UAH Fit Vault Documentation

Release 1.0

James Duggan, Glen Riden, Ryan Wilkins

April 26, 2016

CONTENTS

Contents:

CHAPTER

ONE

USER GUIDE

Table of Contents

1.1 Overview

Here you can find all the documentation you will need to know how to do all the basic tasks that can be performed through the UAH Bit Vault user interface.

1.2 How-To

1.2.1 Account Management

Account management is a part of the system. Accounts may be created, deleted or edited. Follow the below TOC tree to find details on each action for each account.

Contents:

Account Creation

Table of Contents

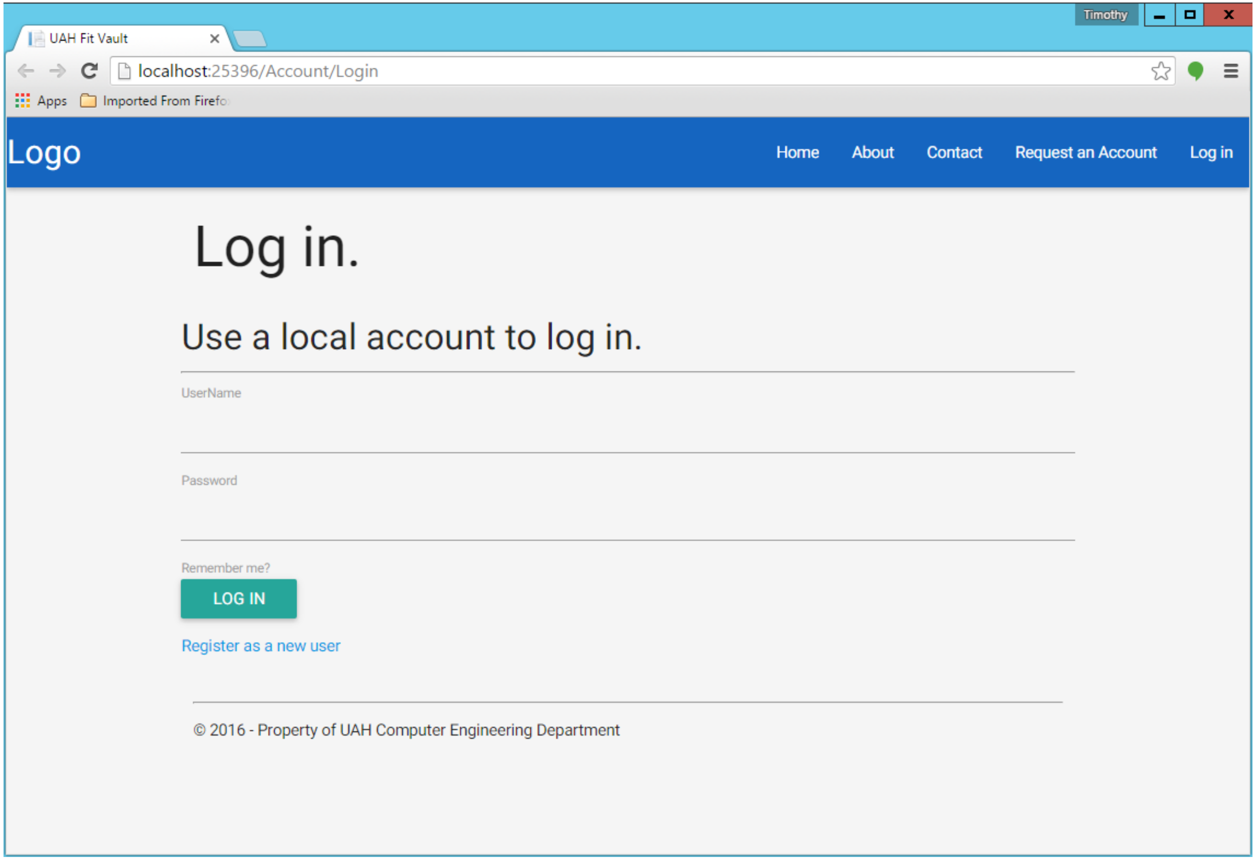
There are 4 types of accounts

- patient
- physician

- experiment admin
- system admin

Physician Account

Physician accounts can be requested at the home/login page. Click on the “Register as a new user” button.



Log in.

Use a local account to log in.

UserName

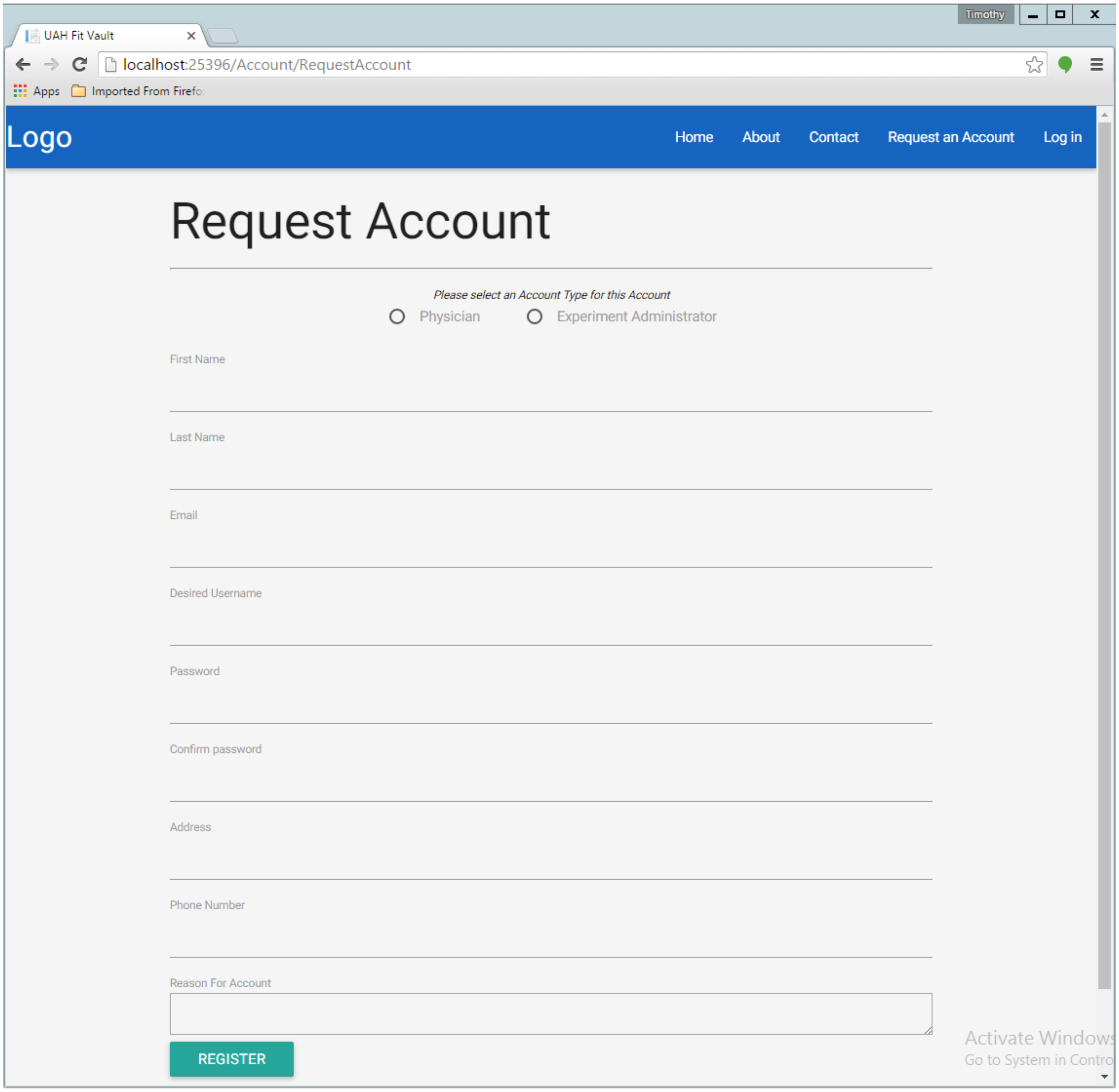
Password

Remember me?

Register as a new user

© 2016 - Property of UAH Computer Engineering Department

This will redirect you to a form to fill out for you new physician account.



Request Account

Please select an Account Type for this Account

○ Physician ○ Experiment Administrator

First Name

Last Name

Email

Desired Username

Password

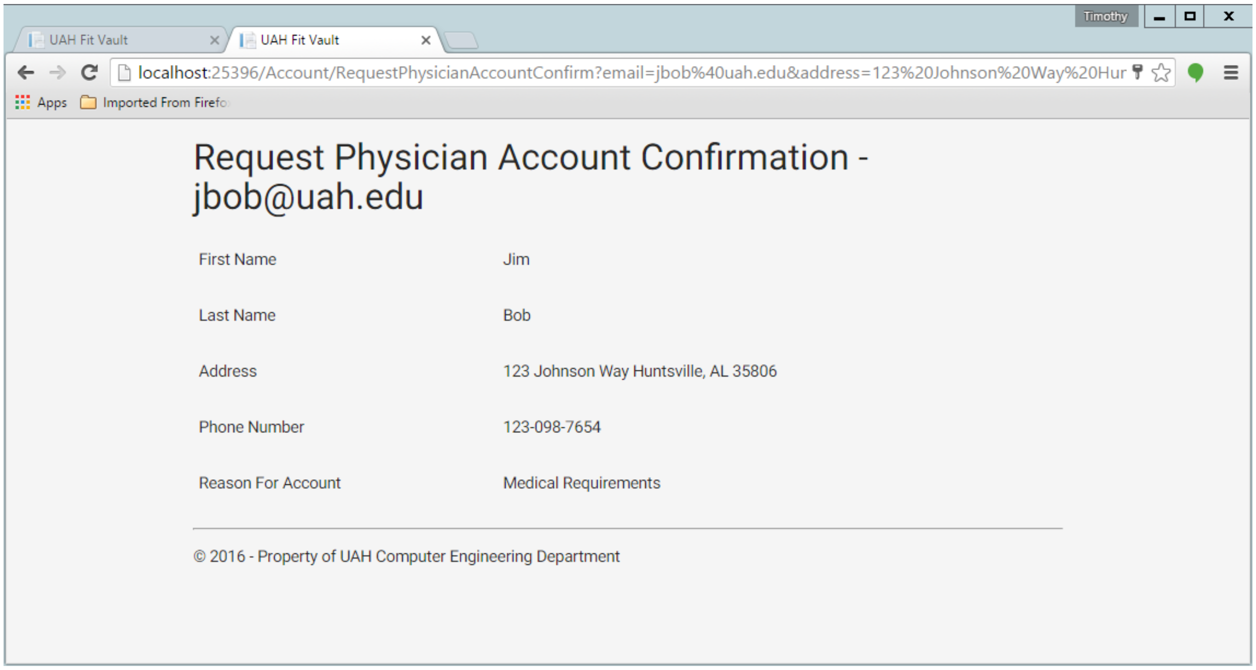
Confirm password

Address

Phone Number

Reason For Account

Once you have filled this out with the appropriate information and selected the "Physician" radio button at the top, you are ready to submit your request. When you click submit you should be brought to a confirmation page.


Timothy
UAH Fit Vault
UAH Fit Vault
localhost:25396/Account/RequestPhysicianAccountConfirm?email=jbob%40uah.edu&address=123%20Johnson%20Way%20Hur
Apps
Imported From Firefo

Request Physician Account Confirmation - jbob@uah.edu

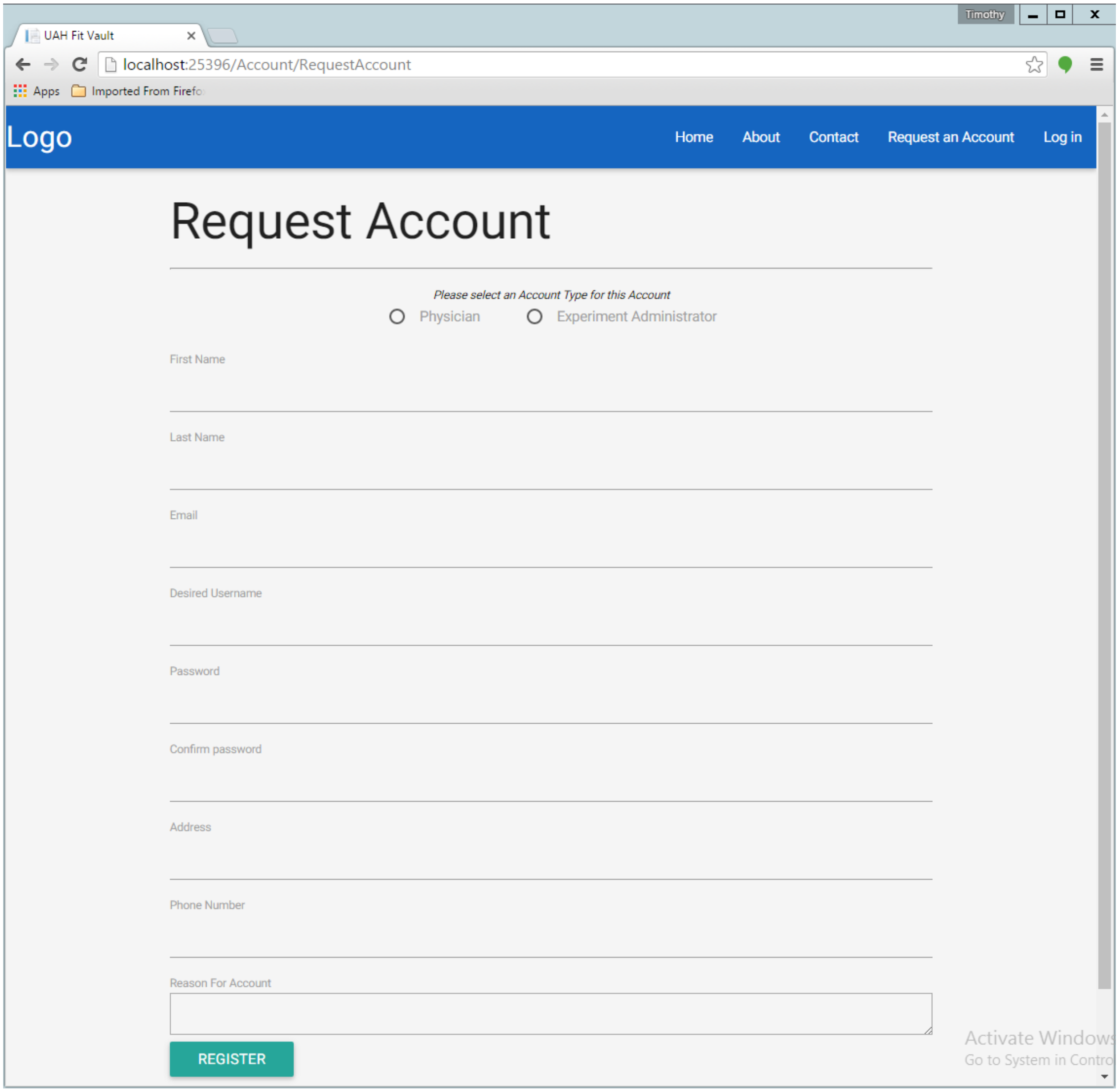
First Name	Jim
Last Name	Bob
Address	123 Johnson Way Huntsville, AL 35806
Phone Number	123-098-7654
Reason For Account	Medical Requirements

© 2016 - Property of UAH Computer Engineering Department

You will not be able to login though until a system admin has approved your account.

Experiment Admin Account

Experiment Admin accounts are created the same way as the physician accounts. Just make sure to select the “Experiment Administrator” radio button at the top of the request account page instead of the “Physician” radio button.



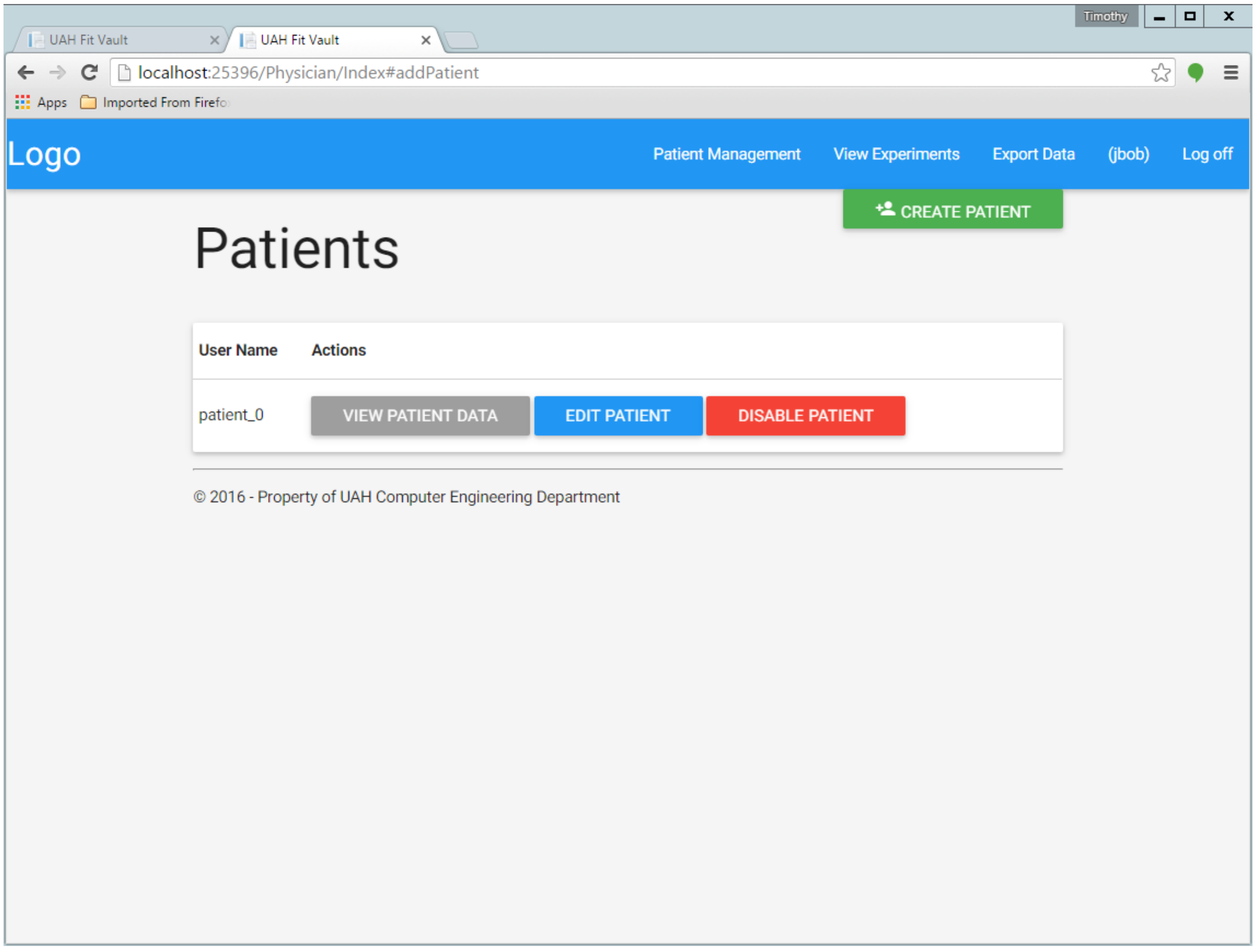
Request Account

Please select an Account Type for this Account

○ Physician ○ Experiment Administrator

Patient Account

Patient accounts can only be created by their physician. From the physician account’s home page click the “CREATE PATIENT” button at the top right of the screen.

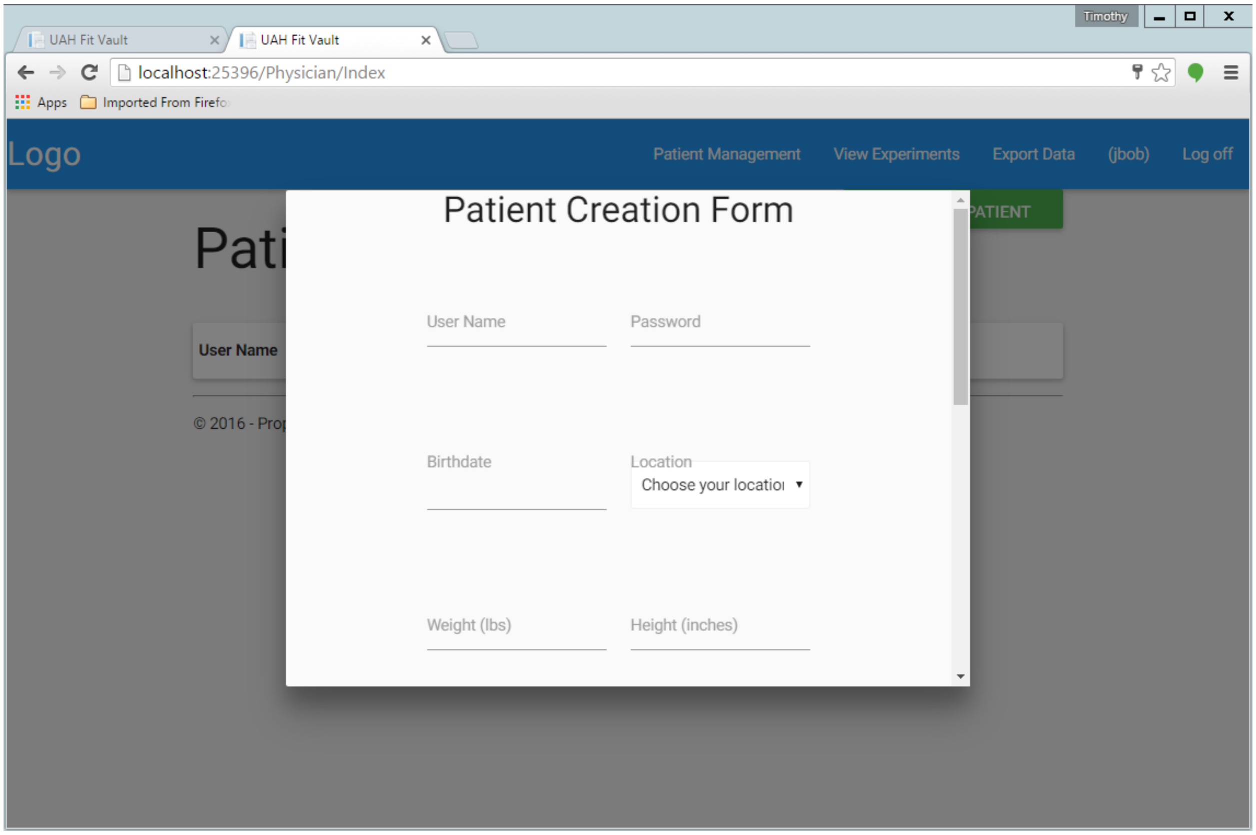


Patients

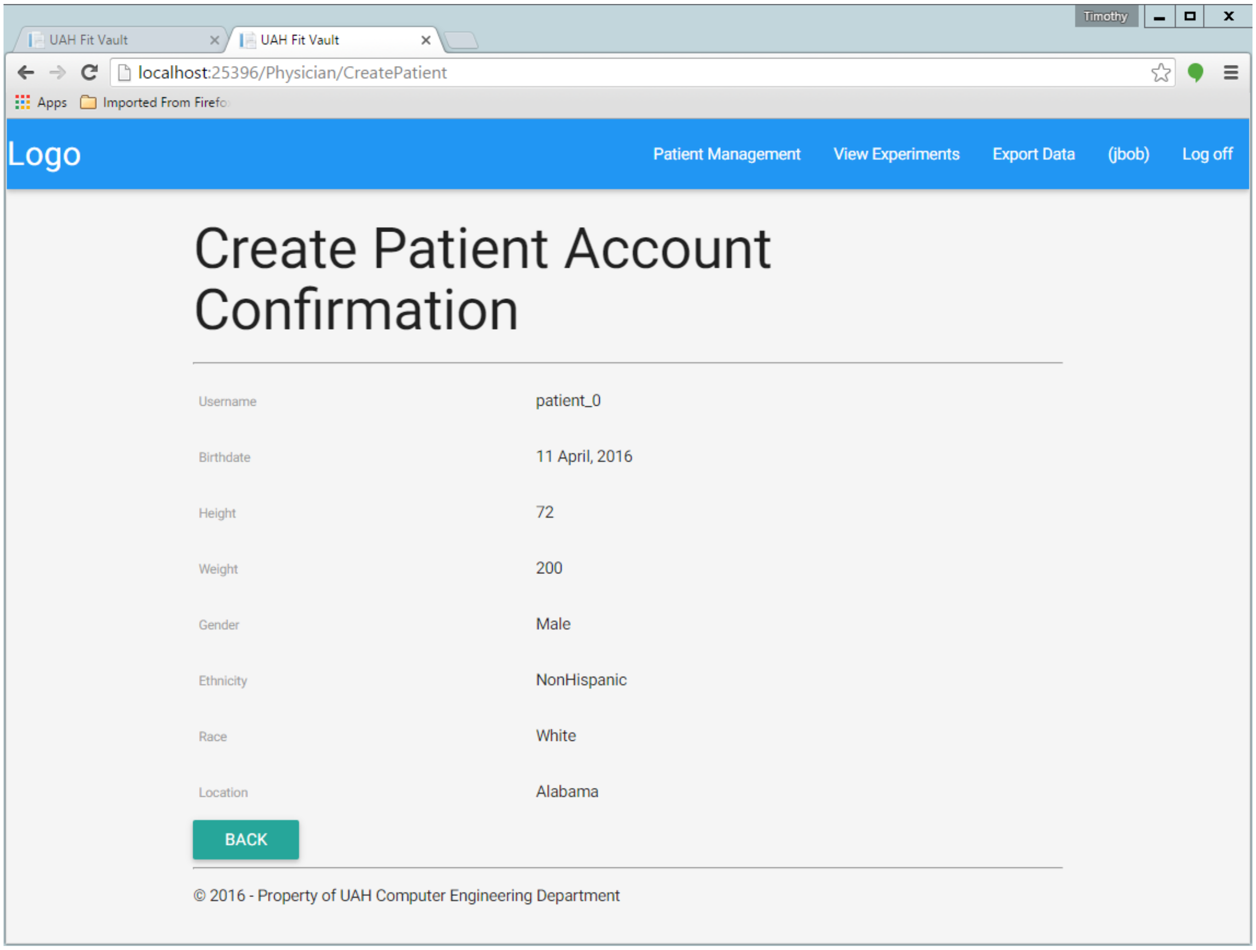
User Name	Actions
patient_0	VIEW PATIENT DATA EDIT PATIENT DISABLE PATIENT

© 2016 - Property of UAH Computer Engineering Department

A popup form should come up with all the information needed to create a patient account.



After all the information has been filled out, you can click the submit button. If everthing worked, you should get a confirmation page.



Create Patient Account Confirmation

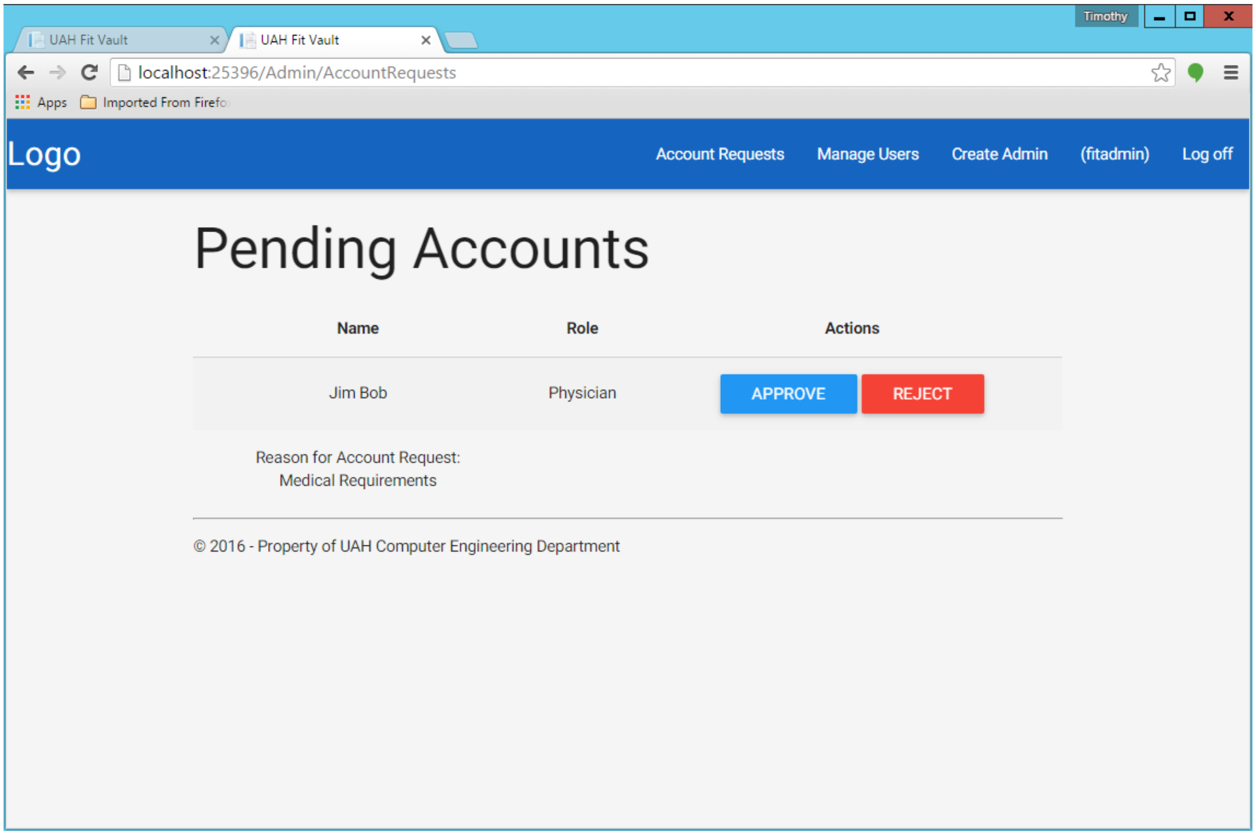
Username	patient_0
Birthdate	11 April, 2016
Height	72
Weight	200
Gender	Male
Ethnicity	NonHispanic
Race	White
Location	Alabama

© 2016 - Property of UAH Computer Engineering Department

System Admin Account

UAHFitVault comes with a system administrator pre-configured. The username is ‘fitadmin’ and the password is ‘Password1!’. We recommend changing this after the installation before the server is made publicly available.

New System Admin accounts can be created by other System Admin accounts. From the System Admin account home page, click the “Create Admin” button at the top right corner of the screen.



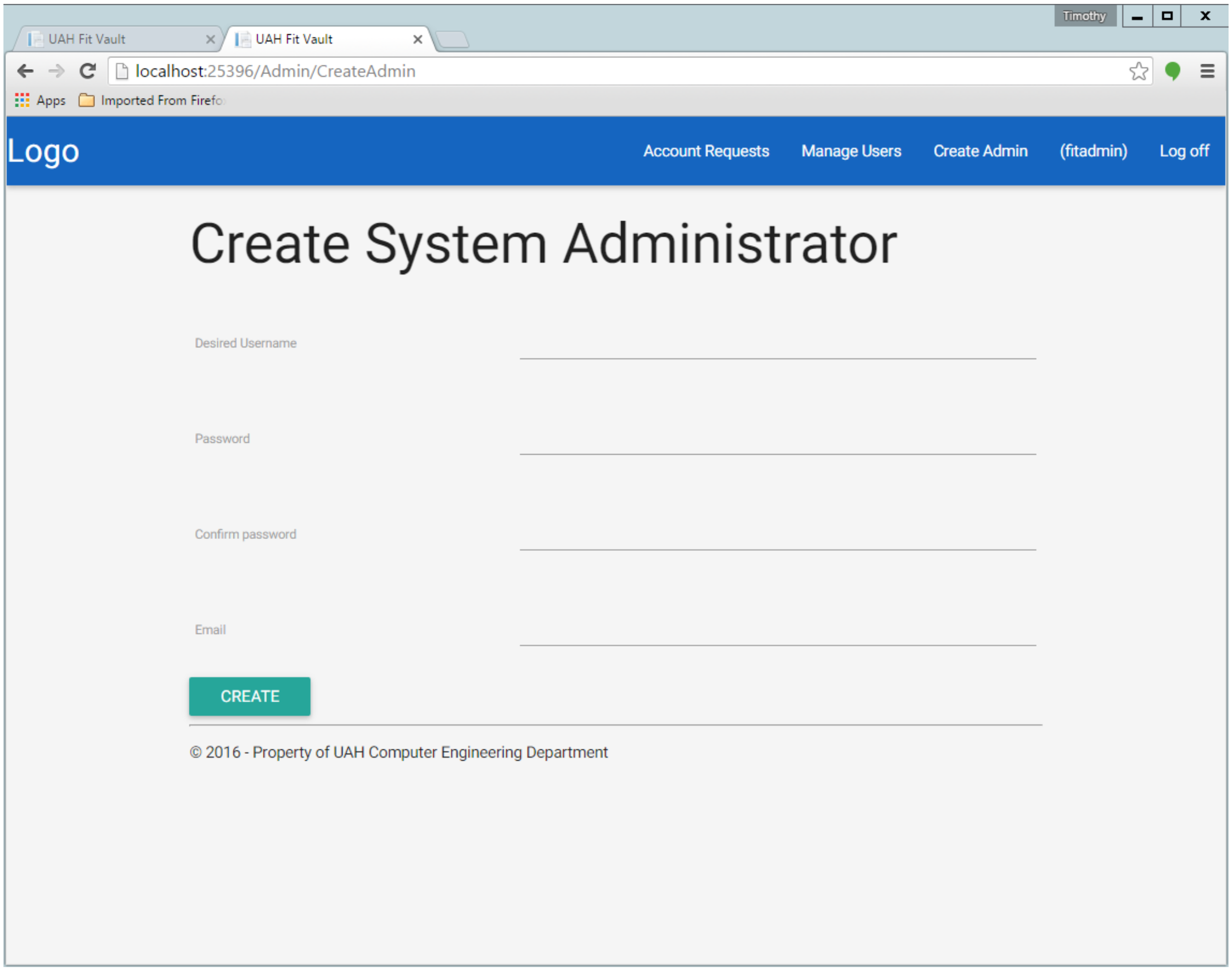
Pending Accounts

Name	Role	Actions
Jim Bob	Physician	APPROVE REJECT

Reason for Account Request:
Medical Requirements

© 2016 - Property of UAH Computer Engineering Department

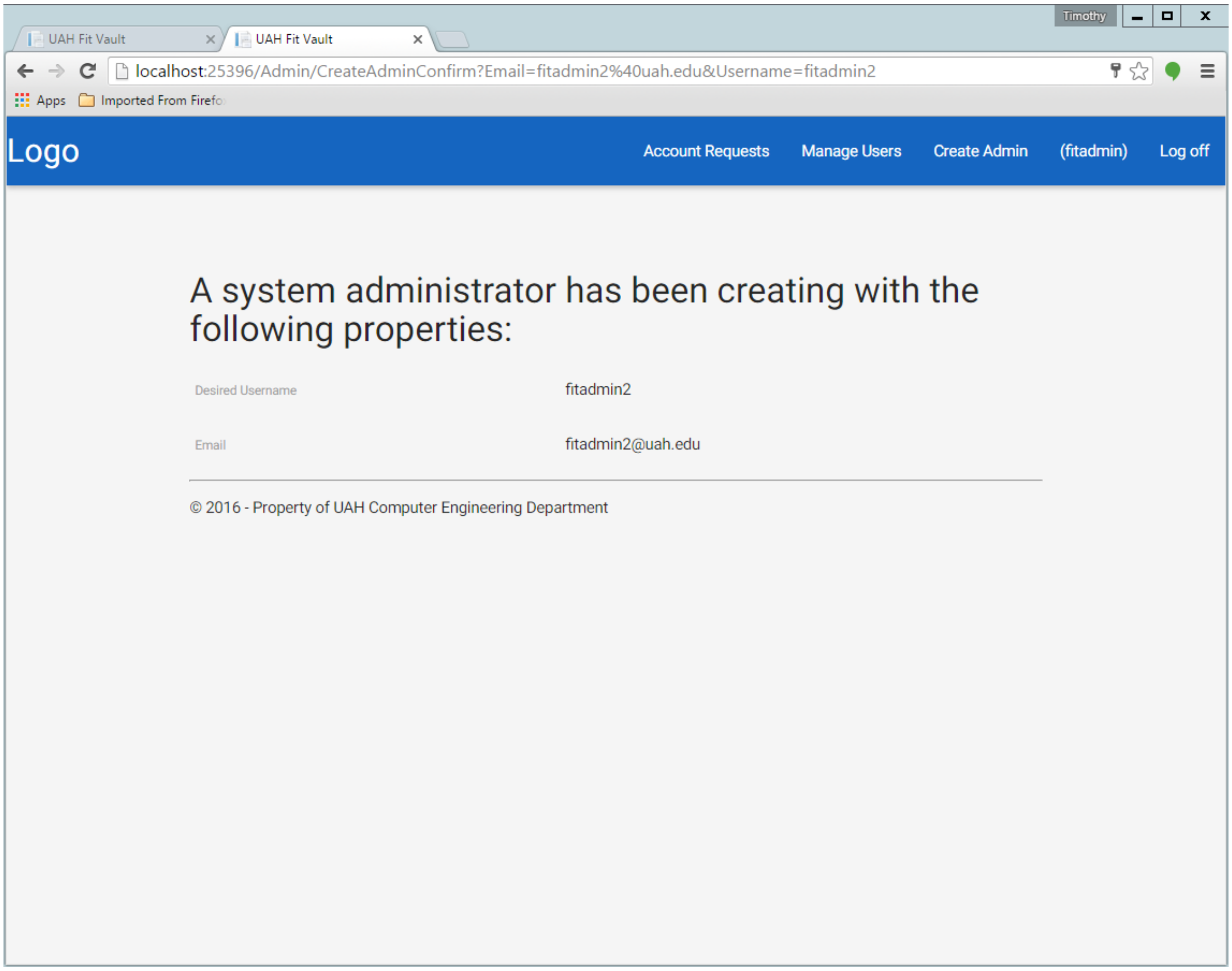
You will be presented with a form to fill out for the system admin account information.



Create System Administrator

© 2016 - Property of UAH Computer Engineering Department

Once the form is filled out and you click the create button, you should get a confirmation page.



A system administrator has been creating with the following properties:

Desired Username fitadmin2

Email fitadmin2@uah.edu

© 2016 - Property of UAH Computer Engineering Department

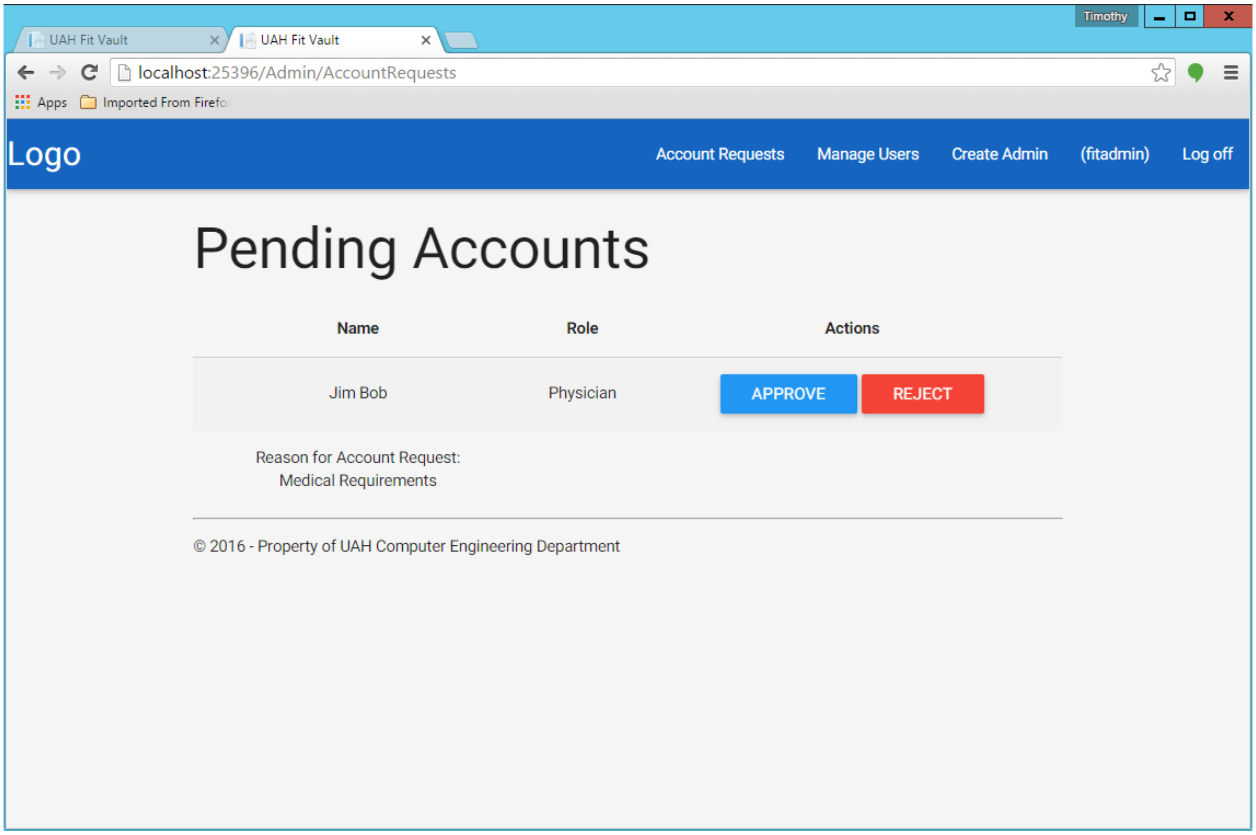
Account Approval

Table of Contents

There are 2 types of accounts that need approval:

- physician
- experiment admin

To approve an account, login with your system administrator credentials. Then, click on the “Account Requests” button at the top right of the page. This should bring you to a page that looks like this:



Pending Accounts

Name	Role	Actions
Jim Bob	Physician	APPROVE REJECT

Reason for Account Request:
Medical Requirements

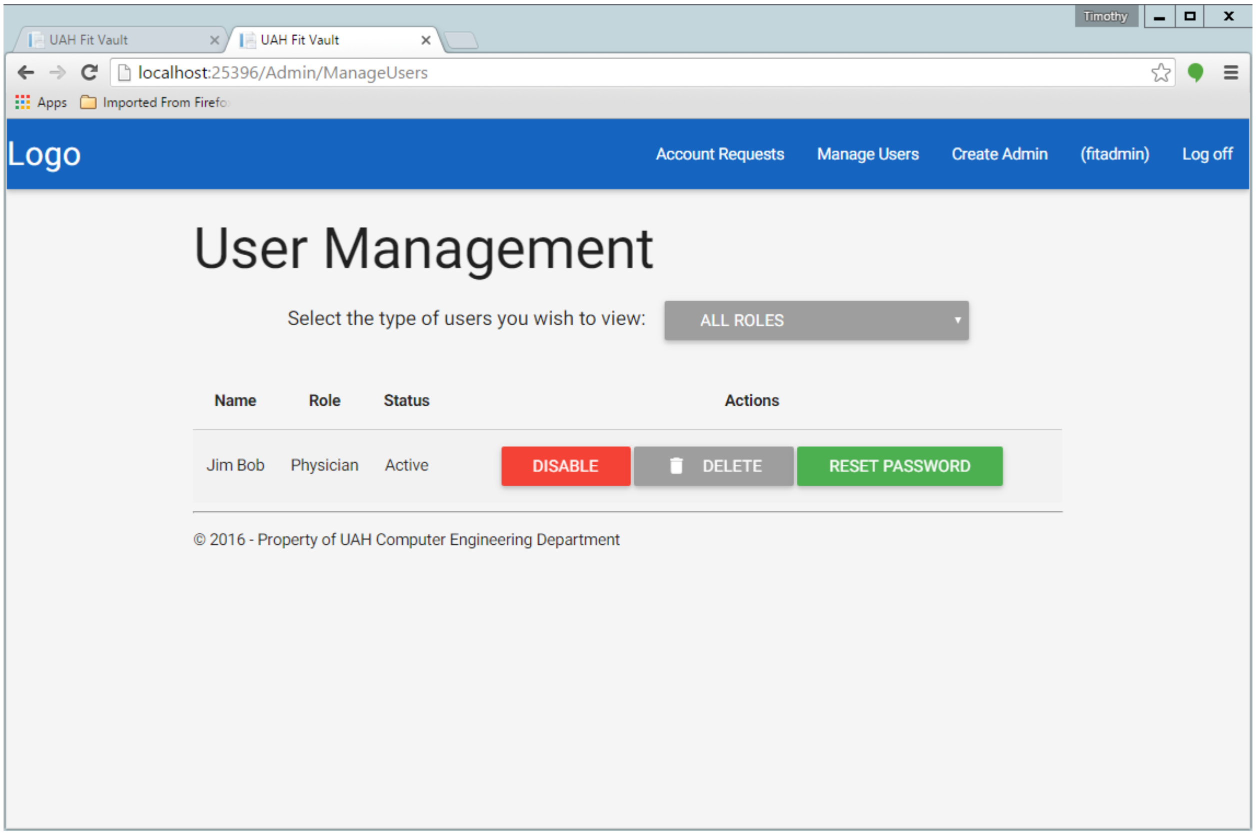
© 2016 - Property of UAH Computer Engineering Department

You can then click on the “APPROVE” or “REJECT” button for the corresponding account you wish to approve or reject.

Account Deletion

Table of Contents

Accounts can only be deleted by the System Administrator. Login as the system admin and click the “Manage Users” button at the top right corner of the screen. You should be taken to a page like the one below:



User Management

Select the type of users you wish to view: ALL ROLES

Name	Role	Status	Actions
Jim Bob	Physician	Active	DISABLE DELETE RESET PASSWORD

© 2016 - Property of UAH Computer Engineering Department

From here you can use the roles filter to help you find the account to delete. Just click the “Delete” button that corresponds to the account you wish to delete. If the delete worked the account should disappear from the list.

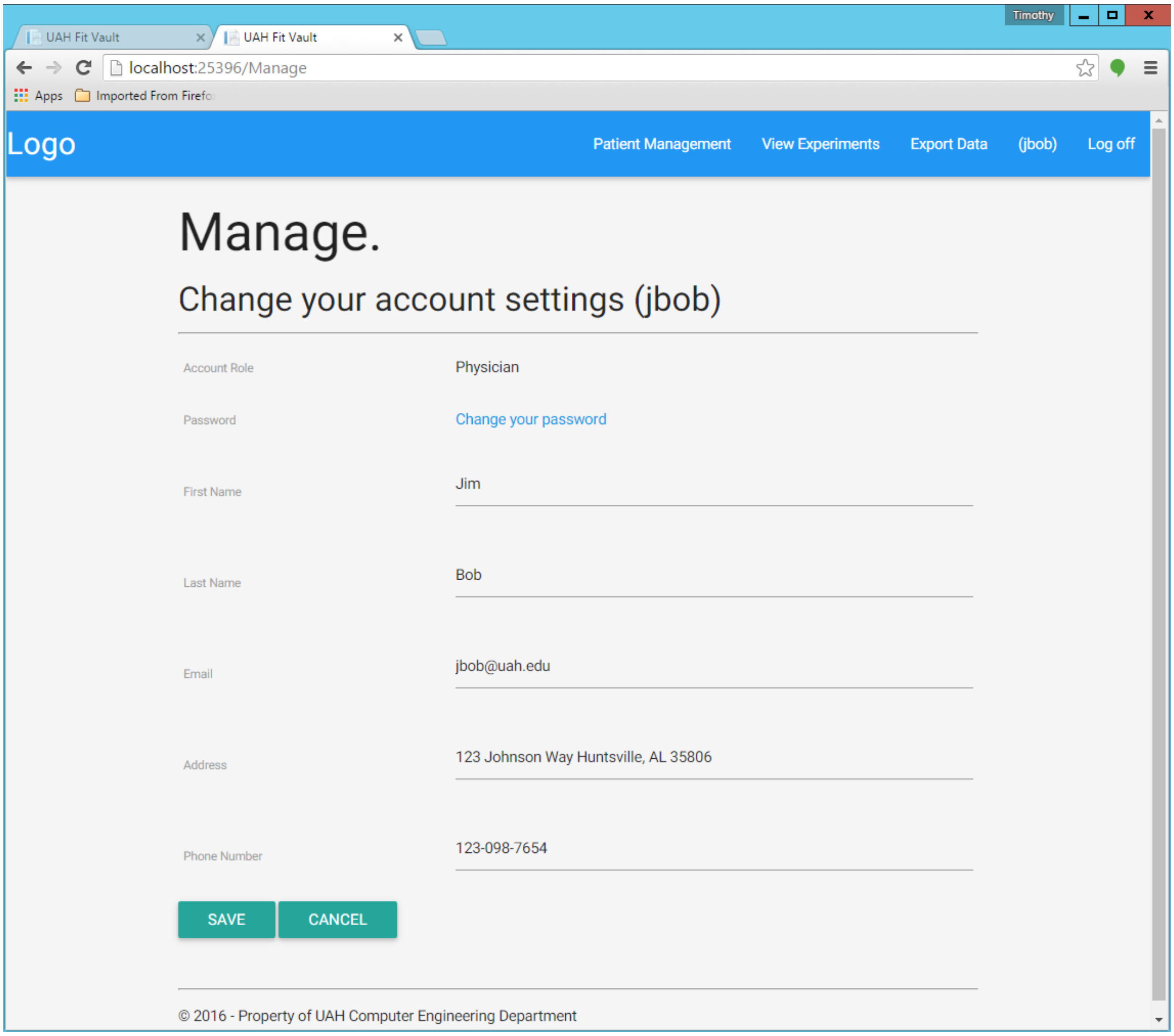
Account Editing

Table of Contents

Users have the ability to edit their account information.

Physician Account

To edit a physician account, you must be logged in as the physician. After login, click on the username at the top right corner of the screen. You should be taken to a page like below:



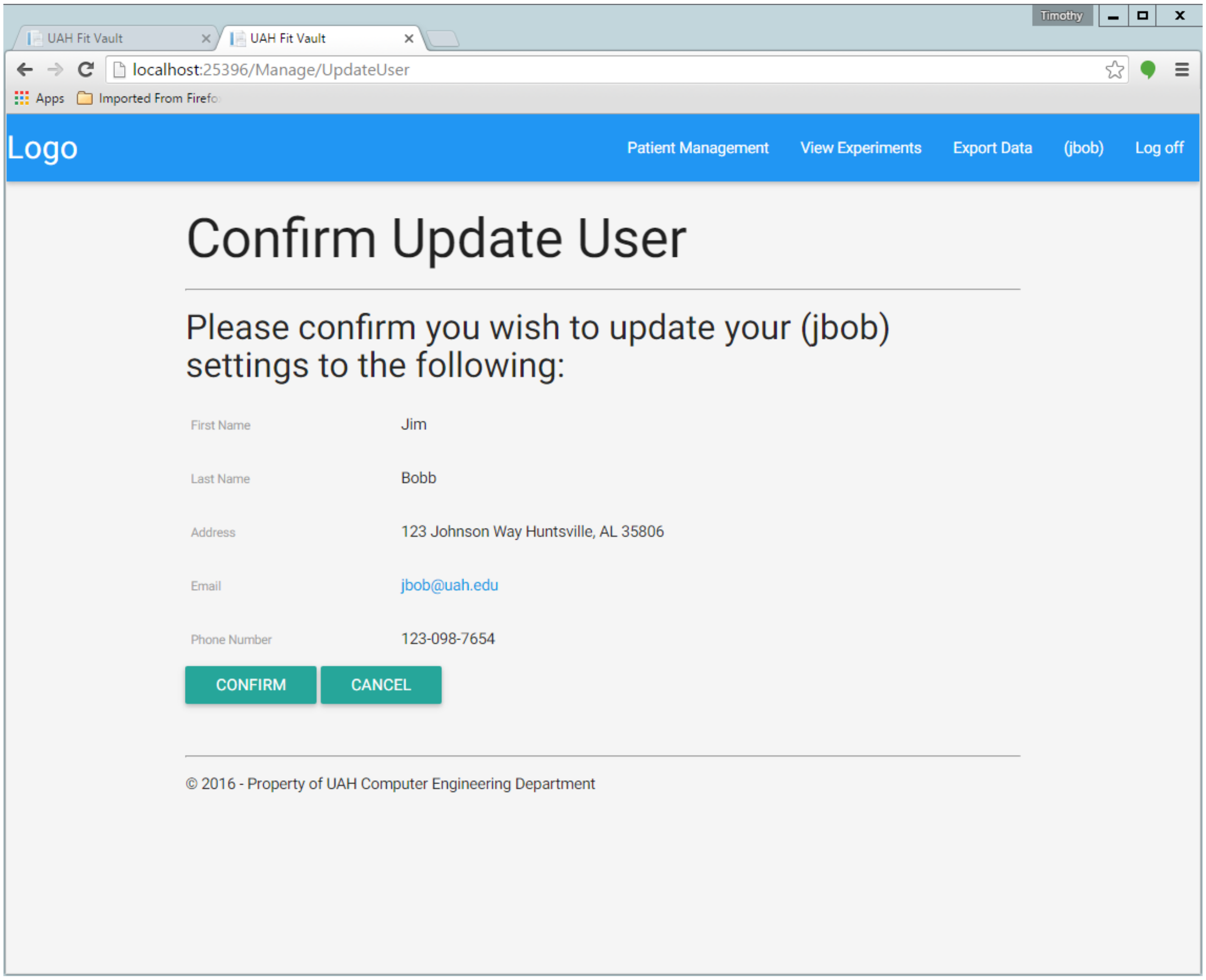
Manage.

Change your account settings (jbob)

Account Role	Physician
Password	Change your password
First Name	Jim
Last Name	Bob
Email	jbob@uah.edu
Address	123 Johnson Way Huntsville, AL 35806
Phone Number	123-098-7654

© 2016 - Property of UAH Computer Engineering Department

After you edit the information you wish to edit, click the “save” button at the bottom. You should be taken to a confirmation page like this:



Confirm Update User

Please confirm you wish to update your (jbob) settings to the following:

First Name	Jim
Last Name	Bobb
Address	123 Johnson Way Huntsville, AL 35806
Email	jbob@uah.edu
Phone Number	123-098-7654

© 2016 - Property of UAH Computer Engineering Department

Click “Confirm” and you changes will be saved.

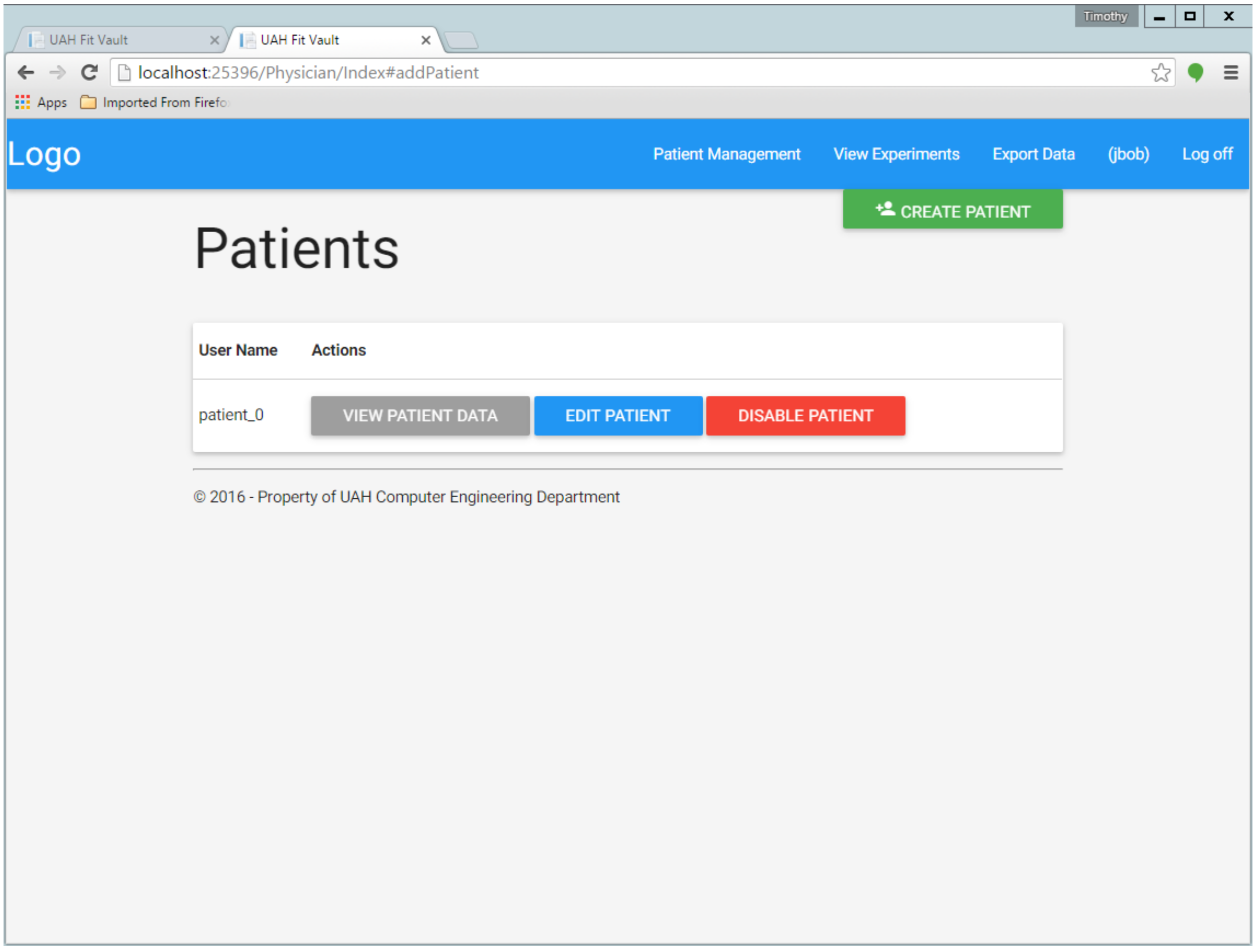
Experiment Admin Account

Editing an Experiment Admin account is exactly the same as editing a *Physician* account.

Patient Account

Patient accounts can be edited by either the patient or the patient’s physician.

Using Physician To edit it using a physician, login using the physician’s credentials then click on the “Patient Management” button at the top right corner of the page. You should then be taken to a page that looks like this:

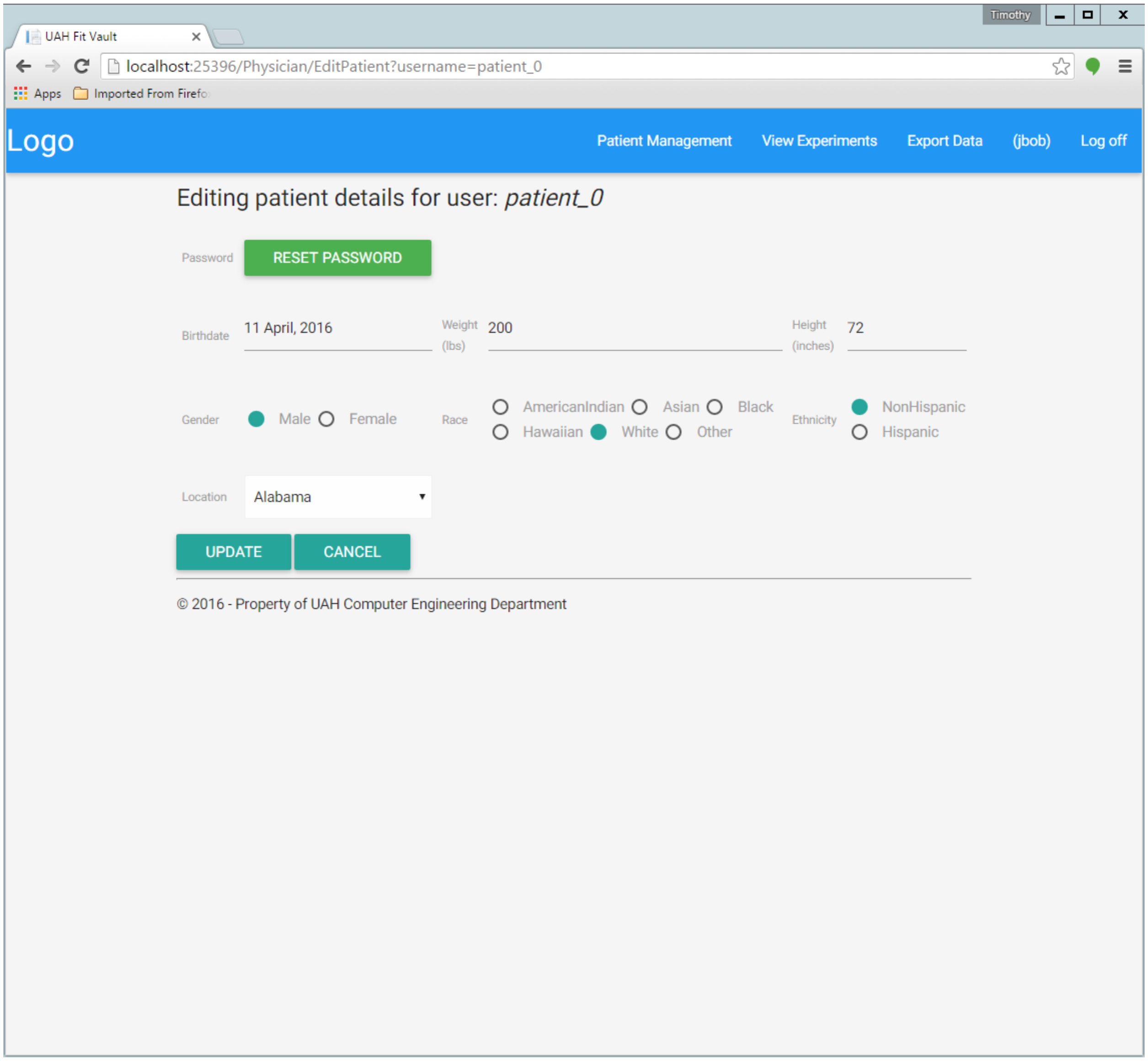


Patients

User Name	Actions
patient_0	VIEW PATIENT DATA EDIT PATIENT DISABLE PATIENT

© 2016 - Property of UAH Computer Engineering Department

Then click on the “EDIT PATIENT” button for the corresponding patient you want to edit. This will bring you to the following page:



localhost:25396/Physician/EditPatient?username=patient_0

Editing patient details for user: *patient_0*

Password

Birthdate: 11 April, 2016

Weight (lbs): 200

Height (inches): 72

Gender: ● Male ○ Female

Race: ○ AmericanIndian ○ Asian ○ Black ○ Hawaiian ● White ○ Other

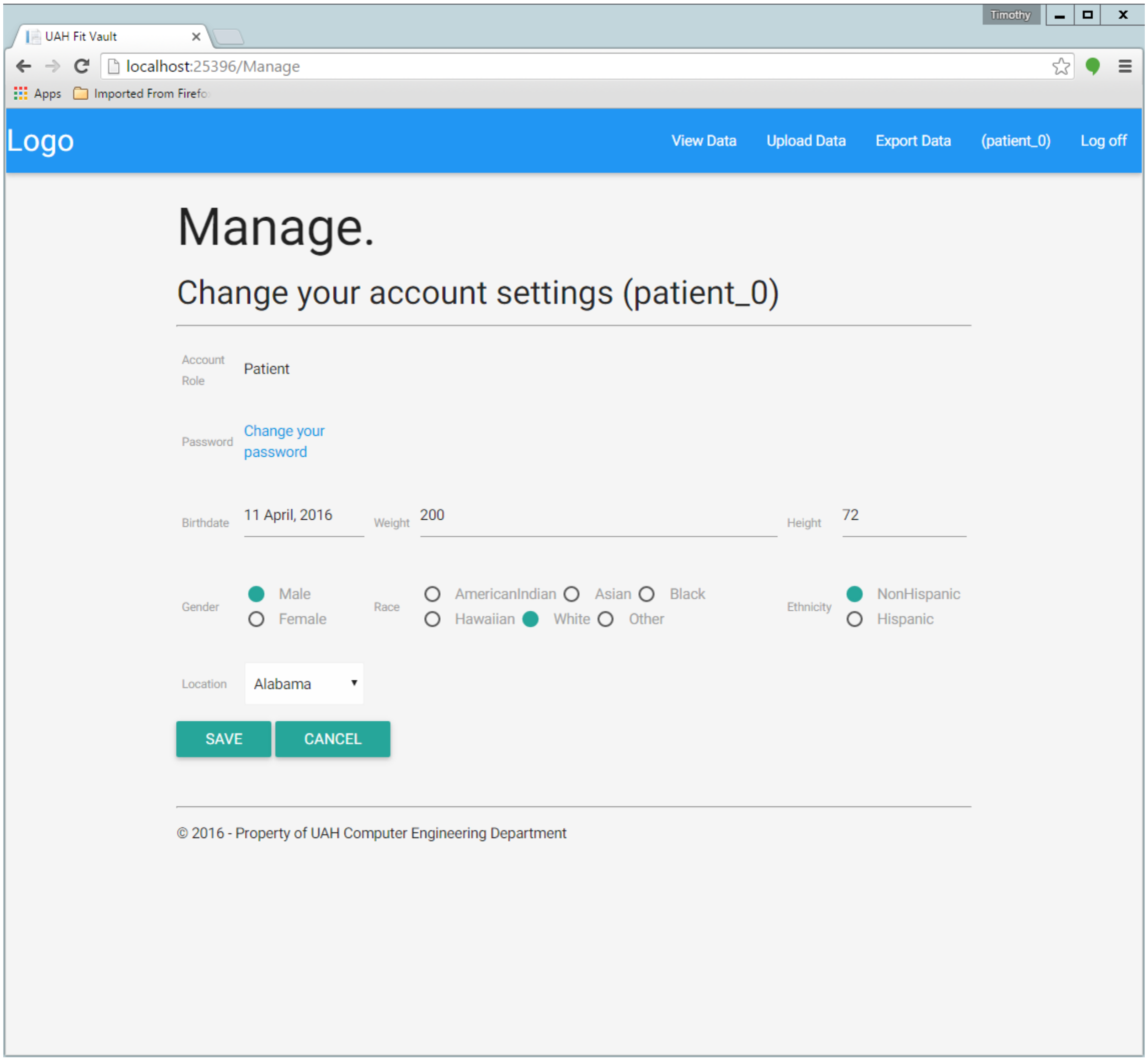
Ethnicity: ● NonHispanic ○ Hispanic

Location: Alabama

© 2016 - Property of UAH Computer Engineering Department

Use the form to edit any of the information you want to change. Then click on the “UPDATE” button.

Using Patient To edit it using a patient, login using the patient’s credentials then click on your patient user name at the top right corner of the page. You should be brought to a page like this:



Manage.

Change your account settings (patient_0)

Account Role: Patient

Password: Change your password

Birthdate: 11 April, 2016 Weight: 200 Height: 72

Gender: ● Male ○ Female

Race: ○ AmericanIndian ○ Asian ○ Black ○ Hawaiian ● White ○ Other

Ethnicity: ● NonHispanic ○ Hispanic

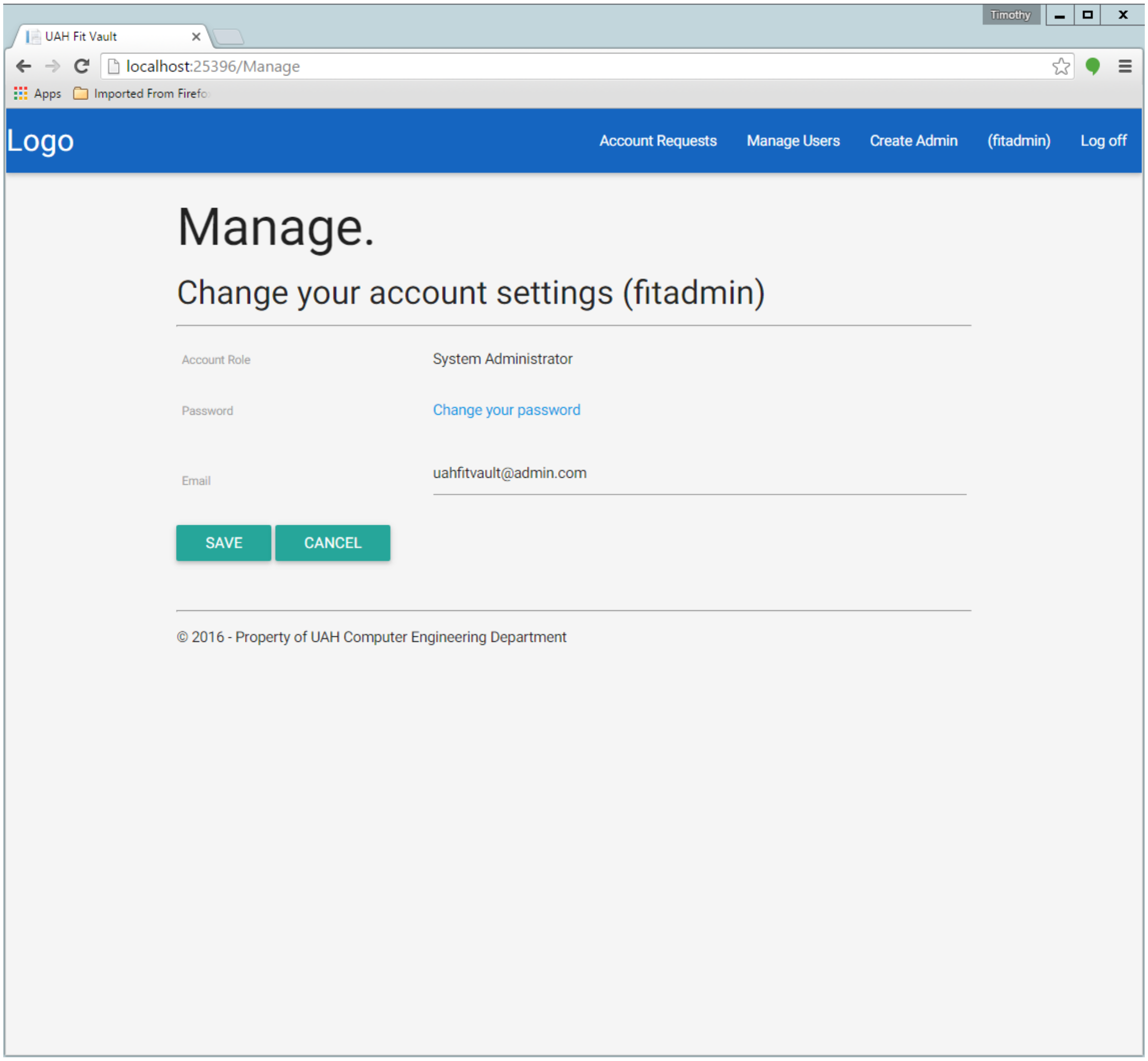
Location: Alabama ▾

© 2016 - Property of UAH Computer Engineering Department

Then fill out the form with any information you want to edit and click the “SAVE” button.

System Admin Account

To edit a System Admin account login as the system admin; then, click on the username at the top right corner of the page. This should bring you to a page that looks like this:



Manage.

Change your account settings (fitadmin)

Account Role	System Administrator
Password	Change your password
Email	uahfitvault@admin.com

© 2016 - Property of UAH Computer Engineering Department

You can then use the form to edit any information you need for the system admin account. Then click the "SAVE" button to save your changes.

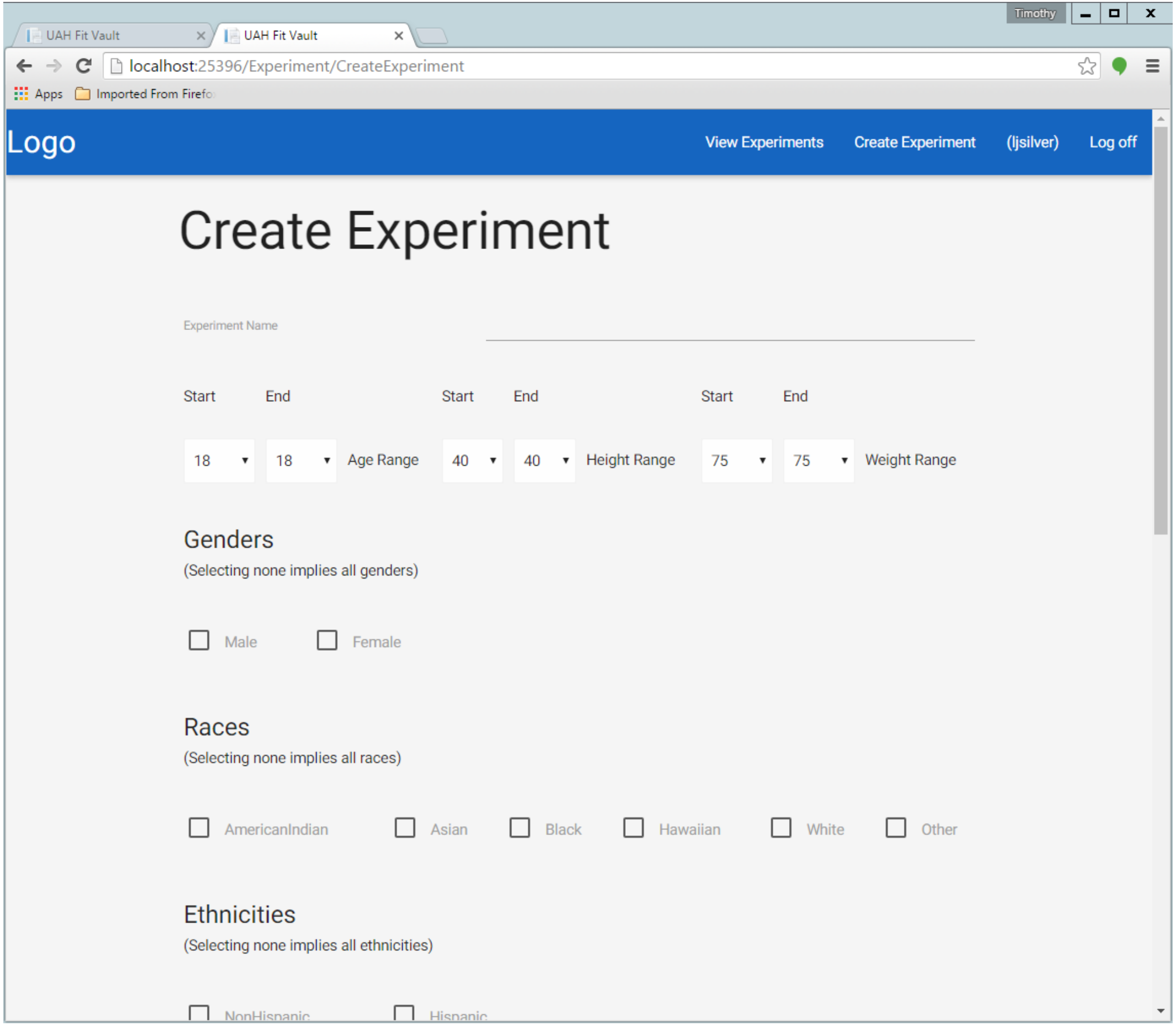
1.2.2 Experiment Management

Contents:

Experiment Creation

Table of Contents

Experiments can be created by an Experiment Admin. To create an experiment go to the home page and login with your experiment admin credentials. Click on the "Create Experiment" button at the top right of the screen. This should take you to a page like this:



Create Experiment

Experiment Name

Start	End		Start	End		Start	End	
18 ▾	18 ▾	Age Range	40 ▾	40 ▾	Height Range	75 ▾	75 ▾	Weight Range

Genders

(Selecting none implies all genders)

□ Male □ Female

Races

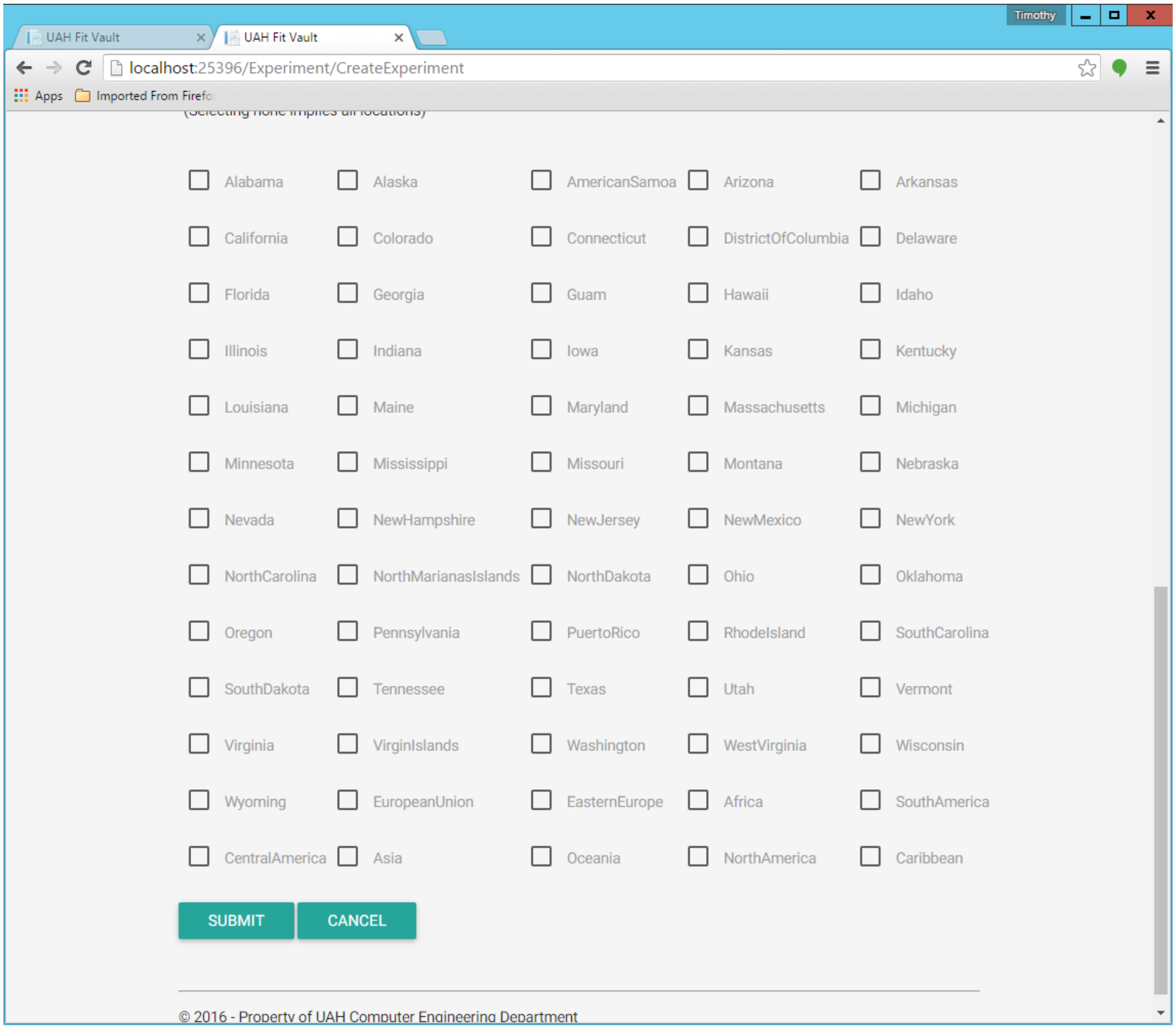
(Selecting none implies all races)

□ AmericanIndian □ Asian □ Black □ Hawaiian □ White □ Other

Ethnicities

(Selecting none implies all ethnicities)

□ NonHispanic □ Hispanic

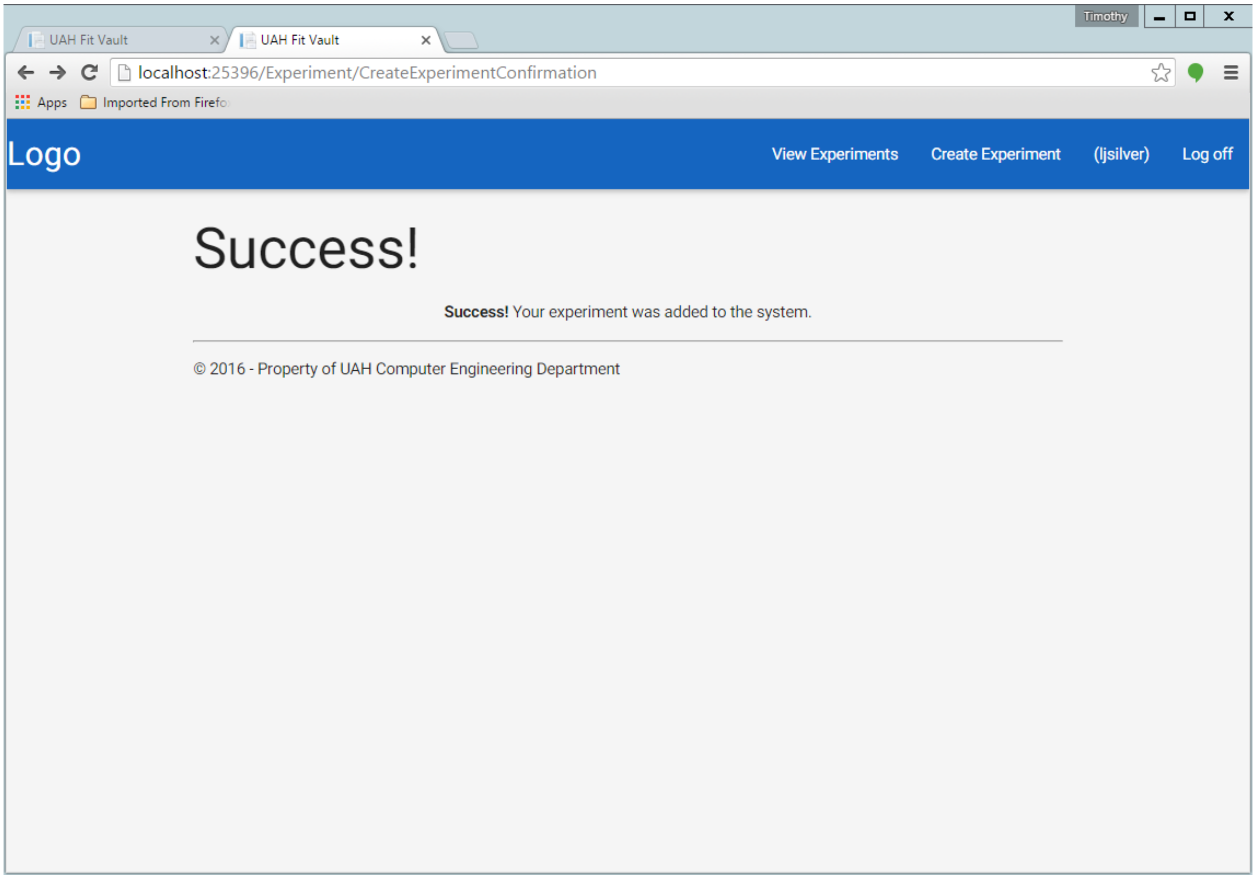


(Selecting none implies all locations)

□ Alabama	□ Alaska	□ AmericanSamoa	□ Arizona	□ Arkansas
□ California	□ Colorado	□ Connecticut	□ DistrictOfColumbia	□ Delaware
□ Florida	□ Georgia	□ Guam	□ Hawaii	□ Idaho
□ Illinois	□ Indiana	□ Iowa	□ Kansas	□ Kentucky
□ Louisiana	□ Maine	□ Maryland	□ Massachusetts	□ Michigan
□ Minnesota	□ Mississippi	□ Missouri	□ Montana	□ Nebraska
□ Nevada	□ NewHampshire	□ NewJersey	□ NewMexico	□ NewYork
□ NorthCarolina	□ NorthMarianasIslands	□ NorthDakota	□ Ohio	□ Oklahoma
□ Oregon	□ Pennsylvania	□ PuertoRico	□ RhodeIsland	□ SouthCarolina
□ SouthDakota	□ Tennessee	□ Texas	□ Utah	□ Vermont
□ Virginia	□ VirginIslands	□ Washington	□ WestVirginia	□ Wisconsin
□ Wyoming	□ EuropeanUnion	□ EasternEurope	□ Africa	□ SouthAmerica
□ CentralAmerica	□ Asia	□ Oceania	□ NorthAmerica	□ Caribbean

© 2016 - Property of UAH Computer Engineering Department

You can then fill out information that will filter out patients for your experiment. Once you have filled out the form completely to your liking, click the “SUBMIT” button at the bottom of the form. If everything worked right, you should be presented with a confirmation page like this:



Success!

Success! Your experiment was added to the system.

© 2016 - Property of UAH Computer Engineering Department

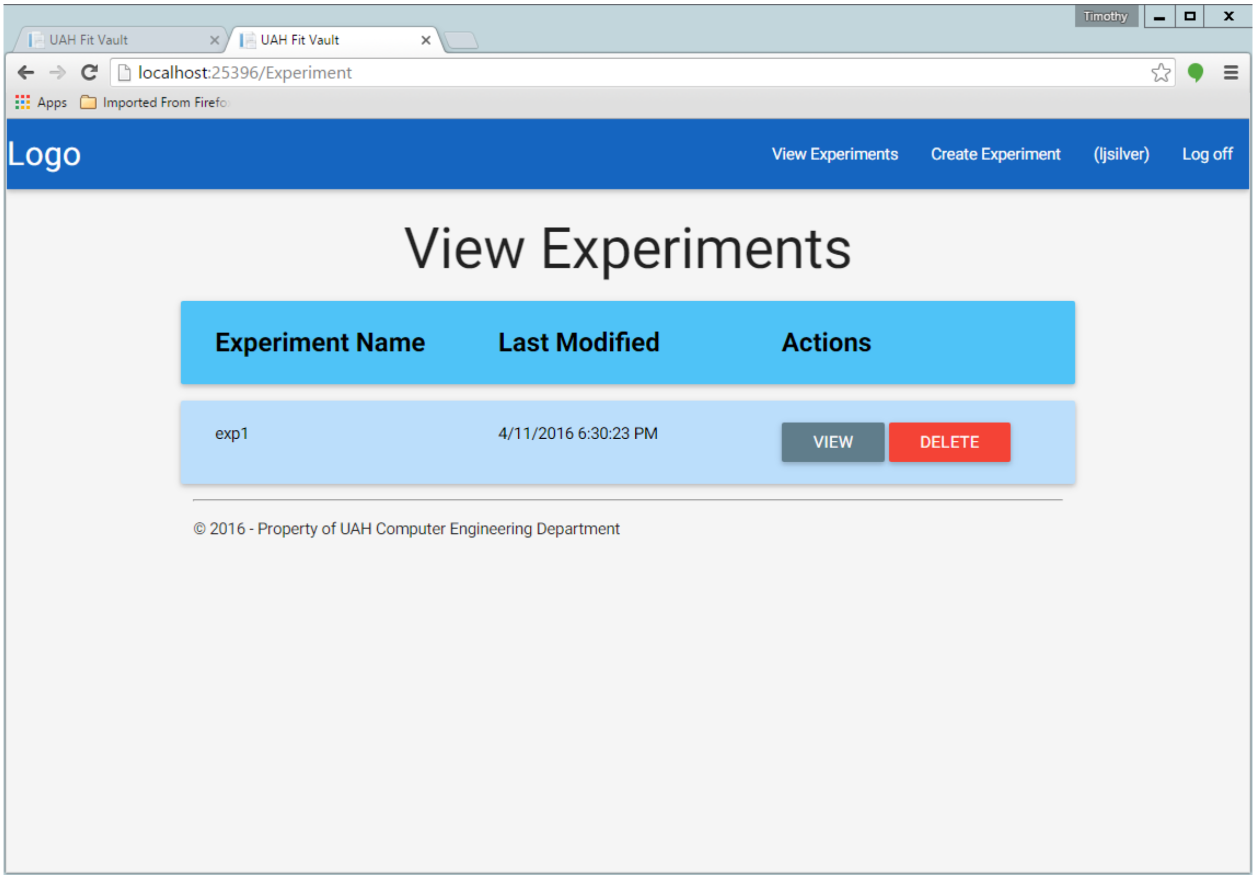
Experiment View

Table of Contents

Both Physicians and Experiment Admins can view experiments.

Experiment Admin

Once an experiment has been created, login with your experiment admin credentials and click on the “View Experiment” button at the top right corner of the page. You should be taken to a page that looks like this:

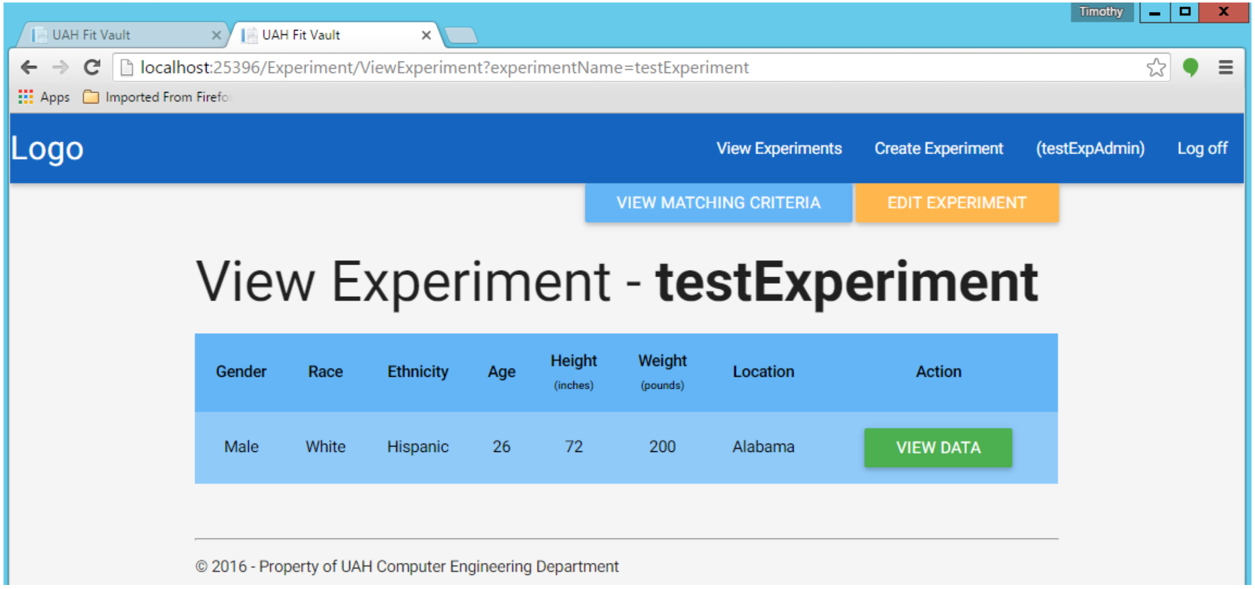


View Experiments

Experiment Name	Last Modified	Actions
exp1	4/11/2016 6:30:23 PM	VIEW DELETE

© 2016 - Property of UAH Computer Engineering Department

You can then click on the "VIEW" button of the corresponding experiment you want to view. Then this should take you to to a page that looks like this:

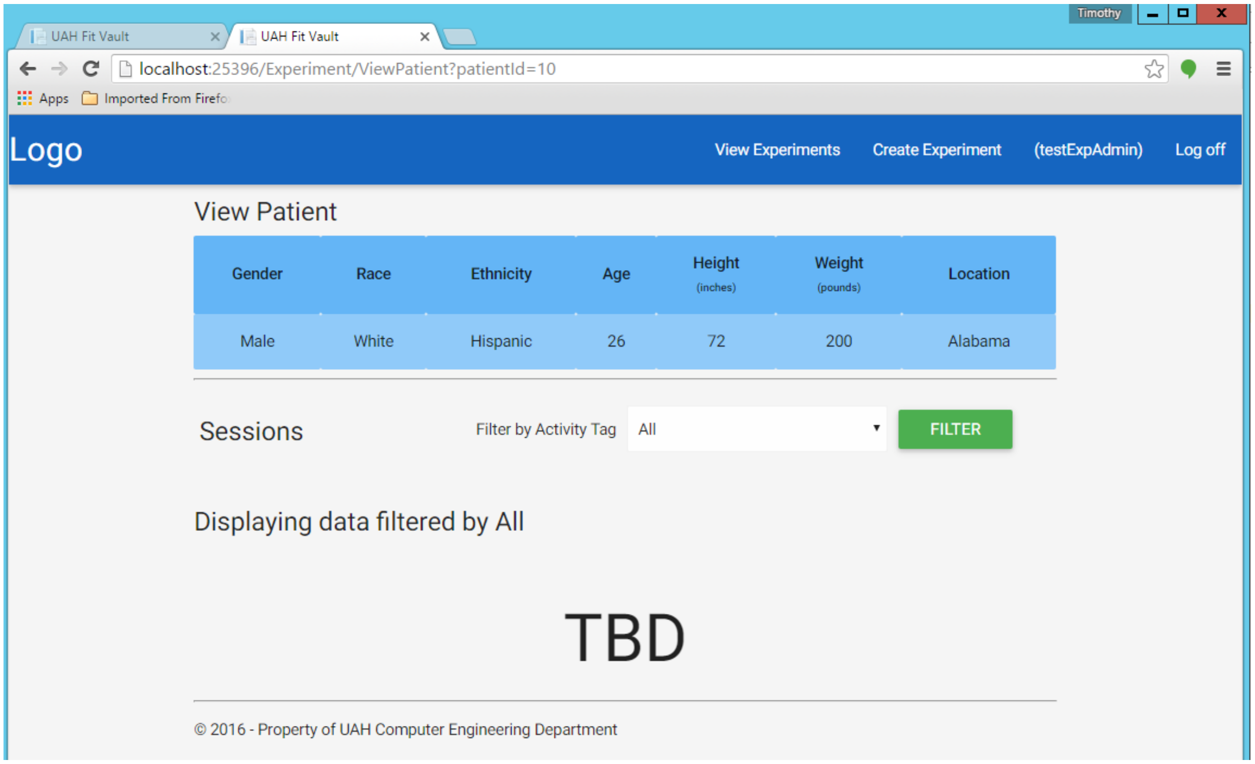


View Experiment - **testExperiment**

Gender	Race	Ethnicity	Age	Height (inches)	Weight (pounds)	Location	Action
Male	White	Hispanic	26	72	200	Alabama	VIEW DATA

© 2016 - Property of UAH Computer Engineering Department

You can now view the data for all the patients that fit into the experiment:



View Patient

Gender	Race	Ethnicity	Age	Height (inches)	Weight (pounds)	Location
Male	White	Hispanic	26	72	200	Alabama

Sessions

Filter by Activity Tag: All

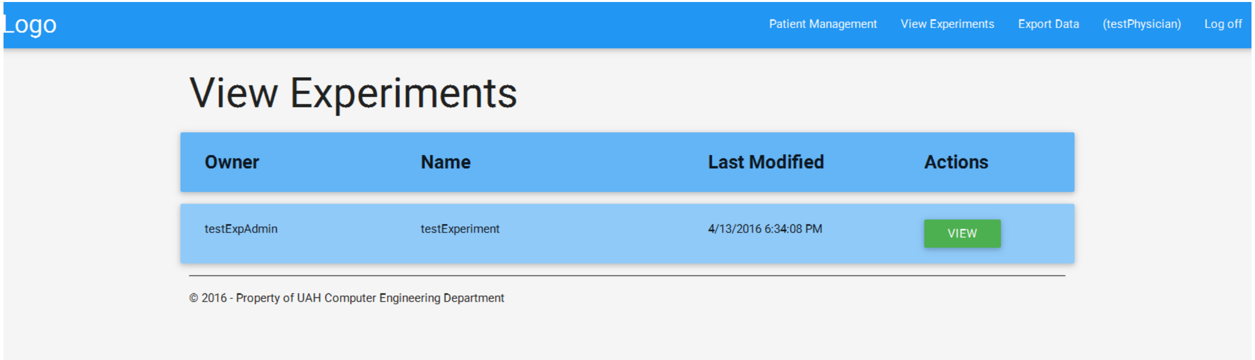
Displaying data filtered by All

TBD

© 2016 - Property of UAH Computer Engineering Department

Physicians

Login with your physician credentials and click the “View Experiments” button at the top right corner of the page. You should be taken to a page that looks like this:

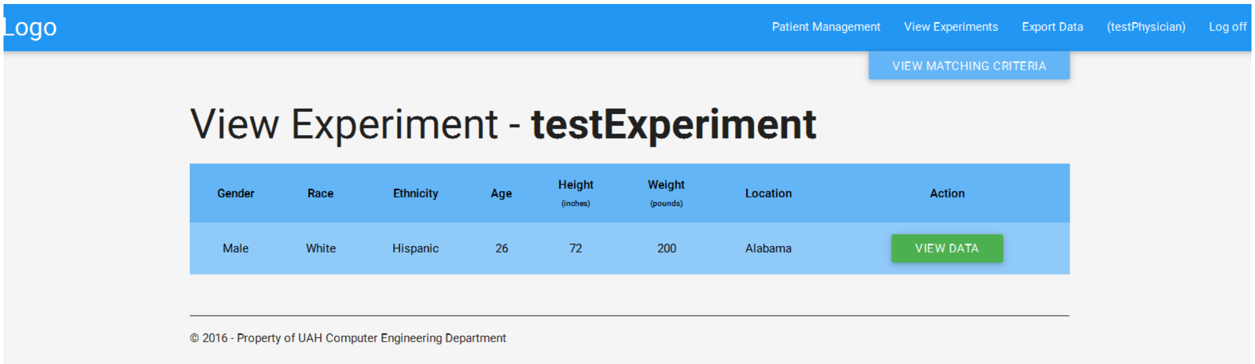


View Experiments

Owner	Name	Last Modified	Actions
testExpAdmin	testExperiment	4/13/2016 6:34:08 PM	VIEW

© 2016 - Property of UAH Computer Engineering Department

You can then click on the green “VIEW” button to the corresponding experiment you want to view. This should take you to a page that looks like this:

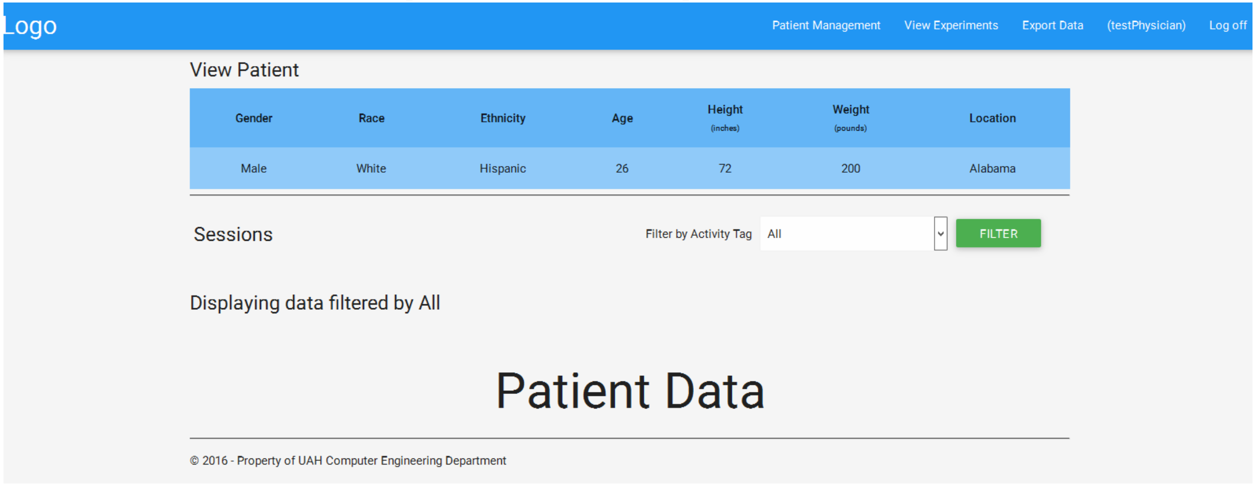


View Experiment - **testExperiment**

Gender	Race	Ethnicity	Age	Height (inches)	Weight (pounds)	Location	Action
Male	White	Hispanic	26	72	200	Alabama	VIEW DATA

© 2016 - Property of UAH Computer Engineering Department

You can now view the data for all the patients that fit into the experiment:



View Patient

Gender	Race	Ethnicity	Age	Height (inches)	Weight (pounds)	Location
Male	White	Hispanic	26	72	200	Alabama

Sessions

Displaying data filtered by All

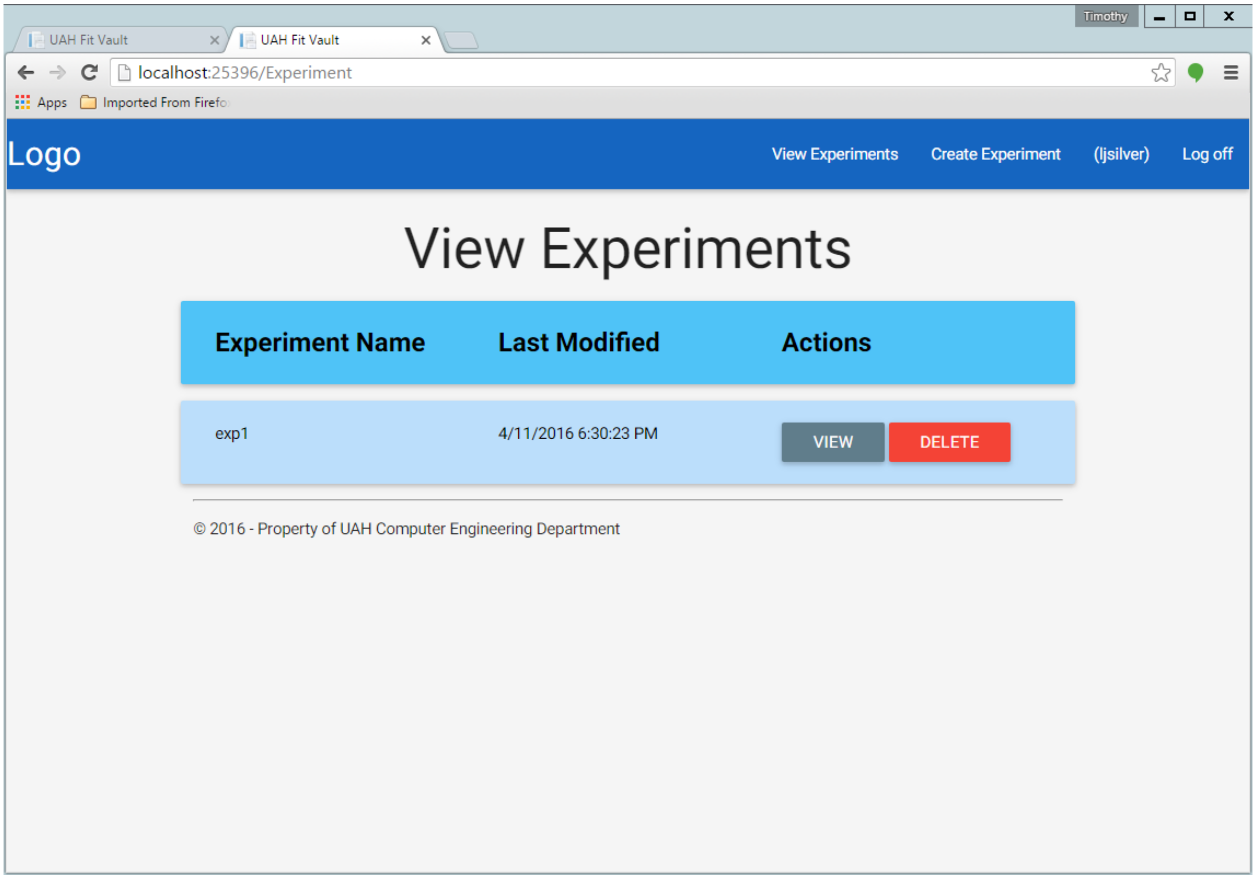
Patient Data

© 2016 - Property of UAH Computer Engineering Department

Experiment Deletion

Table of Contents

Only an Experiment Admin can delete experiments. To do this, simply login with your Experiment Admin credentials, and click on the "View Experiments" button at the top right corner of the page. This should take you to a page that looks like this:



View Experiments

Experiment Name	Last Modified	Actions
exp1	4/11/2016 6:30:23 PM	VIEW DELETE

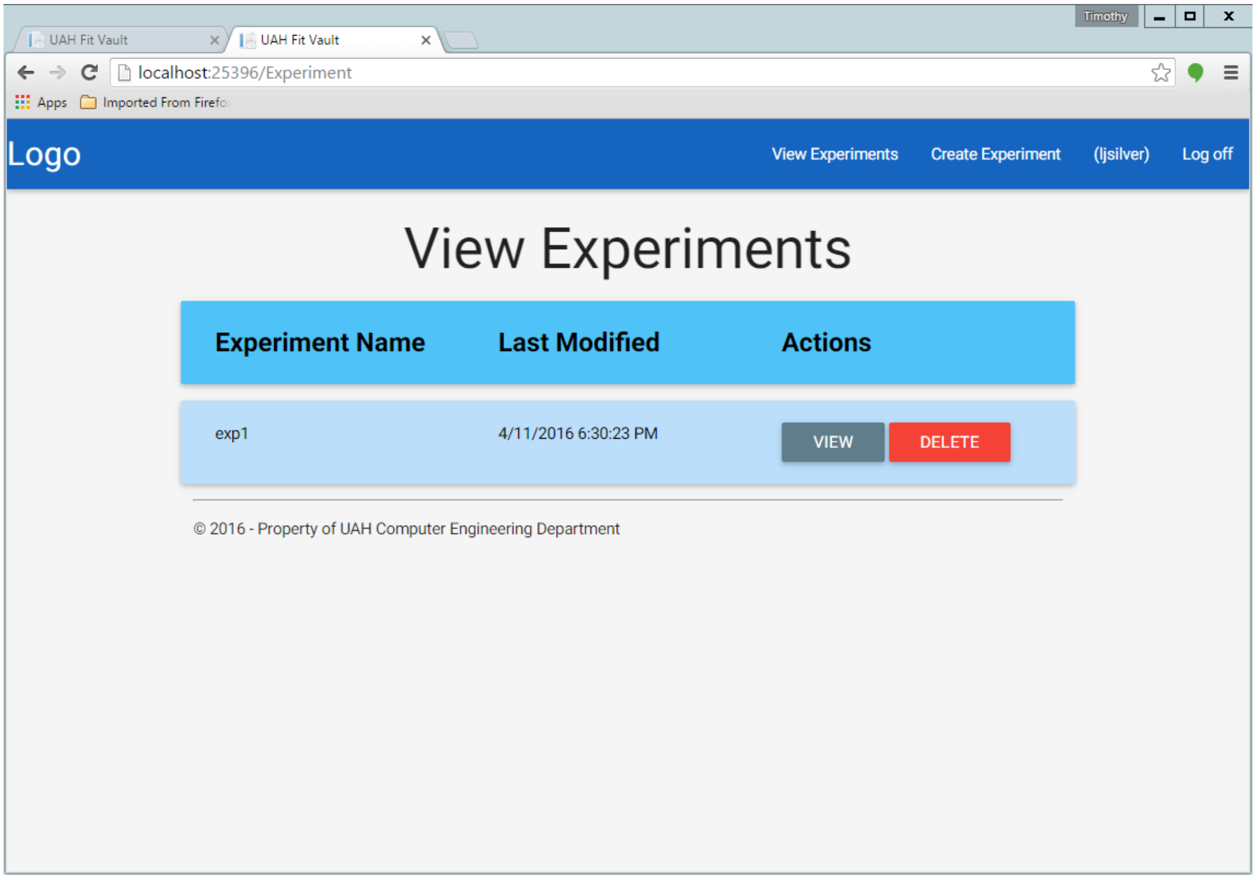
© 2016 - Property of UAH Computer Engineering Department

You can then click on the “DELETE” button of the corresponding experiment you want to delete.

Experiment Exportation

Table of Contents

An Experiment Admin can export experiment data. Login as an experiment administrator. Then click on the “View Experiment” button.

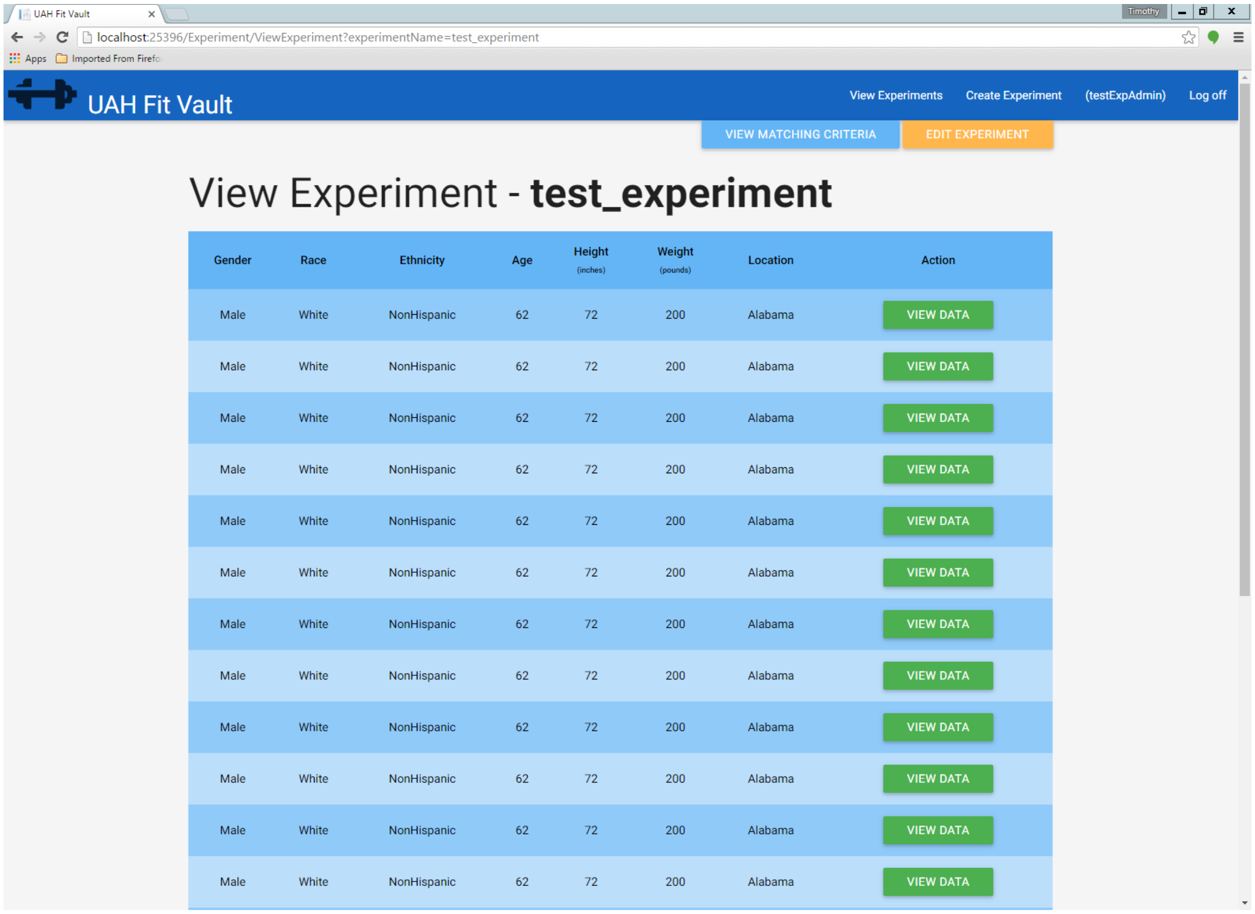


View Experiments

Experiment Name	Last Modified	Actions
exp1	4/11/2016 6:30:23 PM	VIEW DELETE

© 2016 - Property of UAH Computer Engineering Department

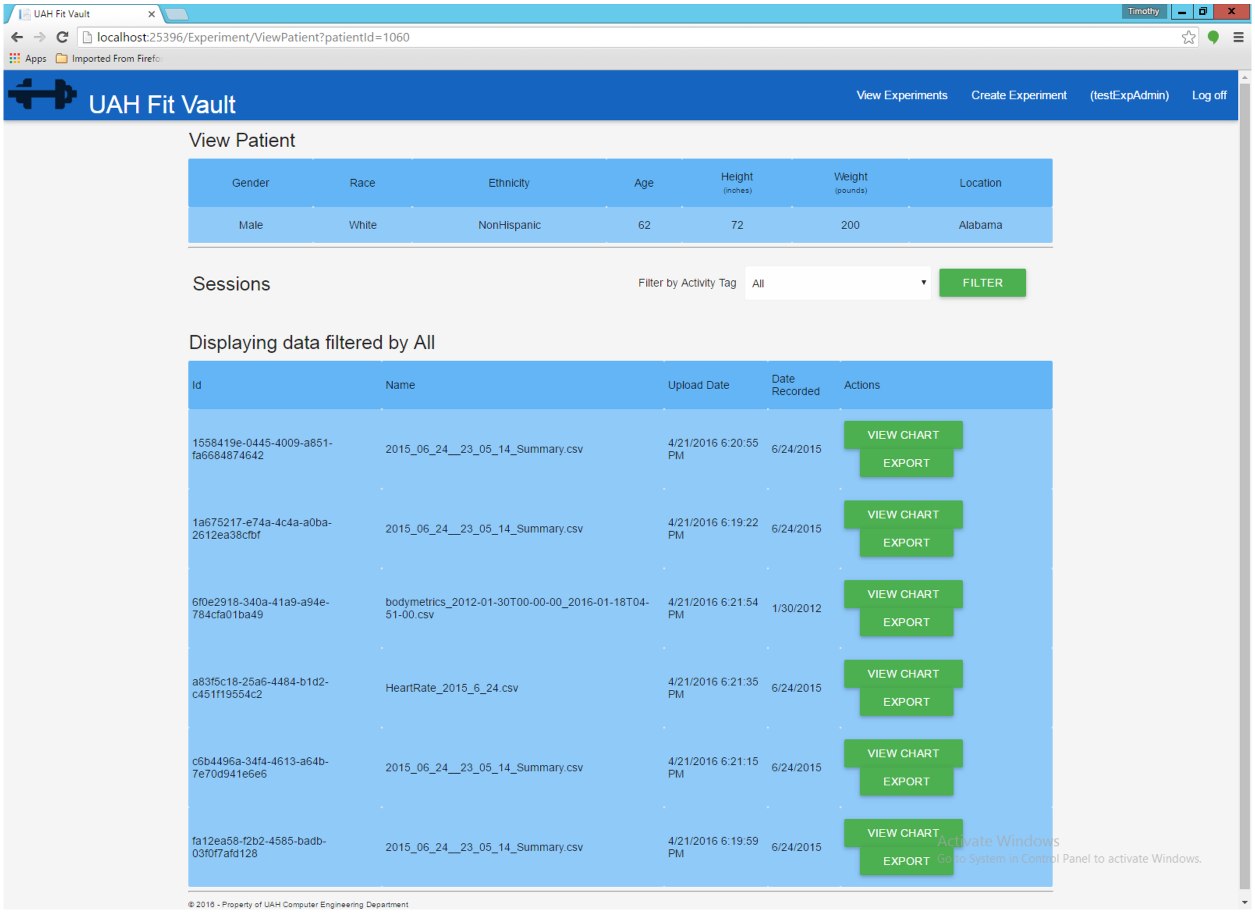
You should then see existing experiments. Click on the “View” button for the experiment you want to export. You should see the following page:



View Experiment - **test_experiment**

Gender	Race	Ethnicity	Age	Height (inches)	Weight (pounds)	Location	Action
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA
Male	White	NonHispanic	62	72	200	Alabama	VIEW DATA

Next click on the “View Data” button for the account you want to download data from. You should see a page that looks like this:



View Patient

Gender	Race	Ethnicity	Age	Height (inches)	Weight (pounds)	Location
Male	White	NonHispanic	62	72	200	Alabama

Sessions

Filter by Activity Tag: All

Displaying data filtered by All

Id	Name	Upload Date	Date Recorded	Actions
1558419e-0445-4009-a851-fa6684874642	2015_06_24__23_05_14_Summary.csv	4/21/2016 6:20:55 PM	6/24/2015	VIEW CHART EXPORT
1a675217-e74a-4c4a-a0ba-2612ea38cfbf	2015_06_24__23_05_14_Summary.csv	4/21/2016 6:19:22 PM	6/24/2015	VIEW CHART EXPORT
6f0e2918-340a-41a9-a94e-784cfa01ba49	bodymetrics_2012-01-30T00-00-00_2016-01-18T04-51-00.csv	4/21/2016 6:21:54 PM	1/30/2012	VIEW CHART EXPORT
a83f5c18-25a6-4484-b1d2-c451f19554c2	HeartRate_2015_6_24.csv	4/21/2016 6:21:35 PM	6/24/2015	VIEW CHART EXPORT
c6b4496a-34f4-4613-a64b-7e70d941e6e6	2015_06_24__23_05_14_Summary.csv	4/21/2016 6:21:15 PM	6/24/2015	VIEW CHART EXPORT
fa12ea58-f2b2-4585-badb-03f0f7afd128	2015_06_24__23_05_14_Summary.csv	4/21/2016 6:19:59 PM	6/24/2015	VIEW CHART EXPORT

You can then click on the "Export" button for the file you want to export. This will bring up your browser's download wizard. Follow the prompts to download the file.

1.2.3 Patient Data Management

Patients have the ability to upload data to the system, view the data they have uploaded, and download that data from the system at any time.

Contents:

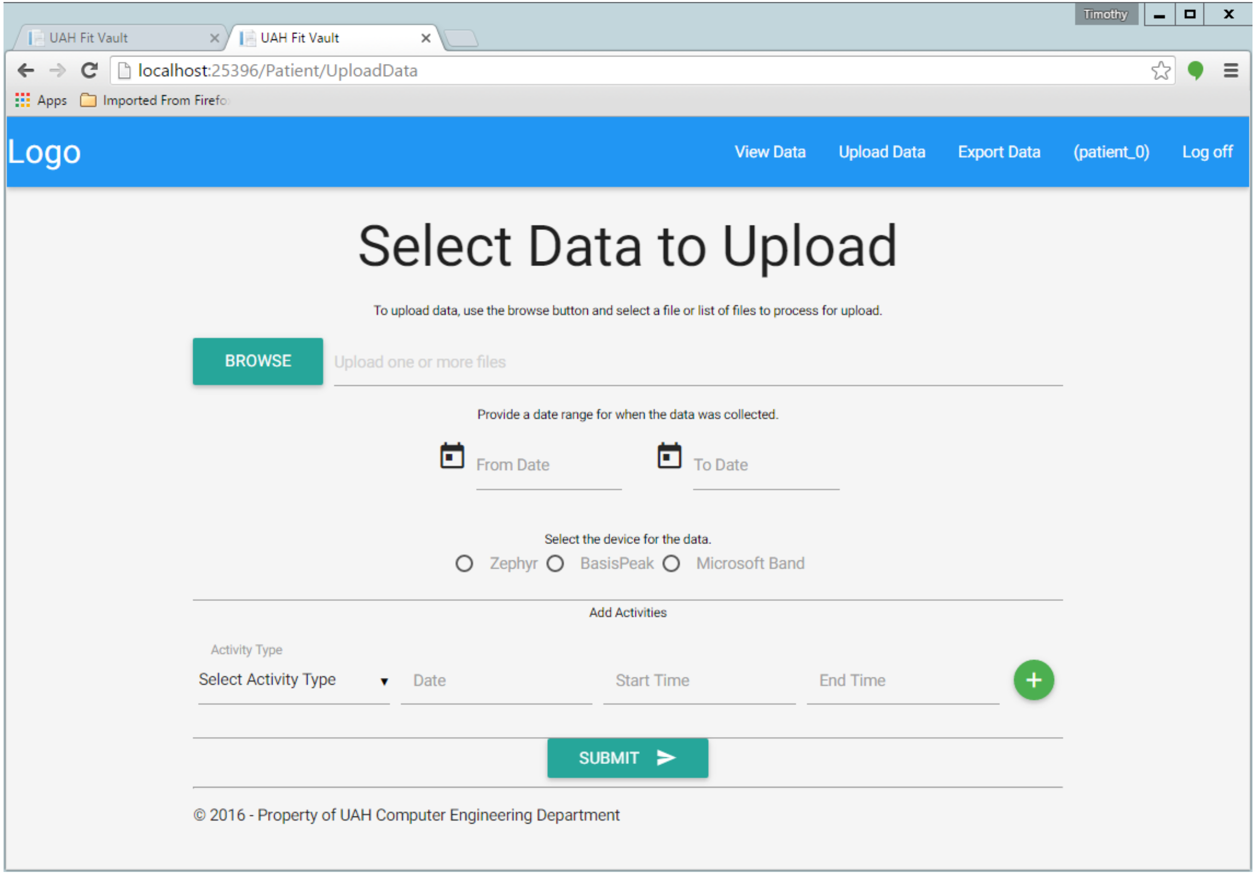
Patient Data Upload

Table of Contents

A patient can upload data to the system from 3 types of devices:

- Zephyr
- Basis Peak
- Microsoft Band

To upload a file, login with your patient account credentials. Then, click on the “Upload Data” button at the top right of the page. This should take you to a screen like this:



Select Data to Upload

To upload data, use the browse button and select a file or list of files to process for upload.

Provide a date range for when the data was collected.

Select the device for the data.

Add Activities

© 2016 - Property of UAH Computer Engineering Department

Click the “BROWSE” button and follow the wizard to the file(s) you want to upload. Fill in the date of collection information and select the device type. At this point you can upload data by clicking the “SUBMIT” button.

Your Physician may also ask you to fill out activities (the second half of the Upload Data page). Activities must be set at the time of the data upload. To fill out an activity simply choose an activity type from the “Activity Type” dropdown, select the date the activity was performed, and select a start and end time for the activity.

You can input as many activities as you want. To add an activity click the green plus button to the right of the activity section of the page.

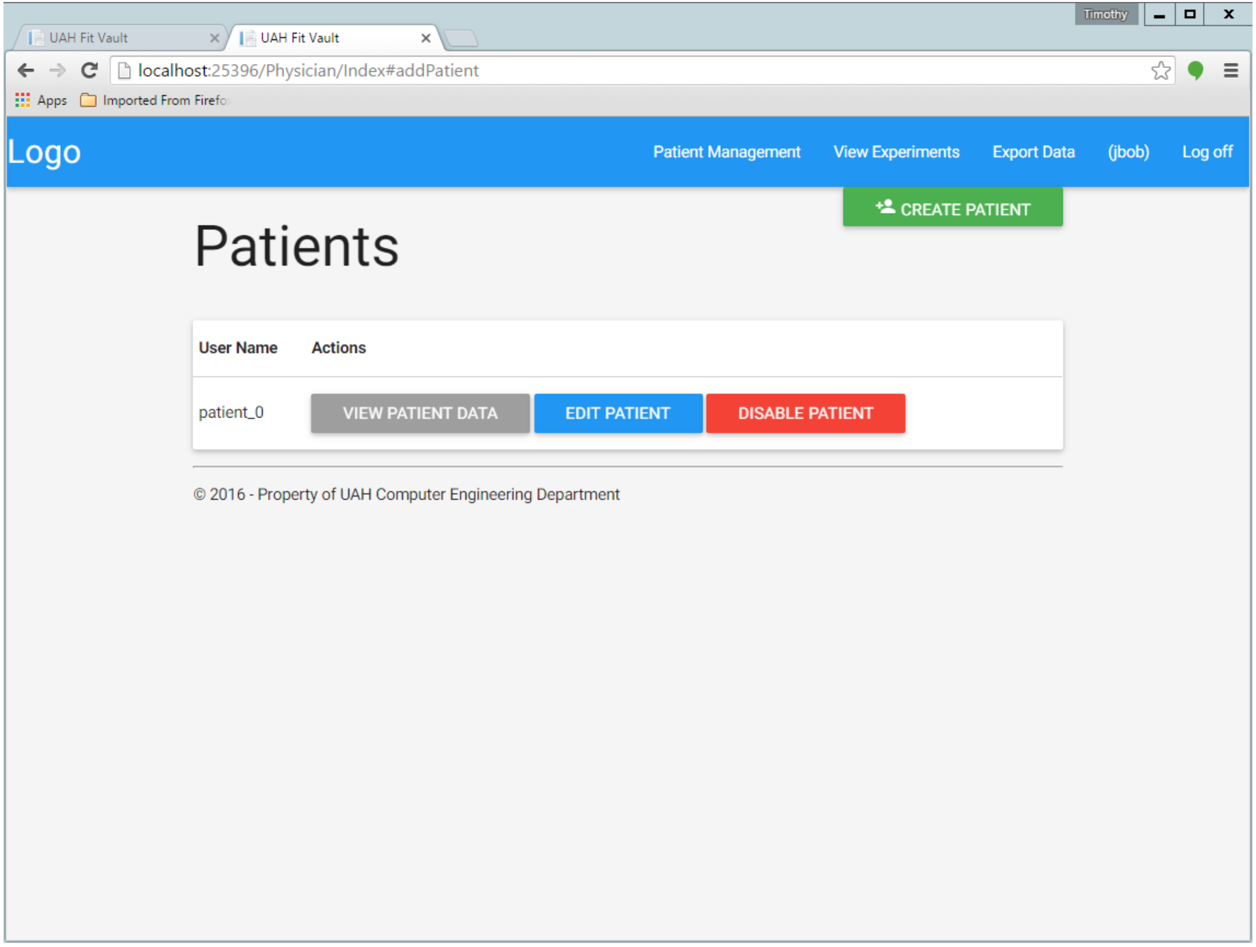
Patient Data View

Table of Contents

Both Physicians and Patients can view a patient’s data.

Physician

To view patient data as a physician, login with your physician account credentials. Click on the “Patient Management” button at the top right of the page. You should then be taken to a page that looks like this:

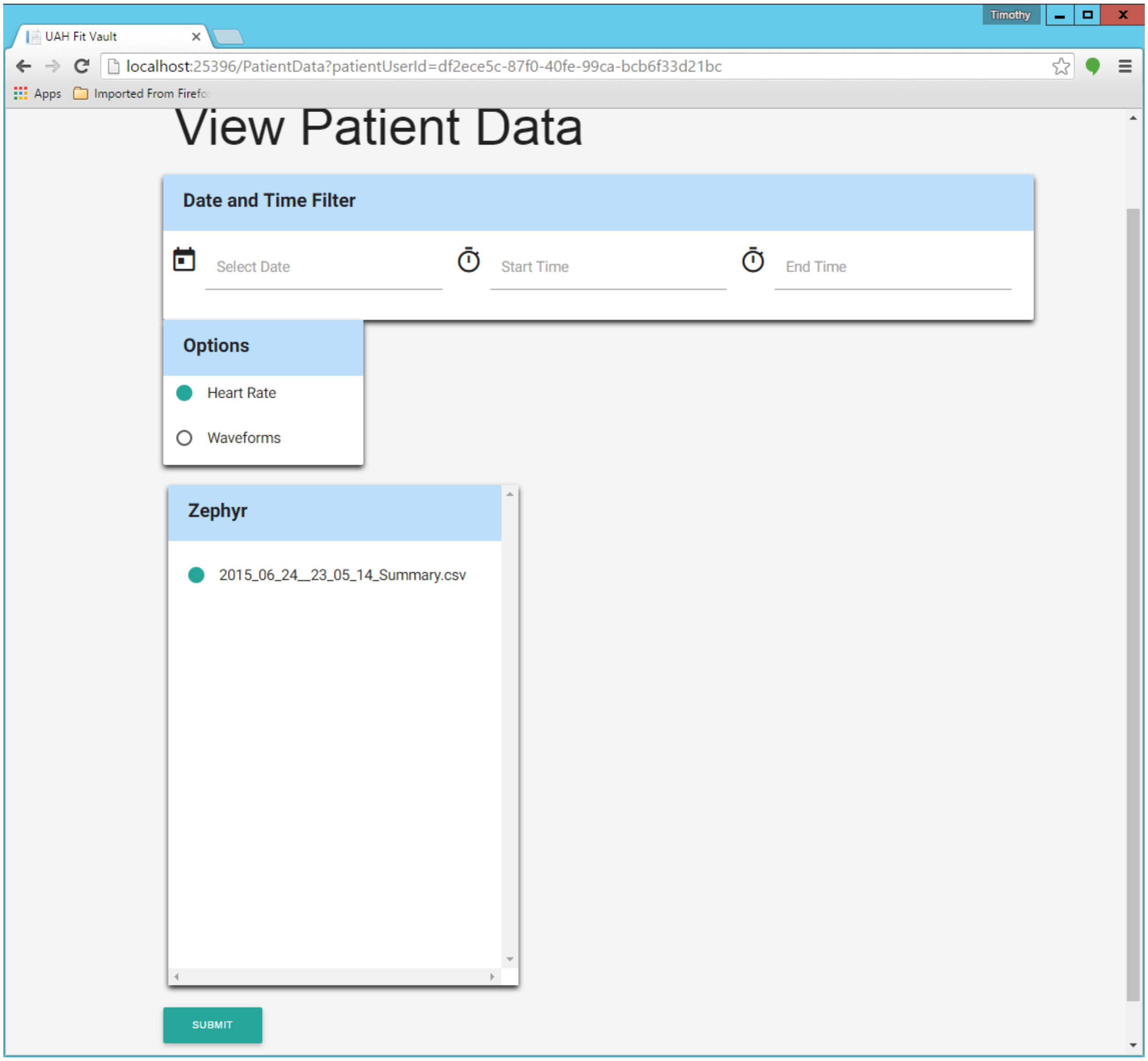


Patients

User Name	Actions
patient_0	VIEW PATIENT DATA EDIT PATIENT DISABLE PATIENT

© 2016 - Property of UAH Computer Engineering Department

You can then click on the “VIEW PATIENT DATA” button for the corresponding patient who’s data you want to view. You will then be taken to a page that looks like this:


Timothy
UAH Fit Vault
localhost:25396/PatientData?patientUserId=df2ece5c-87f0-40fe-99ca-bcb6f33d21bc
Apps
Imported From Firefo

View Patient Data

Date and Time Filter

Select Date

Start Time

End Time

Options

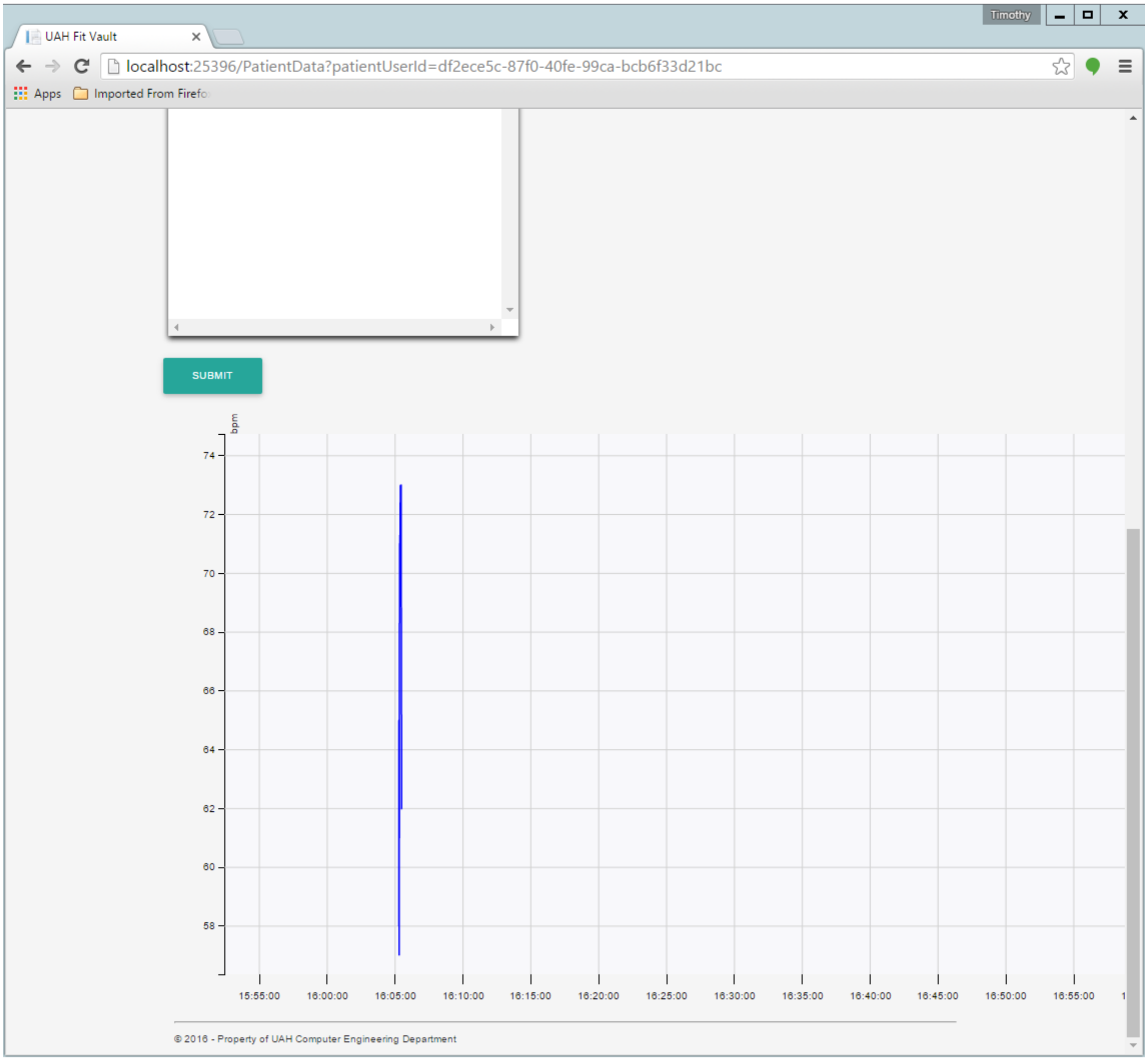
● Heart Rate

○ Waveforms

Zephyr

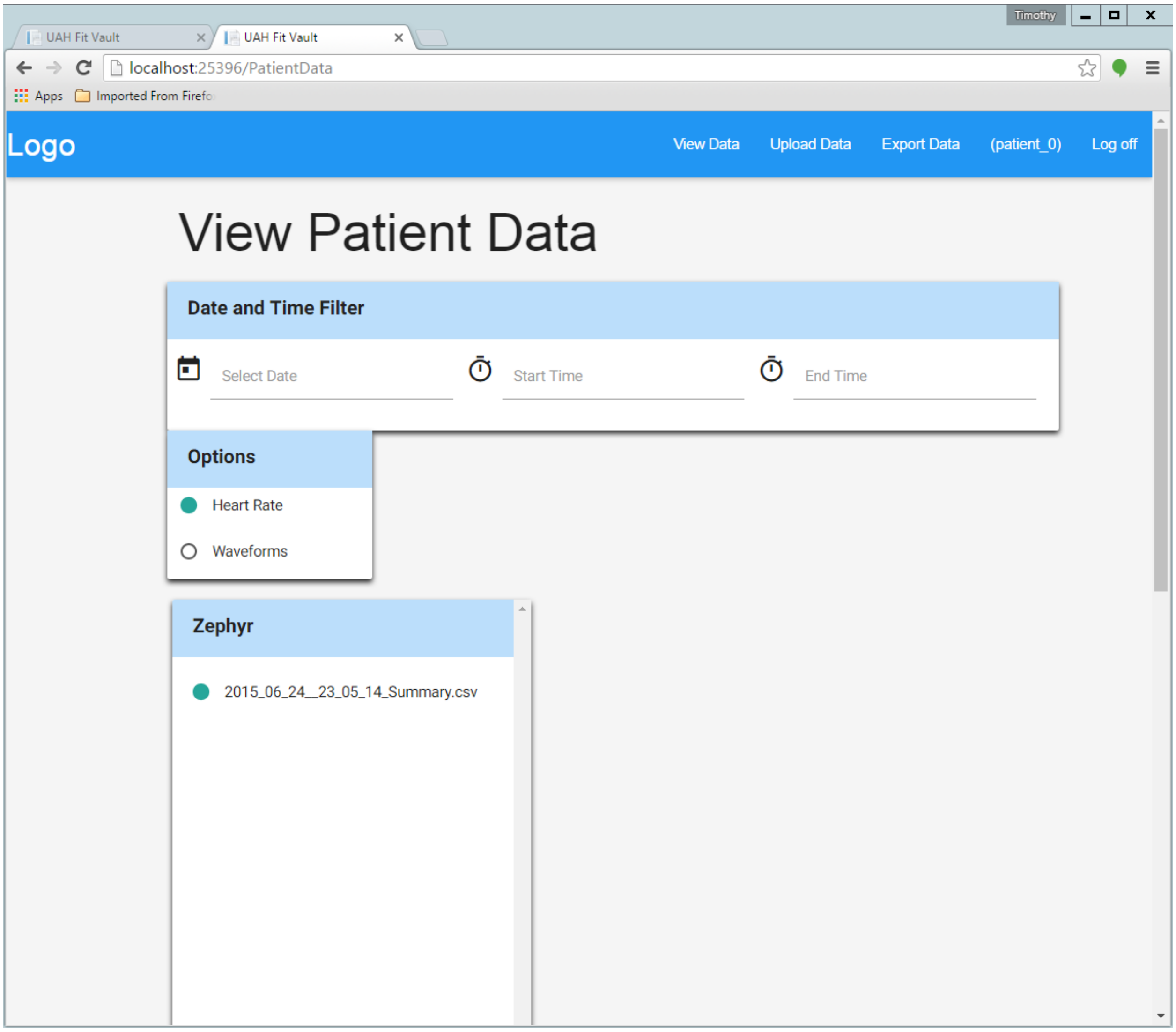
● 2015_06_24__23_05_14_Summary.csv

You can then select a date and a start time and end time. Then select the file you want to view. Click the "SUBMIT" button to display the data at the bottom of the page. It should look like this:

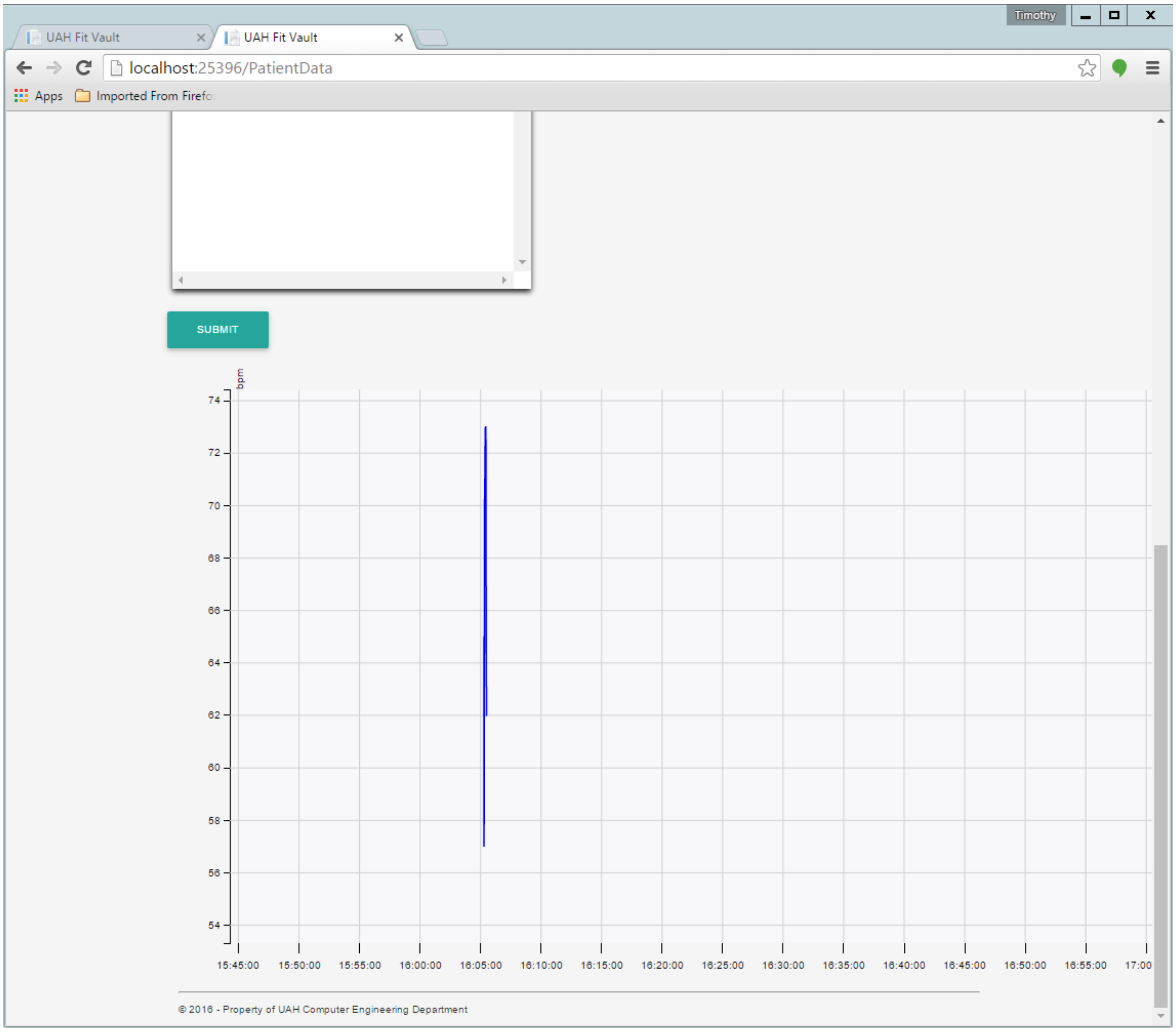


Patient

To view patient data as a patient, login with your patient account credentials. You can then click the “View Data” button at the top right of the page. You should then be taken to a page that looks like this:



You can then select a date and a start time and end time. Then select the file you want to view. Click the "SUBMIT" button to display the data at the bottom of the page. It should look like this:



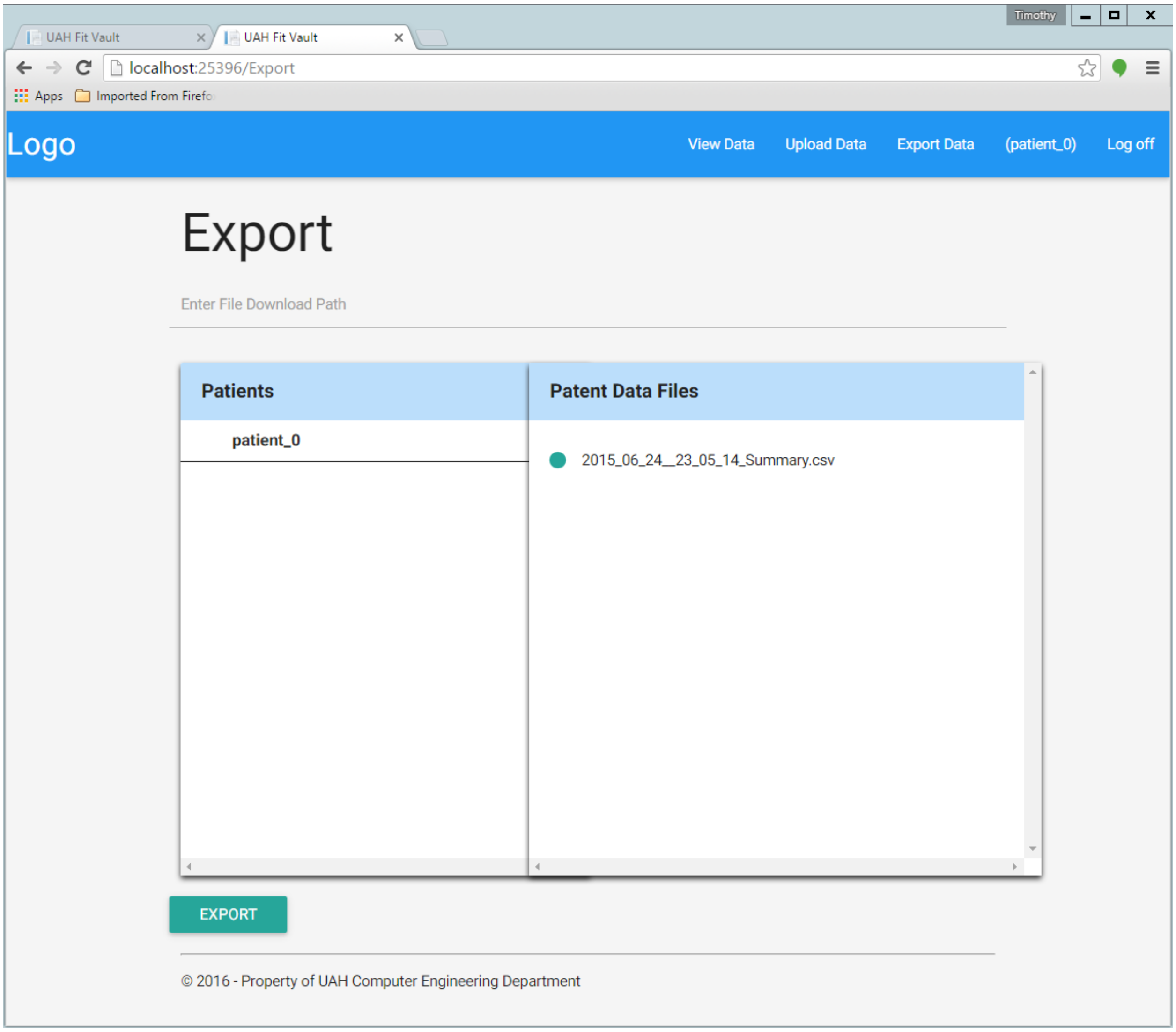
Patient Data Download

Table of Contents

Patients have the ability to export any data they have uploaded.

Patient Account

Login with your patient credentials. You should see an "Export Data" button at the top right of the page. Click on this button and you will be taken to a page that looks like this:



Click on your patient account in the "Patients" window, then click on the file you want to export in the "Patient Data Files" window. Then simply click the "EXPORT" button at the bottom of the page to start the download.

CHAPTER

TWO

UAHEALTH BIT VAULT SOFTWARE TEST DESCRIPTION

Table of Contents

2.1 Sign Off

The sign off below indicates that the project has been accepted as is at the moment of signing. No further obligations are required by the development team. Please sign by your name and date:

```
Dr. Kulick:        ________________________________________ Date: ____________
Dr. Milenkovic:    ________________________________________ Date: ____________
James Duggan:      ________________________________________ Date: ____________
Glen Riden:        ________________________________________ Date: ____________
Ryan Wilkins:      ________________________________________ Date: ____________
```

2.2 Scope

2.2.1 Identification

This document shall apply to the final release of the UAHealth Bit Vault Software.

2.2.2 System Overview

The UAHealth Bit Vault Software is designed to be a place for doctors, patients, and administrators to store patient's biometric data and run experiments on that data. This is the first deployment of this system. The software will be deployed on a server in the UAH lab for Dr. Malenkovich. At the end of the development period, the software will be deployed and ownership shall be transferred to Dr. Malenkovich.

2.2.3 Document Overview

This document is to describe the testing that will be performed on the UAHealth Bit Vault Software for customer acceptance. Test descriptions and all dependent information are included in this document.

2.3 Referenced Documents

None.

2.4 Test Descriptions

2.4.1 Pytest Tests

Table of Contents

Test Preparations

Hardware Preparations

In order to run the Pytest tests, you must have a physical computer capable of running the latest distribution of Ubuntu. Follow the hardware requirements for this distribution. Ubuntu is not required to run the tests, but the hardware requirements for Ubuntu serve as a minimum requirement to run the Pytest tests.

The computer you use must have access to the UAHealth server in order to run the Pytest tests. The computer can be the same machine as the UAHealth server, but this is not required.

Software Preparations

Download the zip file of the github project here: Github and unzip it in your home directory.

You must have the latest Active Python 3.4 installed on this computer. It can be found here: Python. The OS may be Windows 7/2012 or later. If you are using Windows 2012, add “C:\Python34” and “C:\Python34\Scripts” to your “Path” environment variable. You also need to install Firefox. It can be downloaded here: Firefox.

After you download and install Firefox, follow instructions on Mozilla for setting the download automatic action.

The required python packages can be installed by navigating to the project folder and running:

```
pip install -r ./Tests/requirements.txt
```

Other Preparations

Test Execution

To run these tests, open a command prompt and navigate to the Tests folder (<project_path>/med656/Tests). Then execute the following command:

py.test -v -s test_create_account.py test_edit_account.py test_experiment.py test_login.py test_password.py test_security.py test_upload.py test_username.py test_export.py

The test report will be printed out to the screen. It will display all the tests that were run along with their pass/fail status.

Test Details

Follow the table below to find details of the test automation.

test_create_account module

These test cases are designed to test the ability to create and delete accounts

`test_create_account.`**`test_create_experiment_admin`**`(`*logoff*`)`

Requirements:

- 3.1.1.1.3.1: minimum information stored for experiment admin account
- 3.1.1.1.4.2: System Administrators shall be able to enable users
- 3.1.1.7: Experiment administrators create their own accounts
- 3.1.1.7.1: System Administrators shall verify the creation of a experiment admin account
- 3.1.6: The system shall provide account management.
- 3.1.6.1: The system shall allow account creation
- 3.1.6.1.1: The system shall force account approval for experiment admins
- 3.1.1.4: System requires a minimum security criteria
- 3.1.1.4.1: Password must be at least 10 characters
- 3.1.1.4.2: Password shall contain at least 1 upper and 1 lower case character
- 3.1.1.4.3: Password shall contain at least 1 number
- 3.1.1.4.4: Password shall contain at least 1 special character

Pre Conditions:

- logoff fixture

Input:

- user = ‘awong’
- pwd = ‘P@ssword10‘
- email = ‘awong@futurama.com‘
- first_name = ‘Amy’
- last_name = ‘Wong’

- address = ‘304 Wherever Street, New New York City, New New York’
- phone_number = ‘123-456-7890’

Steps	Expected Result	Actual Result
request account	Request confirmed	
approve account	Admin can approve	
login	User can login	

`test_create_account.`**`test_create_patient`**(*login_tphysician*)

Requirements:

- 3.1.1.1.1.1: minimum information stored for patient account
- 3.1.1.1.2.5: Physician user can add a patient to the system
- 3.1.1.3.1: Display patient ID to physician at the time of creation
- 3.1.1.7.2: Physicians shall create their patients accounts
- 3.1.6: The system shall provide account management.
- 3.1.6.1: The system shall allow account creation
- 3.1.6.1.2: Patients shall not need approval for account creation
- 3.1.1.4: System requires a minimum security criteria
- 3.1.1.4.1: Password must be at least 10 characters
- 3.1.1.4.2: Password shall contain at least 1 upper and 1 lower case character
- 3.1.1.4.3: Password shall contain at least 1 number
- 3.1.1.4.4: Password shall contain at least 1 special character

Pre Conditions:

- login_tphysician fixture

Input:

- user = ‘testPatientCreate_<random number>’
- pwd = ‘P@ssword10‘
- byear = ‘1954’
- bmonth = ‘March’
- bday = ‘3’
- loc = ‘Alabama’
- wght = ‘200’
- hght = ‘72’
- gen = ‘male’
- race = ‘white’
- eth = ‘non_hispanic’
- physician = ‘testPhysician’

Steps	Expected Result	Actual Result
create patient account	No error msgs	
login as patient	Patient can login	

`test_create_account.`**`test_create_physician`**`(`*logoff*`)`

Requirements:

- 3.1.1.1.2.1: minimum information stored for physician account
- 3.1.1.1.4.2: System Administrators shall be able to enable users
- 3.1.1.7: Physicians create their own accounts
- 3.1.1.7.1: System Administrators shall verify the creation of a physician account
- 3.1.6: The system shall provide account management.
- 3.1.6.1: The system shall allow account creation
- 3.1.6.1.1: The system shall force account approval for physicians
- 3.1.1.4: System requires a minimum security criteria
- 3.1.1.4.1: Password must be at least 10 characters
- 3.1.1.4.2: Password shall contain at least 1 upper and 1 lower case character
- 3.1.1.4.3: Password shall contain at least 1 number
- 3.1.1.4.4: Password shall contain at least 1 special character

Pre Conditions:

- logoff fixture

Input:

- user = ‘hfarnswroth’
- pwd = ‘P@ssword10‘
- email = ‘hfarnsworth@futurama.com‘
- first_name = ‘Hubert’
- last_name = ‘Farnsworth’
- address = ‘304 Wherever Street, New New York City, New New York’
- phone_number = ‘123-456-7890’

Steps	Expected Result	Actual Result
request account	Request confirmed	
approve account	Admin can approve	
login	User can login	

`test_create_account.`**`test_create_system_admin`**`(`*login_sysadmin*`)`

Requirements:

- 3.1.1.1.4.1: minimum information stored for experiment admin account
- 3.1.1.1.4.2: System Administrators shall be able to enable users
- 3.1.6: The system shall provide account management.
- 3.1.6.1: The system shall allow account creation
- 3.1.1.4: System requires a minimum security criteria
- 3.1.1.4.1: Password must be at least 10 characters
- 3.1.1.4.2: Password shall contain at least 1 upper and 1 lower case character
- 3.1.1.4.3: Password shall contain at least 1 number

- 3.1.1.4.4: Password shall contain at least 1 special character

Pre Conditions:

- logoff fixture

Input:

- user = ‘lnibbler’
- pwd = ‘P@ssword10‘
- email = ‘lnibbler@futurama.com‘

Steps	Expected Result	Actual Result
Login as fitadmin	Login Successful	
Create another system admin	No errors	
login as new system admin	User can login	

test_create_account.**test_delete_experiment_admin**(*logoff*)

Requirements:

- 3.1.6.3: System administrator can delete accounts
- 3.1.6: The system shall provide account management.

Pre Conditions:

- logoff fixture

Input:

- user = ‘awong’
- username = ‘Amy Wong’
- pwd = ‘P@ssword10‘

Steps	Expected Result	Actual Result
delete experiment admin account	No error msgs	
login as patient	User cannot login	

test_create_account.**test_delete_patient**(*logoff*)

Requirements:

- 3.1.1.1.2.6: Physicians can delete patients

Pre Conditions:

- logoff fixture

Input:

- user = ‘pfry’
- pwd = ‘P@ssword10‘
- phy = ‘testPhysician’
- phy_pass = ‘P@ssword10‘

Steps	Expected Result	Actual Result
delete patient account	No error msgs	
login as patient	Patient cannot login	

test_create_account.**test_delete_physician**(*logoff*)

Requirements:

- 3.1.6.3: System administrator can delete accounts
- 3.1.6: The system shall provide account management.

Pre Conditions:

- logoff fixture

Input:

- user = 'hfarnswroth'
- username = 'Hubert Farnsworth'
- pwd = 'P@ssword10'

Steps	Expected Result	Actual Result
delete physician account	No error msgs	
login as patient	User cannot login	

test_edit_account module

These tests are here to verify that the accounts can have their information edited.

`test_edit_account.`**`test_physician_can_edit_his_account`**(*login_tphysician*)

Requirements:

- 3.1.1.1.2.2: Physician can update his account
- 3.1.6.2: System shall allow users to edit their account

Pre Conditions:

- login_tphysician fixture

Input:

Steps	Expected Result	Actual Result
Go to management page	No Errors	
Change the email address	No Errors	
Check the email address	Email has changed	

test_experiment module

These test cases are designed to test the ability to create experiments

`test_experiment.`**`test_create_experiment`**(*login_texpadmin*)

Requirements:

- 3.1.4: The system shall allow experiments to be created.
- 3.1.4.1.1: Experiment administrators shall have the ability to save the experiments that they have created.
- 3.1.4.3: The system shall provide Experiment Administrators with a query builder used to create experiments.

Pre Conditions:

- login_texpadmin fixture

Input:

- expName = 'test experiment'

- startAge = ‘19’
- endAge = ‘24’
- startHght = ‘60’
- endHght = ‘72’
- startWght = ‘130’
- endWght = ‘200’
- genders = [’male’]
- races = [’white’]
- ethnicities = [’hispanic’]
- locations = [’Alabama’]

Steps	Expected Result	Actual Result
create experiment	no errors, see success! page	

`test_experiment.`**`test_delete_experiment`**(*login_texpadmin*)

Requirements:

- 3.1.4.4: The system shall allow Experiment Administrators to delete experiments.

Pre Conditions:

- login_texpadmin fixture

Input:

- name = ‘test experiment’

Steps	Expected Result	Actual Result
delete experiment	no errors, on delete confirmation page	

`test_experiment.`**`test_view_experiment_exp_admin`**(*login_texpadmin*)

Requirements:

- 3.1.4.2: Experiment results shall be viewable by experiment administrators and physicians.

Pre Conditions:

- login_texpadmin fixture

Input:

- name = ‘test experiment’

Steps	Expected Result	Actual Result
view the experiment	no errors, on view experiment page	

`test_experiment.`**`test_view_experiment_physician`**(*login_tphysician*)

Requirements:

- 3.1.4.2: Experiment results shall be viewable by experiment administrators and physicians.

Pre Conditions:

- login_texpadmin fixture

Input:

- name = ‘test experiment’

Steps	Expected Result	Actual Result
view the experiment	no errors, on view experiment page	

test_export module

These test cases are designed to test the ability to export data from the system

`test_export.`**`test_export_data_by_patient`**(*login_tpatient*, *test_patients*)

Requirements:

- 3.1.5: The system shall allow data exporting.
- 3.1.5.2: The system shall allow patients to export their data.
- 3.1.5.4: The system shall provide comma-separated values (.csv) file exports.

Pre Conditions:

- login_tpatient fixture

Input:

Steps	Expected Result	Actual Result
go to export page	your username is visible	
click on your user	you can see an uploaded file	
click on the first available file	file gets highlighted	
fill in the download path	no errors	
click on the export button	download starts	
manually check for the file	files exist on the system	

NOTE:

This test does not work with the latest export changes that were made. It can be run manually for verification.

`test_export.`**`test_export_data_by_physician`**(*login_tphysician*, *test_patients*)

Requirements:

- 3.1.5: The system shall allow data exporting.
- 3.1.5.3: The system shall allow physicians to export the data of their patients.
- 3.1.5.4: The system shall provide comma-separated values (.csv) file exports.

Pre Conditions:

- login_tphysician fixture

Input:

Steps	Expected Result	Actual Result
click on the export button	you see the patient	
click on the patient	you see files listed	
click on the first file	the file is highlighted	
fill in the download path	no errors	
click the export button	the download starts	
check for the file	the file exises	

NOTE:

This test does not work with the latest export changes that were made. It can be run manually for verification.

`test_export.`**`test_export_experiment_results`**(*login_texpadmin*)

Requirements:

- 3.1.5: The system shall allow data exporting.
- 3.1.5.1: The system shall allow experiment results to be exported.
- 3.1.5.4: The system shall provide comma-separated values (.csv) file exports.

Pre Conditions:

- login_texpadmin fixture

Input:

- exp_name = 'test_experiment'
- sage = '18'
- eage = '90'
- shght = '60'
- ehght = '80'
- swght = '100'
- ewght = '250'
- genders = []
- races = []
- eth = []
- loc = []

Steps	Expected Result	Actual Result
create experiemnt	no errors	
view the experiment	there are patients in the experiment	
view the data of a patient in the experiment	no errors	
click the export button	no errors	
manually check for the file	files have been downloaded	

test_login module

These tests are to verify that the accounts can be logged into

`test_login.`**`test_login_bad_pass`**`(`*logoff*`)`

Requirements:

- 3.1.1: System shall provide user authentication

Pre Conditions:

- logoff fixture

Input:

- user = 'testPhysician'
- password = '!nc0rrect'
- expected = 'Invalid login attempt.'

Expected Results:

User 'testPhysician' sees the home page after logging in

Steps	Expected Result	Actual Result
login	Invalid login	

`test_login.`**`test_login_bad_user`**`(`*logoff*`)`

Requirements:

- 3.1.1: System shall provide user authentication

Pre Conditions:

- logoff fixture

Input:

- user = 'fake'
- password = 'P@ssword10'
- expected = 'Invalid login attempt.'

Expected Results:

User 'testPhysician' sees the home page after logging in

Steps	Expected Result	Actual Result
login	Invalid login	

`test_login.`**`test_login_experiment_admin`**`(`*logoff*`)`

Requirements:

- 3.1.1: System shall provide user authentication
- 3.1.1.5: System shall allow user to log out of account

Pre Conditions:

- logoff fixture

Input:

- user = 'testExpAdmin'
- password = 'P@ssword10'

Expected Results:

User 'testPhysician' sees the home page after logging in

Steps	Expected Result	Actual Result
login	Login Success!	

`test_login.`**`test_login_patient`**`(`*logoff, test_patients*`)`

Requirements:

- 3.1.1: System shall provide user authentication
- 3.1.1.5: System shall allow user to log out of account

Pre Conditions:

- logoff fixture

Input:

- user = 'testPatient'
- password = 'P@ssword10'

Expected Results:

User 'testPhysician' sees the home page after logging in

Steps	Expected Result	Actual Result
login	Login Success!	

`test_login.`**`test_login_physician`**`(`*logoff*`)`

Requirements:

- 3.1.1: System shall provide user authentication
- 3.1.1.5: System shall allow user to log out of account

Pre Conditions:

- logoff fixture

Input:

- user = 'testPhysician'
- password = 'P@ssword10'

Expected Results:

User 'testPhysician' sees the home page after logging in

Steps	Expected Result	Actual Result
login	Login Success!	

`test_login.`**`test_login_system_admin`**`(`*logoff*`)`

Requirements:

- 3.1.1: System shall provide user authentication
- 3.1.1.5: System shall allow user to log out of account

Pre Conditions:

- logoff fixture

Input:

- user = 'fitadmin'
- password = 'Password1!'

Expected Results:

User 'testPhysician' sees the home page after logging in

Steps	Expected Result	Actual Result
login	Login Success!	

test_password module

These tests are to make sure password functionality meets the requirements

`test_password.`**`test_password_case`**`(`*logoff*`)`

Requirements:

- 3.1.1.4: System requires a minimum security criteria
- 3.1.1.4.2: Password shall contain at least 1 upper and 1 lower case character

Pre Conditions:

- logoff fixture

Input:

- user = ‘pfry’
- pwd = ‘p@ssword10‘
- email = ‘pfry@futurama.com‘
- first_name = ‘Fry’
- last_name = ‘Phillip’
- address = ‘304 Wherever Street, New New York City, New New York’
- phone_number = ‘123-456-7890’

Steps	Expected Result	Actual Result
request account	password case error	

test_password.**test_password_digit**(*logoff*)

Requirements:

- 3.1.1.4: System requires a minimum security criteria
- 3.1.1.4.3: Password shall contain at least 1 number

Pre Conditions:

- logoff fixture

Input:

- user = ‘pfry’
- pwd = ‘P@ssworddd‘
- email = ‘pfry@futurama.com‘
- first_name = ‘Fry’
- last_name = ‘Phillip’
- address = ‘304 Wherever Street, New New York City, New New York’
- phone_number = ‘123-456-7890’

Steps	Expected Result	Actual Result
request account	password digit error	

test_password.**test_password_length**(*logoff*)

Requirements:

- 3.1.1.4: System requires a minimum security criteria
- 3.1.1.4.1: Password must be at least 10 characters

Pre Conditions:

- logoff fixture

Input:

- user = ‘pfry’
- pwd = ‘P@ssword1‘
- email = ‘pfry@futurama.com‘

- first_name = 'Fry'
- last_name = 'Phillip'
- address = '304 Wherever Street, New New York City, New New York'
- phone_number = '123-456-7890'

Steps	Expected Result	Actual Result
request account	password length error	

test_password.**test_password_special_char**(*logoff*)

Requirements:

- 3.1.1.4: System requires a minimum security criteria
- 3.1.1.4.4: Password shall contain at least 1 special character

Pre Conditions:

- logoff fixture

Input:

- user = 'pfry'
- pwd = 'Password10'
- email = 'pfry@futurama.com'
- first_name = 'Fry'
- last_name = 'Phillip'
- address = '304 Wherever Street, New New York City, New New York'
- phone_number = '123-456-7890'

Steps	Expected Result	Actual Result
request account	password digit error	

test_password.**test_reset_password**(*login_sysadmin*)

Requirements:

- 3.1.1.1.4.3: System Administrators can reset passwords

Pre Conditions:

- login_sysadmin fixture

Input:

- user = 'testPhysician2'
- username = 'test physician2'
- pwd = 'Password1!'

Steps	Expected Result	Actual Result
Set users password to new password	No errors	
Login with the new password	User can login	

test_security module

test_security.**test_experiment_creation_security**(*login_tpatient*)

Requirements:

- 3.1.4.1: Experiments shall only be created by experiment admins

Pre Conditions:

- login_tpatient fixture

Input:

- physician = ‘testPhysician’
- physicianpass = ‘P@ssword10‘
- admin = ‘fitadmin’
- adminpass = ‘Password1!’

Steps	Expected Result	Actual Result
look for the create experiment button	there is no create experiment button	
logoff and login as testPhysician	no errors	
look for the create experiment button	there is no create experiment button	
logoff and login as fitadmin	no errors	
look for the create experiment button	there is no create experiment button	

`test_security.`**`test_patient_only_sees_his_data`**(*test_patients*)

Requirements:

- 3.1.1.1.2.4: Physician can only view his patients data

Pre Conditions:

- test_patients fixture

Input:

- file = path.abspath(‘./Data/BasisPeak/bodymetrics_2012-01-30T00-00-00_2016-01-18T04-51-00.csv’)
- fyear = ‘2015’
- fmonth = ‘June’
- fday = ‘23’
- tyear = ‘2015’
- tmonth = ‘June’
- tday = ‘23’
- device = ‘basis’
- activities = []

Steps	Expected Result	Actual Result
login as the test patient 2	your at the home page	
upload a test data file	no errors	
logoff	back at the login screen	
login as test patient 1	your at the home page	
click the view data button	you see all the files that have been uploaded	
look for the file that was just uploaded	you do not see the file you uploaded in step 2	

`test_security.`**`test_physician_to_patient`**(*logoff*)

Requirements:

- 3.1.1.1.2.4: Physician can only view his patients data

Pre Conditions:

- logoff fixture

Input:

- physician = ‘testPhysician3’
- phys_pass = ‘P@ssword10‘
- phys_email = ‘testphysician3@uah.com‘
- phys_add = ‘here’
- phys_fname = ‘test’
- phys_lname = ‘physicianthree’
- phys_phone = ‘123-456-7890’
- patient = ‘testPatient5’
- pat_pass = ‘P@ssword10‘
- bday = ‘April 1, 1987’
- loc = ‘Alabama’
- wght = ‘123’
- hght = ‘123’
- gen = ‘male’
- race = ‘white’
- eth = ‘non_hispanic’
- test_physician = ‘testPhysician’
- test_phys_pass = ‘P@ssword10‘

Steps	Expected Result	Actual Result
create physician	no errors	
create patient for physician	no errors	
login as testPhysician	login success	
check for patient in management page	patient not listed	

`test_security.`**`test_system_admin_user_management`**(*login_tpatient*)

Requirements:

- 3.1.6.4: Only system admin can enable accounts
- 3.1.6.5: Only system admin can delete accounts

Pre Conditions:

- login_tpatient fixture

Input:

- physician = ‘testPhysician’
- physicianpass = ‘P@ssword10‘
- expadmin = ‘testExpAdmin’
- expadminpass = ‘P@ssword10‘

Steps	Expected Result	Actual Result
look for the manage users button	manage users button does not exist	
logoff and login as the testPhysician	no errors	
look for the manage users button	manage users button does not exist	
logoff and login as the testExpAdmin	no errors	
look for the manage users button	manage users button does not exist	

test_upload module

`test_upload.`**`test_basis_data_upload`**`(`*login_tpatient*`)`

Requirements:

- 3.2.4.2: System shall support Basis Peak data

Pre Conditions:

- login_tpatient fixture

Input:

- activities = []
- file = path.abspath('./Data/BasisPeak/bodymetrics_2012-01-30T00-00-00_2016-01-18T04-51-00.csv')

Steps	Expected Result	Actual Result
upload file	no errors, and returned to /PatientData	

`test_upload.`**`test_mband_data_upload`**`(`*login_tpatient*`)`

Requirements:

- 3.2.4.3: System shall support Microsoft Band data

Pre Conditions:

- login_tpatient fixture

Input:

- activities = []
- file = path.abspath('./Data/MBand/')

Steps	Expected Result	Actual Result
upload file	no errors, and returned to /PatientData	

`test_upload.`**`test_single_file_multi_activity`**`(`*login_tpatient*`)`

Requirements:

- 3.1.2: System shall process medical data
- 3.1.2.1: System shall process one or more medical data file at a time
- 3.1.2.2: System shall provide ability to assign activity
- 3.1.2.3: There will be an interface to select data and activities
- 3.1.2.5: System shall process CSV files

Pre Conditions:

- login_tpatient fixture

Input:

- activities

- activity 1
  - ∗‘type’:’Home Eating’,
  - ∗‘year’:2015,
  - ∗‘month’:’June’,
  - ∗‘day’:24,
  - ∗‘startTime’: ‘11:00 PM’,
  - ∗‘endTime’: ‘11:30 PM’},
- activity 2
  - ∗‘type’:’Home Cooking’,
  - ∗‘year’:2015,
  - ∗‘month’:’June’,
  - ∗‘day’:24,
  - ∗‘startTime’: ‘12:00 PM’,
  - ∗‘endTime’: ‘12:30 PM’

- file = path.abspath(‘./Data/Zephyr/ZephyrTestData/2015_06_24__23_05_14_Accel.csv’)

Steps	Expected Result	Actual Result
upload file	no errors, and returned to /PatientData	

`test_upload.`**`test_single_file_no_activity`**(*login_tpatient*)

Requirements:

- 3.1.2: System shall process medical data
- 3.1.2.1: System shall process one or more medical data file at a time
- 3.1.2.3: There will be an interface to select data and activities
- 3.1.2.5: System shall process CSV files

Pre Conditions:

- login_tpatient fixture

Input:

- activities = []
- file = path.abspath(‘./Data/Zephyr/ZephyrTestData/2015_06_24__23_05_14_Accel.csv’)

Steps	Expected Result	Actual Result
upload file	no errors, and returned to /PatientData	

`test_upload.`**`test_single_file_one_activity`**(*login_tpatient*)

Requirements:

- 3.1.2: System shall process medical data
- 3.1.2.1: System shall process one or more medical data file at a time
- 3.1.2.2: System shall provide ability to assign activity
- 3.1.2.3: There will be an interface to select data and activities
- 3.1.2.5: System shall process CSV files

Pre Conditions:

- login_tpatient fixture

Input:

- activities = [{'type':'Home Eating', 'year':2015, 'month':'June', 'day':24, 'startTime': '11:00 PM', 'endTime': '11:30 PM'}]
- file = path.abspath('./Data/Zephyr/ZephyrTestData/2015_06_24__23_05_14_Accel.csv')

Steps	Expected Result	Actual Result
upload file	no errors, and returned to /PatientData	

`test_upload.`**`test_zephyr_data_upload`**`(`*login_tpatient*`)`

Requirements:

- 3.2.4.1: System shall support Zephyr data

Pre Conditions:

- login_tpatient fixture

Input:

- activities = []
- file = path.abspath('./Data/Zephyr/ZephyrTestData/2015_06_24__23_05_14_BB.dat')

Steps	Expected Result	Actual Result
upload file	no errors, and returned to /PatientData	

test_username module

These tests are to make sure the username functions correctly

`test_username.`**`test_username_cannot_be_copied`**`(`*logoff*`)`

Requirements:

- 3.1.1.2: Usernames must be unique
- 3.1.1.3: The system shall assign unique ids to each created patient.
- 3.1.1.2.1: Users cannot change username to an already existing username

Pre Conditions:

- logoff fixture

Input:

- user = 'copy'
- pwd = 'P@ssword10'
- email = 'copy@copy.com'
- first_name = 'Fry'
- last_name = 'Phillip'
- address = '304 Wherever Street, New New York City, New New York'
- phone_number = '123-456-7890'
- user_2 = 'copy1'

- email_2 = 'copy2@copy.com'

Steps	Expected Result	Actual Result
Request physician account	No errors	
Request same physician account	Username taken error	
Request different account	No errors	

conftest module

conftest.**login_sysadmin**()

conftest.**login_texpadmin**()

conftest.**login_tpatient**()

conftest.**login_tphysician**()

conftest.**logoff**()

conftest.**pre_existing_users**(*request*)

conftest.**pytest_addoption**(*parser*)

conftest.**test_patients**()

WebUI package

Submodules

WebUI.WebUI module These are the web user interface utilities used to drive the web tests.

class WebUI.WebUI.**WebUI**(*baseurl='http://localhost:25396/'*)

This class keeps track of the web session. It uses Firefox as the web driver. Its baseurl is the start page of the software. By default it is the localhost on port 25396.

approve_account(*user*)

Logs in as the system admin and approves an account with the given username

Parameters **username** – Username of the account to approve

Returns

check_create_patient()

Validates that the patient creation was successful

Returns

check_login(*user='', t=5*)

This will verify you are logged in. It checks for the manage account link that only appears if you are logged in.

Parameters

- **t** – A timeout in seconds to wait for the management link to appear
- **user** – Check to see if this is the user that is logged in

Returns

check_logoff(*t=10*)

This will check that there is no logged in user for the session

Parameters **t** – time in seconds to wait for the login button to appear. The default is 10.

Returns

`check_request_account()`

Validates that the account request was successful

Returns

`confirm_account_info`(*email=''*, *address=''*, *phonenum=''*)

This procedure will confirm account information of whatever information is provided.

Parameters

- **email** – the expected email address
- **address** – the expected home address
- **phonenum** – the expected phone number

Returns

`create_admin`(*user*, *pwd*, *email*)

`create_experiment`(*expName*, *startAge*, *endAge*, *startHght*, *endHght*, *startWght*, *endWght*, *genders*, *races*, *ethnicities*, *locations*)

`create_patient`(*phy*, *phy_pass*, *user*, *pwd*, *byear*, *bmonth*, *bday*, *location*, *weight*, *height*, *gender*, *race*, *ethnicity*)

This will create a patient account. It expects the physician to be logged in when called.

Parameters

- **user** –
- **pwd** –

Returns

`delete_account`(*user*)

This logs in as the system admin and deletes the specified user

Parameters **user** – username of the account to delete

Returns

`delete_all_accounts()`

`delete_experiment`(*name*)

`delete_patient`(*phy*, *phy_pass*, *user*)

This logs in as the physician and deletes the specified user

Parameters

- **phy** – name of the physician that the patient belongs to
- **phy_pass** – password of the physician
- **user** – username of the account to delete

Returns

`driver` **= <selenium.webdriver.firefox.webdriver.WebDriver (session=”aad180c5-1dfe-4bf8-bb84-7f1001f62d07”)>**

`export_data`(*path*)

`get_page()`

Helper procedure to return the current web pages source useful to validate responses on the page

Returns

`go_home`()

Navigates the web session back to the base url

Returns

`login`(*user*, *pwd*)

This will make sure you are logged in as the user specified. It will log you out if you are already logged in.

Parameters

- **user** – Username to log in as
- **pwd** – Password for the given username

Returns

`logoff`()

This will log the logged in user off.

Returns

`request_account`(*account_type*, *user*, *pwd*, *email*, *first_name*, *last_name*, *address*, *phone_number*)

This will fill out a request for an account. To request an account you must be logged out therefore this procedure will make sure you are logged out

Parameters

- **account_type** – The type of account to request. can either be Physician or Exp Admin
- **user** – The username of the new account
- **pwd** – The password of the new account
- **email** – The email address of the new account
- **first_name** – The first name of the user
- **last_name** – The last name of the user
- **address** – The home address of the user
- **phone_number** – The phone number of the user

Returns

`reset_user_password`(*user*, *new_pwd*)

This procedure will make the system admin reset an accounts password

Parameters

- **user** – username of the account to touch
- **new_pwd** – the new password to give the account

Returns

`set_account_info`(*pwd=’‘*, *current_pwd=’‘*, *email=’‘*, *address=’‘*, *phonenum=’‘*)

This procedure will set the account info to any info provided if provided It assumes the user is already logged in

Parameters

- **pwd** – the new password desired
- **current_pwd** – the current password. needed to change passwords
- **email** – the new email address

- **address** – the new home address
- **phonenum** – the new phone number

Returns

`upload_files` (*file*, *fyear*, *fmonth*, *fday*, *tyear*, *tmonth*, *tday*, *device*, *activities*)

This will uplaod files for a patient with the given activities

Parameters

- **file** – Relative paths to files to be uploaded
- **fyear** – The year the file started collecting
- **fmonth** – The month the file started collecting
- **fday** – The day the file started collecting
- **tyear** – The year the file stopped collecting
- **tmonth** – The month the file stopped collecting
- **tday** – The day the file stopped collecting
- **device** – The device the file came from. must be 'zephyr', 'basis', or 'band'
- **activities** – A list of dictionaries. dicts look like: {'type':'Home Eating', 'year':1920, 'month':'January', 'day':3, 'startTime': '11:00 AM', 'endTime': '11:30 AM'}

Returns

`view_data` ()

`view_experiment` (*name*)

Module contents

2.4.2 Manual Tests

Table of Contents

Test Preparations

Hardware Preparations

You need a computer with access to the UAHealth server.

Software Preparations

You need the latest Chrome or Firefox version installed on the computer.

Other Preparations

It is expected that you will have run the automated tests before executing these tests. Some of the users and data depend on the automation have run correctly.

Test Details

Multiple File Upload Test

Requirements:

- 3.1.2.1: The system shall process multiple files at the same time

Pre Conditions:

- You must be logged in as the testPatient user using these credentials
- Username: 'testPatient'
- Password: 'P@ssword10'
- You must have access to the data files in the project folder here: ./med656/Tests/Data

Input:

Test Procedure:

Steps	Expected Result	Actual Result
click the 'Upload Data' link in top right corner of page	Taken to the upload page	
click the 'BROWSE' button	Windows file select wizard opens	
navigate to ./med656/Tests/Data/Zephyr/ZephyrTestData/	Files are visible	
highlight the ...Accel.csv and ...BB.csv files by holding ctrl and clicking	Files are highlighted	
click the 'chose' button on the wizard	File names are in the upload page now	
fill in date 6-24-2015 for both from date and to date	Dates are visible on the page	
click on zephyr radio button	Radio button is highlighted	
click "SUBMIT" button	No errors are displayed	

User Privacy Test

Requirements:

- 3.1.1.1.1.2: The system shall prevent any personal identifiable information from being available for a Patient

Pre Conditions:

- The user must be logged off
- A physician named 'testPhysician' must be created
- To check if this user is created login as user 'fitadmin' password 'Password1!'
- Click the 'Manage Accounts' button at the top of the page
- Select 'Physicians' from the Roles dropdown and look for 'testPhysician'
- If physician 'testPhysician' does not exist
  - Follow the steps here (*Physician Account*) to create a physician account with the following information
  - first name = test
  - last name = test
  - username = testPhysician
  - password = 'P@ssword10'
  - email = testphysician@test.com
  - address = here
  - phone number = 123-456-7890
  - reason = test

Input:

- user = 'testPatientPrivacy'
- password = 'P@ssword10'

Test Procedure:

Steps	Expected Result	Actual Result
login as 'testPhysician' with password 'P@ssword10'	Login Success!	
click 'create patient' button	Taken to the create patient page	
verify that none of the information requested is personally identifiable	None of the info is personally identifiable	

Physician Graph View

Requirements:

- 3.1.1.1.2.3: The system shall allow the physician user to view their patient's data graphically.
- 3.1.7: The system shall provide a user interface for displaying medical data in the system.
- 3.1.7.16: The system shall display summary heart rate data on a chart from the Zephyr and BasisPeak.
- 3.1.7.16.1: The chart showing summary heart rate data from the Zephyr and BasisPeak will also show heart rate data from the Microsoft Band.

Pre Conditions:

- The user must be logged off
- A physician named 'testPhysician' must be created
- To check if this user is created login as user 'fitadmin' password 'Password1!'
- Click the 'Manage Accounts' button at the top of the page
- Select 'Physicians' from the Roles dropdown and look for 'testPhysician'
- If physician 'testPhysician' does not exist
  - Follow the steps here *Physician Account* to create a physician account with the following information
  - first name = test
  - last name = test
  - username = testPhysician
  - password = 'P@ssword10'
  - email = testphysician@test.com
  - address = here
  - phone number = 123-456-7890
  - reason = test
- 'testPhysician' must have a patient 'testPatientGraphView'
  - If patient 'testPatient' does not exist
    - Follow the steps here *Patient Account* to create patient 'testPatientGraphView' with to following information
    - user = 'testPatient'
    - pwd = 'P@ssword10'
    - birthday = '3 March, 1954'
    - location = 'Alabama'
    - weight = '200'
    - height = '72'
    - gender = 'male'
    - race = 'white'
    - ethnicity = 'non_hispanic'

Input:

- user = 'testPatientPrivacy'
- password = 'P@ssword10'

Test Procedure:

Steps	Expected Result	Actual Result
Login as 'testPatient' with password 'P@ssword10'	Login Success!	
Follow user guide instructions *here* to upload the following files from this folder <project directory>/med656/Tests/Data/: - BasisPeak/bodymetrics_simple.csv - Zephyr/ZephyrTestData/2015_06_24__23_05_14_Summary.cvs - Band/data.csv	There are no errors while uploading data	
Logout and log back in as 'testPhysician' with password 'P@ssword10'	Login Success!	
Follow instructions *here* to view the 'testPatient' data that was just uploaded.	The summary heart rate graph can be seen for the Zephyr, BasisPeak, and Band	

Database Verification Test

Requirements:

- 3.1.3: The system shall connect to a database
- 3.2.2: The system shall use a SQL database

Pre Conditions

Inputs

Test Procedure

Steps	Expected Result	Actual Result
Login locally or remote in to the UAHealt Bit Vault server desktop	you are at the windows desktop	
Open Microsoft SQL Database Manager	you can see the uahbitvault database	

Internet Access Verification Test

Requirements:

- 3.2.3: The system shall be connected to a network with internet access

Pre Conditions

Inputs

Test Procedure

Steps	Expected Result	Actual Result
Login to the machine the system is running on	You are at the Windows Desktop	
Open a command prompt	You are presented with the Windows prompt	
Execute "ping www.google.com"	Pings respond with 100% pass	

OS Test

Requirements:

- 3.2.1: The system shall run on Windows Server Operating System

Pre Conditions:

Input:

Test Procedure:

Steps	Expected Result	Actual Result
Login to the machine the system is running on	You are at the Windows Desktop	
Open a command prompt	You are presented with the Windows prompt	
Execute “systeminfo”	The ‘OS Name:’ is ‘Microsoft Windows Server’	

CHAPTER

THREE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX