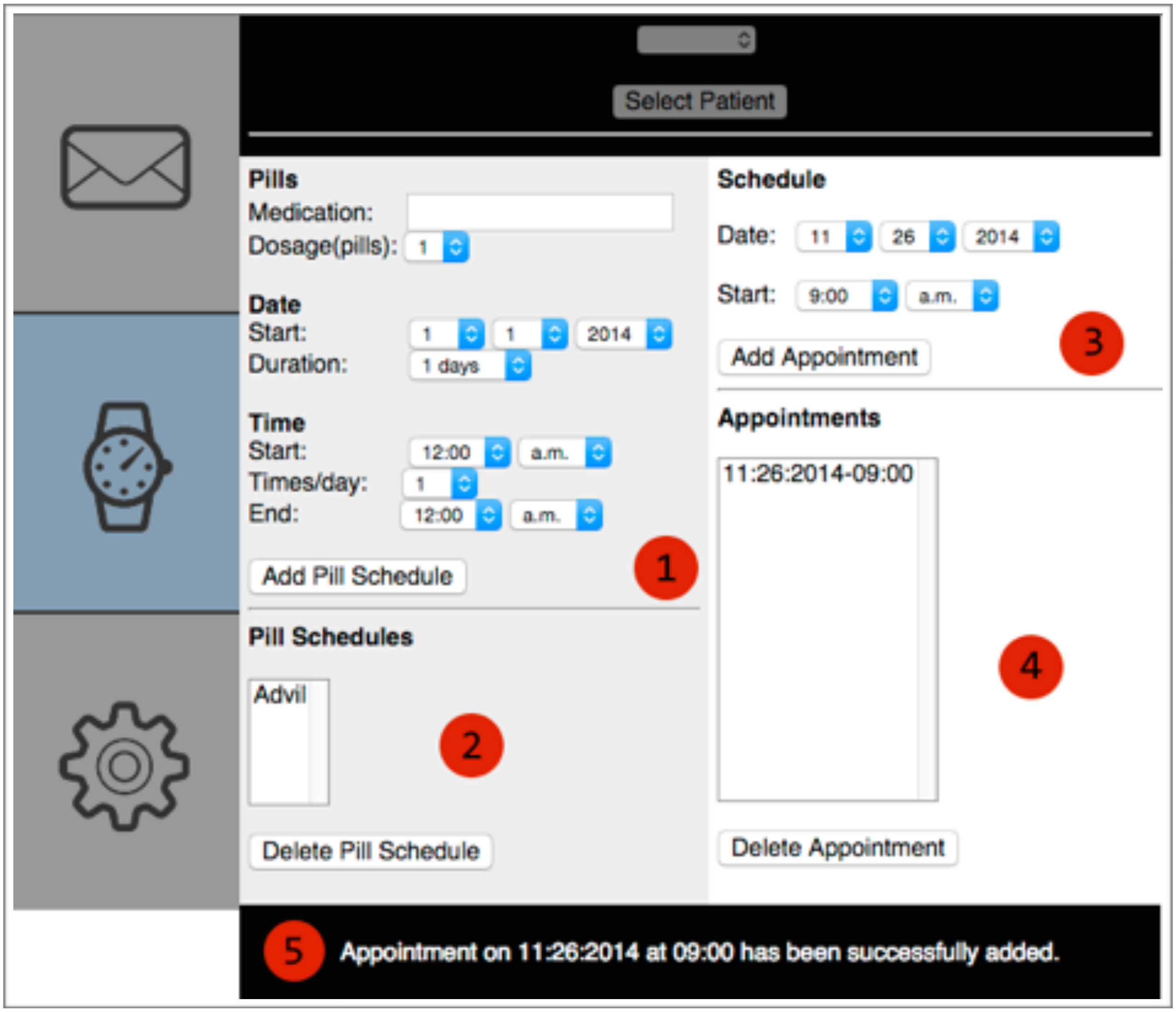
Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19 1p Advil - 2	20 1p Advil - 2	21	22
23	24	25	26 9a docdoc	27	28	29
30	1	2	3	4	5	6

IV. Scheduler (Doctor Only)

The scheduler is used by a doctor to set medication reminders, and make appointments with patients. Navigate to the scheduler by logging in as a doctor, and clicking on the tab indicated by the watch icon. Select a patient from the dropdown list and click “Select Patient”.



The screen should look like the one below.



Pills
Medication:
Dosage(pills): 1

Date
Start: 1 1 2014
Duration: 1 days

Time
Start: 12:00 a.m.
Times/day: 1
End: 12:00 a.m.

Pill Schedules
Advil

Schedule
Date: 11 26 2014
Start: 9:00 a.m.

Appointments
11:26:2014-09:00

Appointment on 11:26:2014 at 09:00 has been successfully added.

The Pill/Appointment Manager has five sections:

1. “Select Patient” section
2. “Add Pill Schedule” section
3. “Delete Pill Schedule” section
4. “Add Appointment” section
5. “Delete Appointment” section
6. Status section

Each of these sections is described below in detail.

1. “Add Pill Schedule” section

- Type a pill name next to the “Medication” label.
- Select a “Dosage(pills)”.
- Select month, day and year of the start date.
- Select the number of days this pill schedule will cover.
- Select time and a.m. or p.m. indicator for the pill start time.
- Select “Times/day”.
- Select time and a.m. or p.m. indicator for the pill end time.
- Click “Add Pill Schedule”. If there are no user entry errors, a pill schedule will be added to the “Pill Schedules” list and the database. The pill schedule name will be the name entered. This pill schedule will appear in the Calendar and trigger SMS Alerts. The status section will indicate that a pill schedule has been added.

Reasons a pill schedule may not be added:

- No pill name is entered
- Duplicate pill name (already in “Pill Schedules” list)
- Day “31” selected for months with only 30 days (April, June, September, November)
- Day “30” or “31” selected for February
- Day “29” selected for February in a non-leap year
- End time is less than start time

The status section will indicate all of the above input errors.

2. Delete "Pill Schedule" section

- Select pill schedule to delete
- Click "Delete Pill Schedule" button. Pill schedule will be deleted from "Pill Schedules" list and the database. The status section will indicate that a pill schedule has been deleted.
- If no pill schedule is selected or the "Pill Schedules" list is empty and the "Delete Pill Schedule" button is clicked, the status section will indicate that no pill schedule was selected.

3. "Add Appointment" section

- Select month, day and year of the appointment date.
- Select time and a.m. or p.m. indicator for the appointment time.
- Click "Add Appointment". If there are no user entry errors, an appointment will be added to the "Appointments" list and the database. The appointment name will be in the format "MM:DD:YYYY-HH:MM". Time is in military time format. These appointments will appear in the Calendar and trigger SMS Alerts. The status section will indicate that an appointment has been added.

Reasons an appointment may not be added:

- Duplicate appointment (already in "Appointments" list)
- Day "31" selected for months with only 30 days (April, June, September, November)
- Day "30" or "31" selected for February
- Day "29" selected for February in a non-leap year

The status section will indicate all of the above input errors.

4. Delete "Appointment" section

- Select appointment to delete
- Click "Delete Appointment" button. The appointment will be deleted from "Appointments" list and the database. The status section will indicate that an appointment has been deleted.

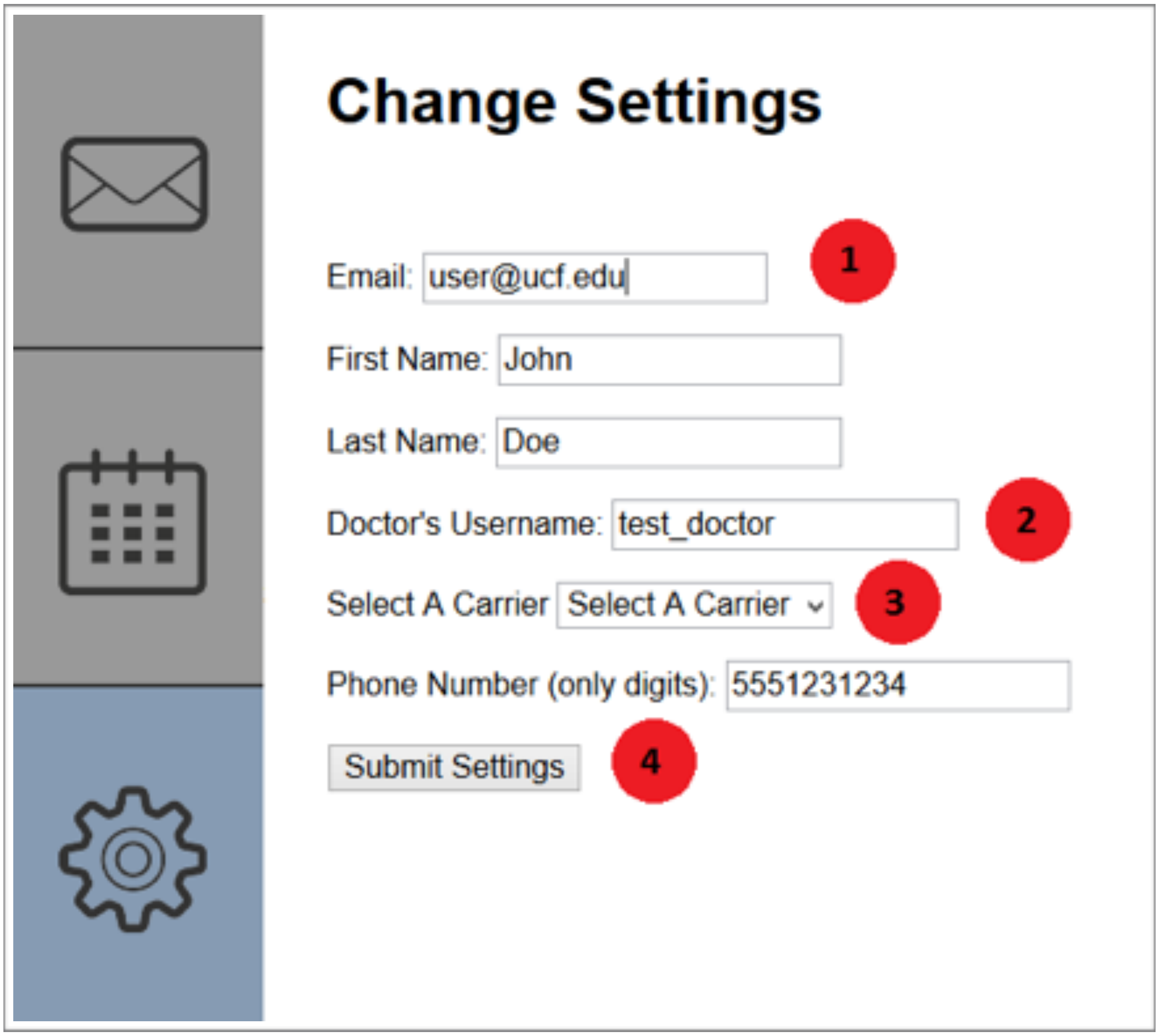
- If no appointment is selected or the “Appointments” list is empty and the “Delete Appointment” button is clicked, the status section will indicate that no appointment was selected.

6. Status section

- This section displays the status of addition and deletion of pill schedules and appointments. If there are any user input errors, this will also be indicated in this section.

V. Settings

The settings are used to change user information. Navigate to the settings by clicking on the tab indicated by the gear icon. The settings module is shown below, along with a description of its features.



Change Settings

Email: user@ucf.edu

First Name: John

Last Name: Doe

Doctor's Username: test_doctor

Select A Carrier: Select A Carrier

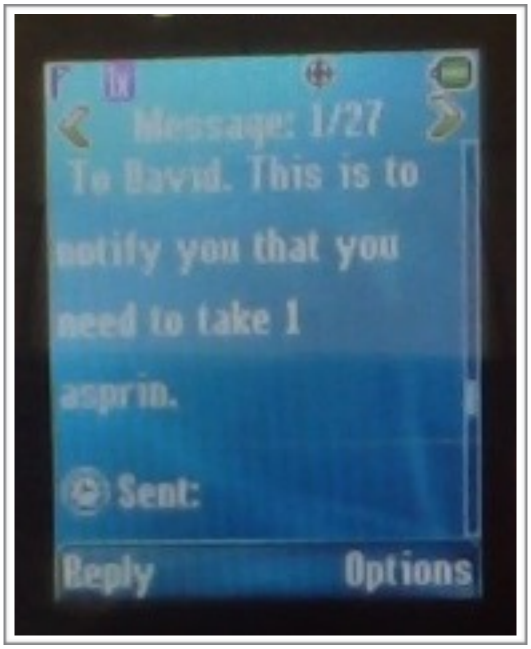
Phone Number (only digits): 5551231234

1. You may change you email here. It must be an address that is not being used by another account.
2. Type the username of your doctor here to link your account to theirs. Note that this will become the doctor that will set your appointments and medication reminders.
3. Select your cell phone’s carrier here from the dropdown menu.
4. Clicking Submit Settings will save everything you entered in this form.

VI. SMS (Patients Only)

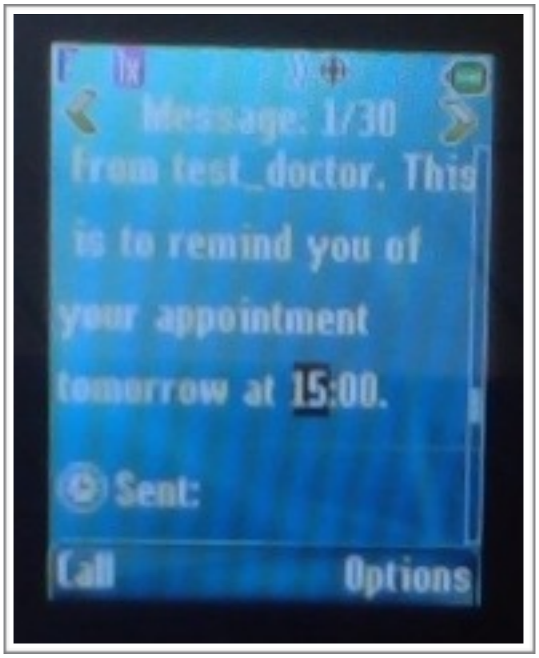
The SMS feature reminds patients to take their medication, or to send reminders of upcoming appointments with their doctor. To receive SMS reminders, the patient must add their cell phone number and carrier in the settings tab.

The patient’s doctor must set the reminders under the scheduler tab. When the patient has a medication reminder, they will receive a reminder, via SMS, within 15 minutes of the time they should take the medication. Below is an example of an SMS medication reminder.



To David. This is to notify you that you need to take 1 asprin.

When the patient has an appointment reminder, they will receive a reminder, via SMS, the day before the appointment is scheduled. Below is an example of an appointment reminder.



From test_doctor. This is to remind you of your appointment tomorrow at 15:00.

Sent:

Build Instructions for Online Health Management System

First we need to begin with an Ubuntu server. Our team used Ubuntu 14.04, the most recent release with long term support. Our server is a virtual private server running on Microsoft's Azure cloud service, but there are many other providers such as Amazon, Linode, and DigitalOcean that all have attractive offers. If you want to host it yourself, you could also run it inside a virtual machine on your computer.

Here are the steps to setting up the LAMP stack (Linux, Apache, MySQL, PHP) on the server. I used the instructions detailed here https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-14-04

I will summarize the actions needed to take here:

1. Log on the server via SSH
2. To update the package list on the server, run

   ```
   sudo apt-get update
   ```

3. Next, we will install Apache. Run this command:

   ```
   sudo apt-get install apache2
   ```

4. To install and configure MySQL, Run these commands:

   ```
   sudo apt-get install mysql-server php5-mysql
   ```

   ```
   sudo mysql_install_db
   ```

   ```
   sudo mysql_secure_installation
   ```

5. Now we will install php:

```
sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

Now we will set up the database with all of the required tables.

1. Access the MySQL shell as root by using this command

   ```
   mysql -u root -p
   ```

2. Create a database called *project*:

   ```
   CREATE DATABASE project;
   ```

3. Now we need to specify that our actions will be for this database. Enter:

   ```
   USE project;
   ```

4. Lets create the table to store user account information:

```
5. CREATE TABLE IF NOT EXISTS `users` (
6. `user_id` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
7. `email` varchar(30) NOT NULL,
8. `username` varchar(16) NOT NULL,
9. `fname` varchar(20) DEFAULT NULL,
10.  `lname` varchar(20) DEFAULT NULL,
11.  `doctorName` varchar(16) DEFAULT NULL,
12.  `isDoctor` char(1) NOT NULL,
13.  `hashPassword` char(60) NOT NULL,
14.  PRIMARY KEY (`user_id`),
15.  UNIQUE KEY `email` (`email`),
16.  UNIQUE KEY `username` (`username`)
   ) ;
```

17. Now we will create the table to hold perscription information:

```
18.  CREATE TABLE IF NOT EXISTS `medicine` (
19.  `pill_id` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
20.  `assigned_to` varchar(16) NOT NULL,
21.  `assigned_by` varchar(16) NOT NULL,
22.  `medicine` varchar(30) NOT NULL,
```

```
23.  `date` char(10) NOT NULL,
24.  `time` char(5) NOT NULL,
25.  `num_pills` tinyint(1) NOT NULL,
26.  PRIMARY KEY (`pill_id`)
   ) ;
```

27. This table will hold our appointments information:

```
28.  CREATE TABLE IF NOT EXISTS `appointments` (
29.  `appt_id` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
30.  `patient` varchar(16) NOT NULL,
31.  `doctor` varchar(16) NOT NULL,
32.  `date` char(10) NOT NULL,
33.  `time` char(5) NOT NULL,
34.  PRIMARY KEY (`appt_id`)
   ) ;
```

35. This table will hold the conversations between our users and doctors

```
36.  CREATE TABLE IF NOT EXISTS `chat` (
37.  `chat_id` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
38.  `date` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
39.  `sent_by` varchar(16) NOT NULL,
40.  `sent_to` varchar(16) NOT NULL,
41.  `message` varchar(1000) NOT NULL,
42.  PRIMARY KEY (`chat_id`)
   ) ;
```

43. This table allows us to turn on and off the SMS alerts

```
44.  CREATE TABLE IF NOT EXISTS `alertSwitch` (
45.  `switch` char(1) NOT NULL
   ) ;
```

We will need to manually add a row to this table for it to be functional. To add this row, enter:

```
INSERT INTO alertSwitch (switch) VALUES ('0');
```

Now we will copy over the PHP sources to the webserver. Use an SFTP client like FileZilla to connect to the server and navigate to the directory */var/www/html*

Copy over the source files into this directory. This directory is what Apache uses for their web root by default. This means that you any file will be accessible via HTTP at your server's IP address. If you have the server installed on your local machine, it will be available in your web browser at http://127.0.0.1/.

The SMS alerts are off by default. If you would like to turn them on, run this command in the terminal (SSH):

```
php -f /var/www/html/onoff.php
```

To turn back to the off state, just run the same command again.

Now we will create a symlink for the calendar to read our JSON events

```
ln -s /var/www/html/json /var/www/html/partials/json
```

Lastly, we need to make sure the permissions are set for our files

```
sudo chmod -R 777 /var/www/html
```

Now you have successfully installed our Online Health Monitoring System!

Online Health Monitoring System

Project Legacy

COP 4331, Fall, 2014

Modification history:

Version	Date	Who	Comment
v0.0	8/15/00	G. H. Walton	Template
v1.0	9/1/2014	C. N. McCue	First Revision

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

Roles

Since team formation and project selection took two weeks, it is considered Phase 0. Phase I consists of Deliverables I. Below is the work breakdown of Phase 1.

Deliverables I Work Distribution	**Team member(s)**	**Work Percentage**
Concepts of Operations	Ethan, Chris M.	50%, 50%
Software Requirements Specification	Jon, David	50%, 50%
Project Management Plan	Chris M.	100%
Test Plan	Chris C., James	50%, 50%
Video Chat Research	James	100%
Quality Assurance, Revision and Compilation	Chris M.	100%

Phase 2 is synonymous with Deliverables II. Below is the work breakdown of Phase II.

Deliverables II Work Distribution	**Team member(s)**	**Work Percentage**
High-Level Design	Chris M.	100%
Detailed: Database Manager/Login/Settings	James	100%
Detailed: Alert Manager	David	100%
Detailed: Navigator	Ethan	100%
Detailed: Chat Manager	Jon	100%
Detailed: Pill/Appointment Manager	Chris M.	100%
Detailed: Calendar	Chris C.	100%
Quality Assurance, Revision and Compilation of DII	Chris M.	100%
Presentation Slides	All	100%
Presentation Outline, Revision and Compilation	Chris M.	100%

Although source code is part of Deliverables III, we consider coding as a separate phase. Phase 4 consists of coding. Below is the work breakdown.

Code Work Distribution	**Team member(s)**	**Work Percentage**
Research	All	100%
Database Manager/Login/Settings	James	100%
Alert Manager	David	100%
Navigator	Ethan	100%
Chat Manager	Jon	100%

Pill/Appointment Manager	Chris M.	100%
Calendar	Chris C., Jon	50%, 50%
Database Integration Help	James	100%
Code Integration	All	100%
Unit Testing	All	100%
Integration Testing	All	100%
System Testing	All	100%
Refactoring	All	100%

Phase 3 consists of Deliverables III. For source code, look at the above table. Below is the work breakdown for Deliverables III.

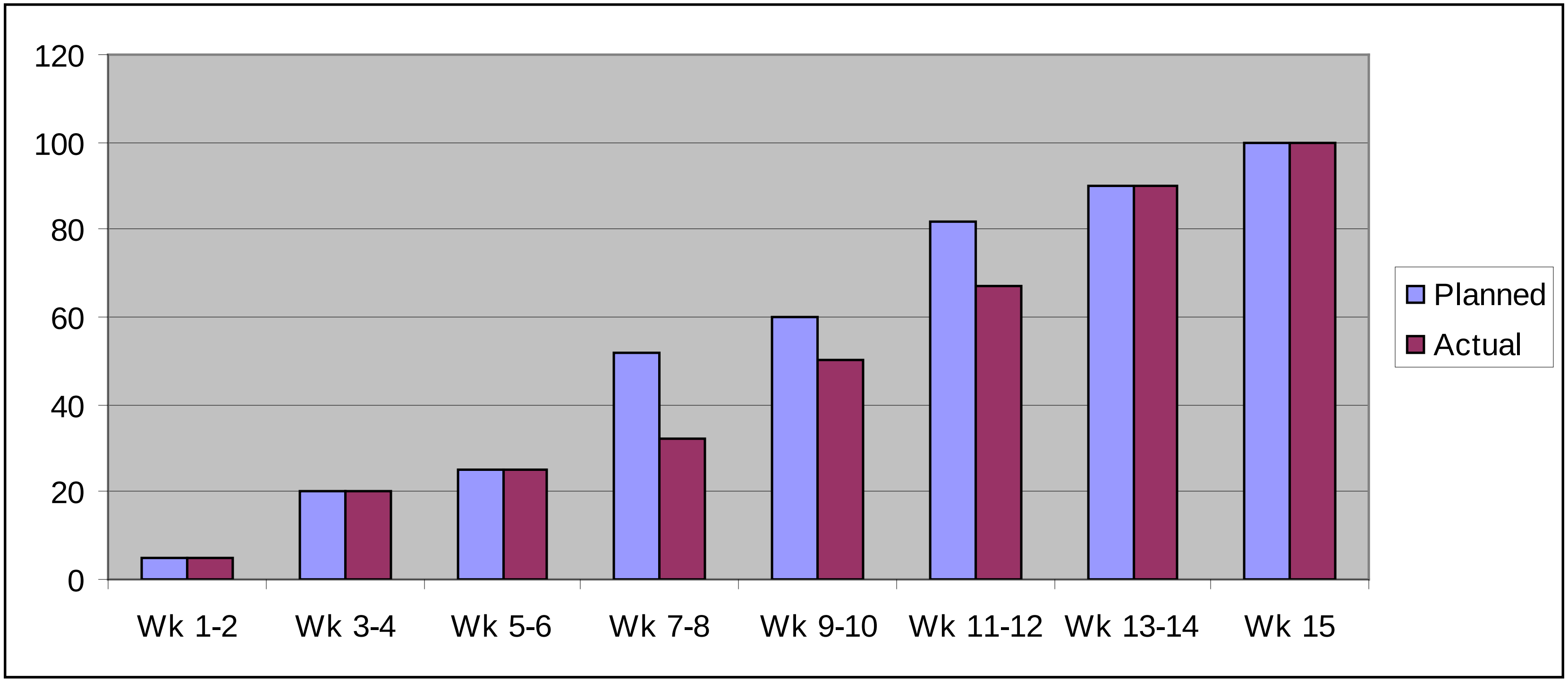
Deliverables III Work Distribution	**Team member(s)**	**Work Percentage**
“User Manual” per each module	All	100%
User Manual Revision and Compilation	Ethan	100%
Build Instructions	James	100%
Project Legacy	Chris M.	100%
Test Results per each module	All	100%
Test Results Revision and Compilation	Chris C.	100%
Quality Assurance, Revision and Compilation of DIII	Chris M.	100%
Presentation Modules	All	100%
Presentation Revision and Compilation	Chris M.	100%

Below are all the routine administrative tasks.

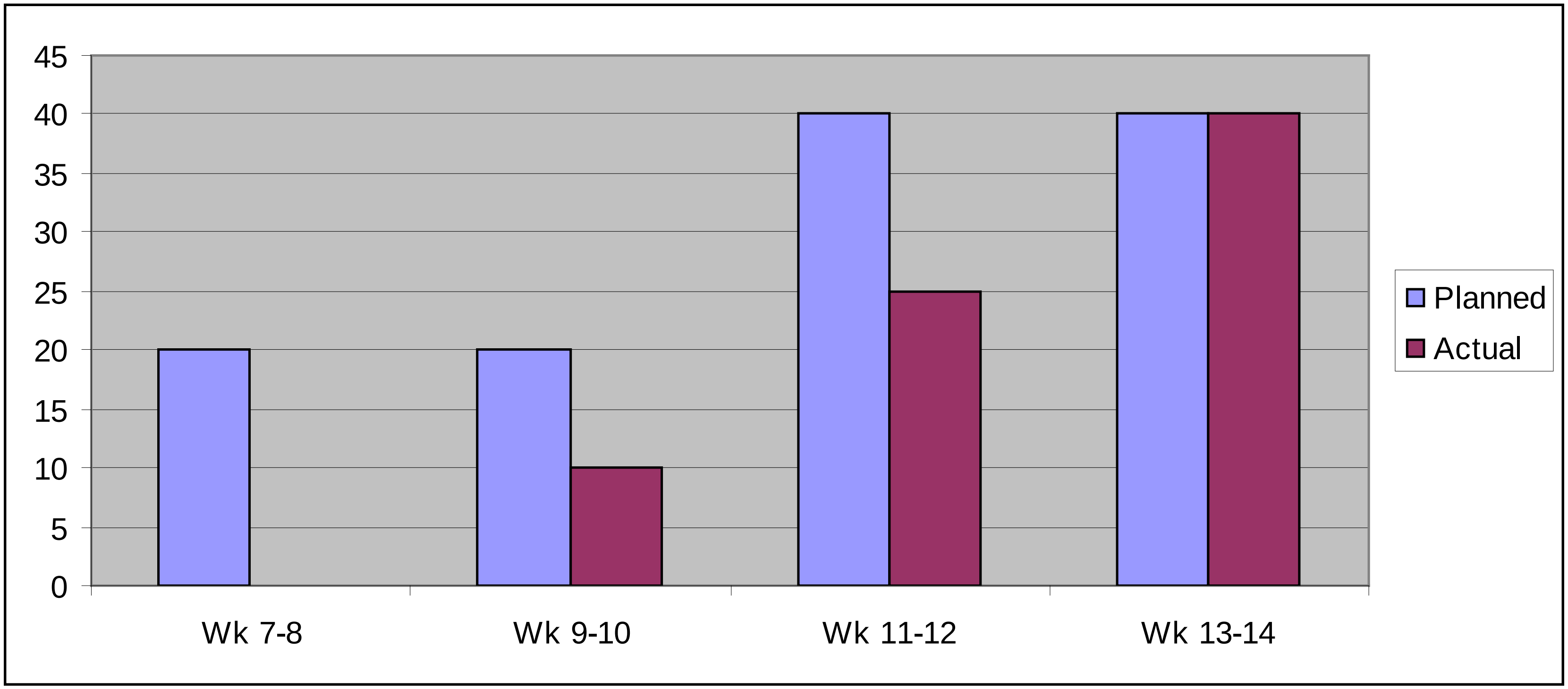
Routine Task Work Distribution	**Team member(s)**	**Work Percentage**
Analysis, Bi-weekly Project Reports and Log Tally	Chris M.	100%
Meeting Agenda and Minutes	Chris M.	100%
Website Administration	Jon, Chris M.	60%, 40%
Individual Websites	All	100%
Liaison	Chris C., David	100%, 100%
Quality Assurance	Chris M.	100%
Configuration Management	All	100%

Analysis

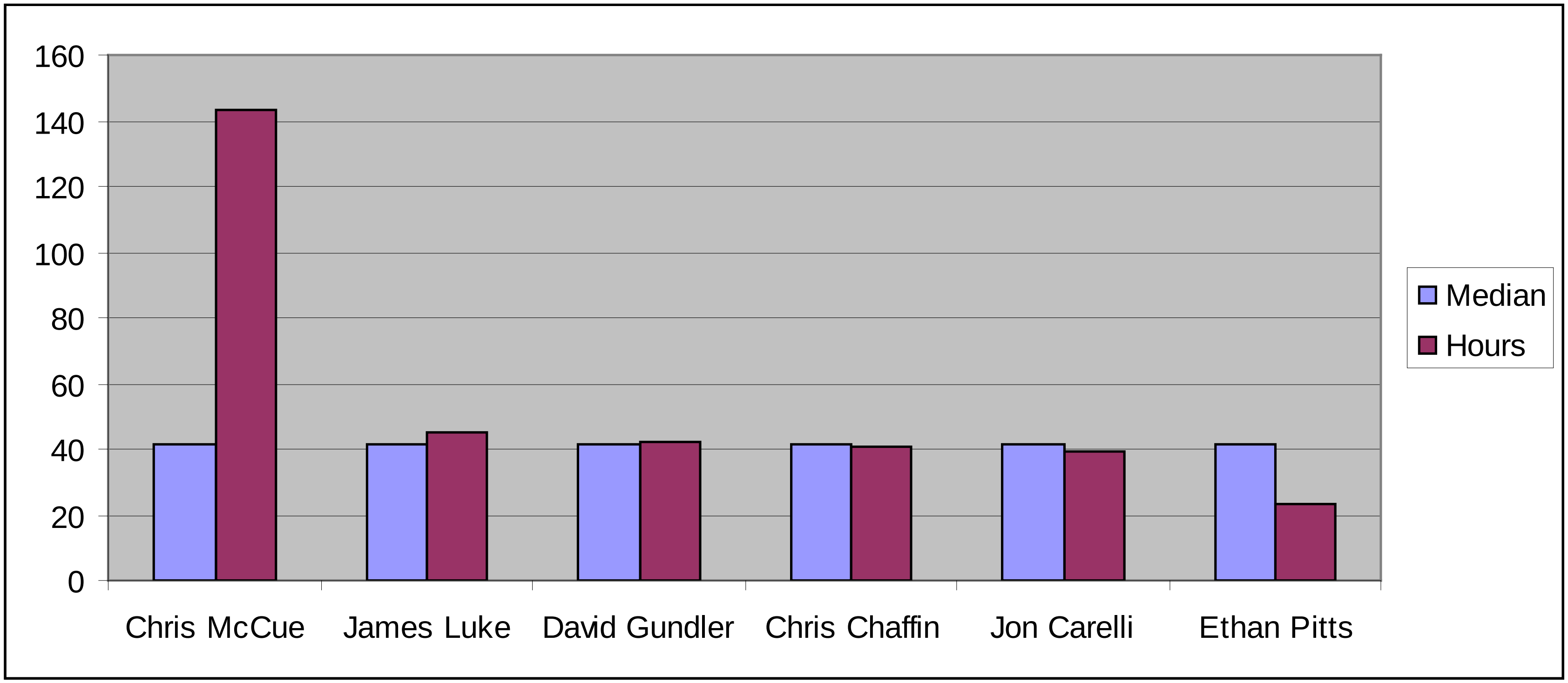
The chart below compares our overall progress metrics from the bi-weekly reports with the expected overall metrics. As you can see, the progress was fairly steady. Except for weeks 7-12, we were on schedule throughout the project. Our progress was behind for weeks 7-12 due to coding. During those weeks, our documentation was on schedule.



The chart below takes a closer look at coding for weeks 7-14. This was the only major issue we encountered. Though we were behind, we rapidly caught up by week 14.



The chart below shows the hours worked per team member. Most team members were close to the median of 41.25 hours. Chris McCue was extremely overworked, while Ethan was underworked.



- Assessment of the Quality of the Final Product:

  The Online Health Monitoring System functions according to the “must have” requirements. The pill/appointment manager adds pill schedules and appointments for patients. It is robust. Pill schedule error checking consists of: no pill name entered, invalid dates, end time is less than start time and duplicate pill names. Appointment error checking consists of: invalid dates and duplicate appointments. The Calendar is updated with pill schedules and appointments. The Alert Manager sends pill schedules and appointment SMS alerts. The Chat Manager allows for doctor-patient and patient-doctor communication. The project works well. Everything is as expected. One can login to the website specified on the User’s Manual and use all the functionality.

- Recommended Use of the Final Product:

The project meets the specified requirements. However, it would need modification for use in the real world. Currently, a patient can have one doctor, according to our requirements. More security would be required to make sure no one fakes being a doctor. These are examples of minor changes that would need to be implemented for real world use. The risk of lawsuits with medical applications is so great that we would hesitate to use this in the real world. However, in the university world, this is delivered as requested.

- Known Problems:

All work was finished to meet project requirements. There are only two minor known problems. The calendar of newly registered users shows the most recently logged in patient's pill and appointment schedule until the new user gets adds a pill or appointment scheduled. Potential fixes include: New users automatically getting a dummy pill assigned to them once they register. Change algorithm to only display pills/appointments if a value is returned from database. If not, display nothing. Also, the SMS spam filter is sensitive. When the SMS code is turned off, that day's information will be lost.

- Adherence to Project Plan:

Overall, the team was able to adhere to the project plan very well. Administrative and technical tasks were divided equally among team members. Communication was maintained weekly with individual logs, emails and meetings before Deliverables. Corrective actions were taken as needed. For example, corrective actions were taken with the Test Plan. During revisions of deliverables, there was quality assurance to make sure standards were met on documentation. Risks were minimized by managing risks early on.

Overall, the team followed project time estimates well. Weekly logs included progress metrics. This made it easy to assess weekly progress. Occasionally, factors, such as homework in the Processes class or tests in other classes, may have delayed items a day or two. However, a cushion was built into time estimates to easily allow for this. Therefore, we were able to accommodate these minor deviations. Much of the time was spent just keeping up with the high priority items, such as next deliverable, presentation or meeting deadline. Therefore, coding had to be delayed a few weeks because of Deliverables II. It was suppose to be in parallel with Deliverables II. This was our only major deviation.

Because of this, we were not able to adhere to our planned software life cycle process well. We were not able to spend much time in prototyping stage of incremental phased development. We had to go straight for integration. Our process more closely resembled agile development.

The root cause of deviations from time estimates was the daunting task of documentation. The only major time deviation was coding. This was delayed because of the effort needed for Deliverables II. Since source code was not due until Deliverables III, it was lower priority. Besides this, there were only minor deviations of a day or two because of work load outside the project. Lower priority items did have a time lag. Outside of analysis of individual logs to assess progress and take corrective action at crucial points of the project, the Project Manager found the weekly analysis and bi-weekly project report to be a cumbersome, time-consuming activity of little worth. The Project Manager had a good sense of project time management without these reports and found them superfluous. Therefore, these low priority items often had to wait until all higher priority items were completed.

This delayed reports for weeks. Configuration management also lagged. It took a while to post documents on the website after their completion. This was considered a lower priority item to be done when time permitted.

- Defect Analysis:

  There were a total of 44 defects. One defect was introduced, discovered and corrected during the code prototyping/unit testing phase. This was an email error. There were 38 defects that were introduced, discovered and corrected during the code integration/integration testing phase. Most of these errors occurred because of Database or Navigator integration. Until our code was integrated with the database, we could not really test our modules. That is why so many errors were discovered in the integration testing phase. There were 5 defects that were introduced during the code integration stage, but discovered by other team members and corrected during the system testing stage. These errors consisted of David finding errors with the Pill/Appointment Manager and Chris finding errors with the Calendar. In summary, most of our faults were caught and corrected in the same phase of the project. Also, most of our faults were due to integration. This was the most challenging stage of the project.

- Quality Assurance:

  In the end, quality assurance was performed by the Project Manager during deliverable revision. This was sufficient to produce documentation in line with team quality standards. This was done in a timely manner before submission of deliverables. However, it was a burden for the Project Manager. This could have been improved if there were a person dedicated to quality assurance. However, this person would have to have a good sense of documentation. This is a rare quality to find among computer scientists and engineers.

  The testing activities for our code were sufficient. Testing activities were performed in parallel with coding. Each module coder tested his code as he wrote it. Integration revolved around the Database Manager and Navigator. Integration testing was done by the module coder as we first integrated with the Database Manager and, then, with the Navigator. The tests were as timely as they could be, given that we were rushing to finish code before the demo. System testing occurred organically as we were putting the finishing touches on our modules. Given the time constraints of this project, our testing was optimal. If we had more time for this project, we would have each team member independent inspect the whole system for faults.

- Configuration Management:

  Every team member was responsible for configuration management of his documentation or code. The team website was used as a hub for all documents. The author retained a copy as well. Overall, these were adequate for our small project in the short project life cycle. However, there was always a lag in the posting of documents on the website since this was considered a lower priority. There was only a minor problem with configuration management of the SRS document when three of us were modifying it at the same time. We were able to revise the SRS properly. For a larger project, our configuration management would need improvement. We could have improved configuration management by using stricter guidelines, such as a document change form, and/or having a dedicated person for configuration management.

For source code, we uploaded our source code to the server during development. For final source code, we used Dropbox as a repository. We never experienced any configuration management problems with our code.

- Suggestions for the Future:

Given the current team dynamics, the author feels that the technical processes put into practice were optimal. Time management and communications were handled well. However, these practices did cause overload for the author. This would be an area for improvement. No advice can offered as to how this would be implemented as the author has not yet found a way to do so.

For future project teams, advice would be as follows:
Start early with team formation and project selection.
Set modest and realistic expectations for your project.
Set deadlines with the expectation of procrastination being the norm. In other words, make early deadlines before real deadlines.
Make quality assurance and team expectations clear at the beginning.
Meet and communicate frequently (at least weekly through individual log).
Follow-up on progress through progress metrics in individual logs and remind frequently of upcoming action items.

If assigned a project 10 times the scope of this project, work distribution would need to be more specific and guidelines would need to be more specific. Tasks, such as configuration management, quality assurance and web maintenance, would need to be assigned to specific individuals. There would need to be meetings just to discuss guidelines and expectations, such as those in the Project Management Plan. Much more time would be needed to carefully articulate Deliverables I. The software life cycle process would be followed more strictly. In summary, more time would be needed before the design phase.

For a project 100 times the scope of this one, even the minutest tasks would require dedicated personnel. For example, there would be an individual dedicated to the project website. The team structure would need to be more hierarchical. There would be teams for design, teams for coding and teams for administrative tasks. A programmer may only have a vague idea of the big picture as tasks are subdivided again and again. For example, a programmer may work on a few classes of a module. The coder would never work on anything from Deliverables I. Deliverables III would become much more important as the code would need to be maintained. Code maintenance would be an important item to consider.

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

***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