

System and Unit Test Report

Product: NoteQuest

Team name: Project NOTE

Date: 7/17/2020

Test Name Format:

- <route> -> <Request Type>: <testing purpose> | <expected outcome>

Disclaimer:

We are mostly testing server routes and database integration

upload_file -> POST: notes | PASS

User Stories:

- As a student, I want to be able to upload notes - Sprint 1
- As a student, I want to be able to create a folder for my class - Sprint 2

Frontend Scenario:

- Start app; Login; Click POST
 - Subject = <CSE>
 - Class No. = <115A>
 - Professor = <Jullig>
 - UPLOAD FILE = <insert of a file of your choice>
- Hit the *POST NOTES* button
- User should receive a browser prompt stating that the file has been uploaded
- To check if uploaded head to that class and professor under *browse*

Backend Scenario:

- Send a post request to: localhost:8080/upload_file
- Include the following keys:
 - class = <CSE115A>
 - prof = <Jullig>
 - avatar = <uploaded file>
- Client should be returned a 200 response, while also uploading the file successfully

upload_file -> POST: notes | FAIL

User Stories:

- As a student, I want to be able to upload notes - Sprint 1

- As a student, I want to be able to access other students notes - Sprint 2
- As a student, I want to be able to create a folder for my class - Sprint 2

Frontend Scenario:

- Start app; Login; Click POST
 - Subject = <CSE>
 - Class No. = <115A>
 - Professor = <Jullig>
 - UPLOAD FILE = <insert of a file of your choice>
 - Now leave out any one of the previous fields
- User should receive a browser prompt stating they are missing a field

Backend Scenario:

- Send a post request to: localhost:8080/upload_file
- Include the following keys:
 - class = <CSE115A>
 - prof = <Jullig>
 - avatar = <uploaded file>
 - Omit one of the above fields
- Client should be returned a 400 response, while not uploading the file at all

send_data -> GET: notes | PASS

User Stories:

- As a student, I want to be able to query information about the notes that are uploaded in the app- Sprint 1
- As a student, I want to be able to access other students notes - Sprint 2

Frontend Scenario:

- Start app; Login; Click browse; enter:
 - Class = <CSE130>
 - Instr = <Alvaro>
- User should be able to see all the notes for that class and professor

Backend Scenario:

- Send a get request to: localhost:8080/send_data and assign the keys
 - mode = <file>
 - class = <CSE130>
 - prof = <Alvaro>
- Client should now see all the files for this class and professor

send_data -> GET: notes | FAIL

User Stories:

- As a student, I want to be able to query information about the notes that are uploaded in the app- Sprint 1
- As a student, I want to be able to access other students notes - Sprint 2

Backend Scenario:

- Send a get request to: localhost:8080/send_data and assign the keys
 - mode = <file>
- Client should not receive any data because the class and professor wasn't specified

send_data -> GET: prof | PASS

User Stories:

- As a student, I want to be able to query information about the notes that are uploaded in the app- Sprint 1
- As a student, I want to be able to access other students notes - Sprint 2

Frontend Scenario:

- Start app; Login; Click browse; enter:
 - Class: <CSE130>
- User should be able to see all the professors that are associated with that specific class when they click on *instr*

Backend Scenario:

- Send a get request to: localhost:8080/send_data and assign the keys
 - mode = <prof>
 - class = <CSE130>
- Client should be get returned a list of all the professors for that specific class

send_data -> GET: prof | FAIL

User Stories:

- As a student, I want to be able to query information about the notes that are uploaded in the app- Sprint 1
- As a student, I want to be able to access other students notes - Sprint 2

Backend Scenario:

- Send a get request to: localhost:8080/send_data and assign the keys
 - mode = <prof>
 - class = <leave empty>
 - Client should be returned a 400 error; returned a message stating bad input
-

send_data -> GET: class| PASS

User Stories:

- As a student, I want to be able to query information about the notes that are uploaded in the app- Sprint 1
- As a student, I want to be able to access other students notes - Sprint 2

Frontend Scenario:

- Start app; Login; Click browse;
 - All classes are shown which makes the request
- User should be able to see all the professors that are associated with that specific class when they click on *instr*

Backend Scenario:

- Send a get request to: localhost:8080/send_data and assign the keys
 - mode = <class>
 - Client should be get returned a list of all the class
-

send_data -> GET: class| FAIL

User Stories:

- As a student, I want to be able to query information about the notes that are uploaded in the app- Sprint 1
- As a student, I want to be able to access other students notes - Sprint 2

Backend Scenario:

- Send a get request to: localhost:8080/send_data and assign the keys
 - mode = <some invalid mode>
 - Client should be get returned with a 400 error and a message stating bad mode
-

send_class_codes -> GET: class codes | PASS

User Stories:

- As a student, I want to upload my notes to a specific class and professor- Sprint 2

Frontend Scenario:

- Start app; Login; Click POST;
- Click on Subject
- User should be able to see all the class codes at UCSC

Backend Scenario:

- Send a get request to: localhost:8080/send_class_codes
 - Client should receive a list of all the class codes at UCSC
-

verify_email-> POST: email | PASS

User Stories:

- As a student, I want to be able access other students notes- Sprint 1 & 2

Frontend Scenario:

- Click signup; fill out the fields
 - Email = <username>@ucsc.edu
 - Password = <your account password>
 - Name = <your name>
- Click the sign up button; prompted with account created and verify your email

Backend Scenario:

- Send a POST request to : localhost:8080/verify_email
 - Keys:
 - email = <username>@ucsc.edu (valid ucsc.edu email)
 - The client is returned with a 200 status code and a message stating good email
-

verify_email-> POST: forbidden email | FAIL

User Stories:

- As a student, I want to be able access other students notes- Sprint 1 & 2

Frontend Scenario:

- Click signup; fill out the fields

- Email = <username>@<some other domain>
- Password = <your account password>
- Name = <your name>
- Click the sign up button; prompt back with “please enter a valid ucsc email”

Backend Scenario:

- Send a POST request to : localhost:8080/verify_email
 - Keys:
 - email = <username>@<some other domain>(non ucsc email)
- The client is returned with a 403 status code and a message stating non UCSC email is sent

verify_email-> POST: no email | FAIL

User Stories:

- *As a student, I want to be able access other students notes- Sprint 1 & 2*

Frontend Scenario:

- Click signup; fill out the fields
 - Email = <no email entered>
 - Password = <your account password>
 - Name = <your name>
- Click the sign up button; prompt back with “please enter an email address”

Backend Scenario:

- Send a POST request to : localhost:8080/verify_email
 - Keys:
 - email = <NULL>
- The client is returned with a 400 status code and a message stating a bad request is sent with no email

rating-> GET: rating | PASS

Frontend Scenario:

- Click Notes; Navigate to the class and professor of your choice
- Look at a note
- If you like it press the like button

- Refresh the page
- Click Notes; Navigate back to the same class and professor of your choice
- Check if the notes still exist

Backend Scenario:

- Send a GET request to : localhost:8080/get_rating with query parameters as follows
 - User : username of the user rating the notes
 - Mode: upvote or downvote from user
 - Class: class the notes belongs to
 - Prof : professor of the class the notes belongs to
 - Url : url of the notes being rated
 - The client is returned with a 200 status after the document is rated
-