

System and Unit Test Report
GroupThink
GroupThink Development Team
11/23/2016

System Test Scenarios:

SPRINT1:

1. *As a user I want a simple UI for easy readability*
2. *As a user I want to login so that I can access my class*
3. *As a user, I want the ability to create a project so that other people in the class can view my project*
4. *As a user, I want a profile so I can edit my information and it is available to other users*
5. *As a user I want to be able to view all current projects*

Covers user stories 1-5

Scenario:

1. Go to <http://groupthink115.pythonanywhere.com/> on any web browser
2. On the right side of the screen, click **Log in**
3. Click **Sign Up**
4. Fill out the following:
 - a. First name = <Test>
 - b. Last name = <Person>
 - c. Email = <testing@example.com>
 - d. Username = <test_person>
 - e. Password = <test1234>
 - f. Confirm password = <test1234>
 - g. Phone = <12345678900>
 - h. Bio = <Irrelevant>
 - i. Skills = <C++>
 - i. Click +
 - j. Skills = <Python>
 - k. Coursework = <CS115>
 - l. Picture = <none>
5. Click submit
6. Click on + **My Courses**. The user should not be enrolled in any courses at initial sign-up time.
7. On the right side of the screen, click **Join Course**

8. Course id = <I>
9. Click submit
 - a. If the entered course does not exist, display error to user and prompt for an alternate Course id.
10. Click on **View CS115** ->
11. User should be able to see a list of all projects in the class from this page
12. Click on + **Post Project**
13. Fill out the following:
 - a. Project Name = <Test Project>
 - b. Current Members = <testing@example.com>
 - c. Project Info = <This is a test project>
 - d. Needed Skills = <Python>
 - e. Click Accepting Members
14. Click submit
 - a. The new project should be created and you will be redirected to a page that displays the new project.
15. Click View All Projects
 - a. The new project should will now be added to the list of projects in that class and can be viewed by all people enrolled in that class.
16. Click +**View Course Members**
17. Find Test Person and click **Profile**
18. User should see a profile for themselves

SPRINT 2:

1. As a project owner, i would like to edit a project
2. As a user I would like to be able to distinguish between open and closed projects.

Covers user stories 1-2

Scenario:

1. Go to <http://grouphink115.pythonanywhere.com/> on any web browser
2. On the right side of the screen, click **Log in**
3. Fill out the following
 - a. E-mail = <testing@example.com>
 - b. Password = <test1234>
4. Click **Log In**
5. Find **Test Project** in the list under My Projects, and click **Edit**
6. Change the following Fields
 - a. Project Name = <Changed Name>
 - b. On current members, click +

- c. *Current members = test@test.com*
- d. *Project Info = <I changed the name of this project>*
- e. *Uncheck Accepting Members*
- 7. Click **Submit**
- 8. *User should see the new information about the project, including the name, info, current members, skills needed, and whether or not the project is currently accepting members*
- 9. Click **View All Projects**
 - a. *If a project is still accepting members, 'Yes' will be displayed to the right for each project.*

SPRINT 3:

- 1. *As a user I want to view all member profiles in my class*
- 2. *As a project owner, I would like to see matching students so I can assemble a team.*
- 3. *As a user I want to be able to be in multiple classes so that I can use GroupThink with more than one group project.*
- 4. *As a user, I would like to view other user profiles so that I can evaluate possible teammates.*
- 5. *As a user, I would like GroupThink to be a hosted website, so that I can access the site from anywhere.*
- 6. *As a professor (admin) I would like a class page that is private to my students, so that only my current students can access projects/profiles.*
- 7. *As a professor (admin) I would like to view statistics on class participation, to facilitate project and group management.*

Covers user stories 1-5

Scenario:

- 1. Go to <http://groupthink115.pythonanywhere.com/> on any web browser
- 2. On the right side of the screen, click **Log in**
- 3. Fill out the following
 - a. E-mail = [<testing@example.com>](mailto:testing@example.com)
 - b. Password = [<test1234>](mailto:testing@example.com)
- 4. Click **Log In**
- 5. On the left side, under My Courses
- 6. Click on **CS115 ->**
- 7. Click on **CS115 members**
- 8. *User should see a list of all students enrolled in this course*
- 9. *If you and a member have taken the same coursework, a statement will appear below their username.*
- 10. Click on **Profile** for any student
- 11. Clicking on **Bio** should show a short biography of the user

12. Clicking on **Projects** should display a list of projects that user is in
13. Clicking on **Skills** should display a user-defined list of skills that user entered

Covers user stories 6-7

Scenario:

1. Go to <http://grouphink115.pythonanywhere.com/> on any web browser
2. On the right side of the screen, click **Log in**
3. Fill out the following
 - a. E-mail = <testing@example.com>
 - b. Password = <test1234>
4. Click **Log In**
5. Under My Courses click **View all**
6. Click **+Create Course**
7. Fill out the following:
 - a. What is the name of the course? = <Example Class>
 - b. Please enter the course ID: = <Test1>
 - c. Please enter a brief course description: = <description>
 - d. Please enter the current term = <Fall 2016>
8. Click **Submit**
9. The user will be redirected to *Enrolled Courses* page, where the new course will be listed at the bottom.
10. Find *Example Class* in the list of courses and on the right side, click **Statistics**
11. On this page, the user will be able to see enrolled students, the current number of projects, and a list of the students that are currently not in a project.

System Test Scenarios:

Testing for each function:

```
def get_user_name_from_email(email):
```

```
    This function is thoroughly tested in test.py
```

- Tests that email returns correct name
- Tests what happens when there is in the case of an incorrect email

```
def index():
```

```
    if user is logged in, they will see their projects and classes
```

```
    if user is not logged in, they will see the login page
```

```
def edit_course():
```

```
    This will edit course information
```

```
    Editing the course name will change the name of the course
```

- If this is left blank, user will be prompted to return a value

- Otherwise, the name is changed

When creating a course, the course ID will be unique

- Will return an error if not unique

Editing the course info will edit the course description

- The course info field is OK to be blank

Admin email will be the admins email

- Won't be able to change this at all

Editing term changes the listed term

- Leaving the field blank prompts user to Please enter a term
- Otherwise anything can be entered for the term

def enrolled_courses():

This will display a list of courses that the user is currently enrolled in

If the user is not logged in, there are no courses to display

def join():

This will have the user enter a course ID and enroll the user in a course with the corresponding ID

- If the user enters nothing, they will be prompted to "Enter a value"
- If the user enters an ID that is not in the database, they will get a "Course not found" error
- If they enter a correct ID, they will be enrolled in the course

def project_list():

This will display a list of projects for a given course, it returns a dictionary of projects and associated info for a given course id

If the user tries to view projects and is in multiple courses, user must first select a course

- If they do not, they get a "No course selected" error
- Otherwise, this passes

Then, a list of projects is displayed based on the course_id

- If no projects, display no projects
- If some projects, display them

def project():

(in default.py)

Error cause if invalid course_id or project_id is entered arguments (see "Known Problems")

If only the course_id is passed and no project_id is present, the user will be redirected to the project list for that course.

Project will return a database object from the project table, based on what arguments are passed.

Course returns a database object from the course table.

Students returns all students enrolled in the given course object.

Matching Students creates a list for student names

(in project.html)

If the user is viewing a project they created, an 'Edit' button will appear on the top right of the project display.

- Clicking the 'Edit' button will take the user to edit_project();
- If a project is edited, the "Last Edited" time at the bottom of the project display will be updated appropriately.

The project display will show all updated information pulled from the project database table including: Project Name, Project Info, Current Members, Needed

Skills, if the project is accepting members or not, and recommended Members IF they exist.

-User names will be displayed under the Recommended Members if a user currently enrolled in this course has listed a 'skill' on their profile that is the same as a 'skill' needed for this project.

def member_validation(form):

This will make sure a user is in a database

If only a single user is passed in, we check that the email is in the database/enrolled

- If not, return that user does not exist/is not enrolled
- Otherwise, this passes

If multiple users are passed in, check that emails are in database/enrolled

- If not, return a list of problem emails
- Otherwise, this passes

def edit_project():

Changing the project name will change the name of the project

- Leaving the field blank will return an error that says "Enter a value"
- Entering anything else will update the name of the project

Editing the current team members will update the table

- Leaving the field blank will return an error "Add yourself to current members!"
- Adding a single email that is not in the database will return "name@domain.x does not exist"
- Adding a single email that is not enrolled in the course will return an error "name@domain.x is not enrolled"
- Adding multiple emails that do not exist will return a list of emails that do not exist
- Adding multiple emails that do are not enrolled will return a list of emails that do not exist
- Adding a mixture of emails that exist and do not exist will return a list of emails that do not exist
- Adding emails that are in the course will pass
- Removing an email will remove them from the project, but not in the database, this is a bug

Changing the project description will change the information about the project

- Leaving the field blank will return an error "Enter a value"
- Entering anything else will update the project info

Editing the skills needed will change skills needed

- Leaving the field blank is OK
- Adding any number of skills to the skills field is OK

Checking/unchecking the "Looking for members" box will change if the project is looking for members

- Ticking the box will update OK
- Unticking the box will update OK

Ticking the "Check to delete" box will delete the project

- Leaving it unchecked changes nothing, this is OK
- Ticking it will delete the project, however they are still listed as being in a project in the database, this is a bug (see "Known Problems")

def profile():

(in default.py)

'Current_profile' should return the current user from the auth_user database table.

args= request.args(0) should return the current user's username

-If the username does not exist (no one is logged in), you will be redirected to login page.

-Current_profile should have all the correct information from the auth_user -database (ie Name, Bio, Skills, Picture), check the information reflected on the page vs. the information in the auth_user database.

'projects' should return all projects the current_profile has created, despite what course the projects are affiliated with.

-In the project database, any project that has the current user email listed under 'project.user_email' should be returned in 'projects'. Any project that does NOT have the current user email listed under 'project.user_email' should NOT be returned in 'projects' (this includes projects that the current user is a member of, but not a PO).

(in profile.html)

Clicking on specific tab will display appropriate information based on

'current_profile'

-If the user has not entered a bio, joined any projects, or listed any skills the appropriate default message will be displayed.

Clicking on logout button will log the current user out and redirect to login page.

Clicking on edit profile button will take the user to the edit profile page with all of their existing information in their respective textbox.

-If the user edits First name, Last name Username, Phone, Bio, Past Courses, and Picture, it is reflected in their profile and in the auth_user database.

-The user cannot edit email after sign-up. (see "Known Problems").

Viewing another user's email:

-Edit Profile/logout button should now display when viewing others profiles.

-Clicking on the message button will not do anything (see "Known Problems")

-Clicking on the 'Projects' tab will list all of the projects affiliated with the user profile you are viewing, regardless of course.

def members():

(in default.py)

'Course' will return a database object from the course table according to the course_id in request.args(0)

-If no course_id is passed, you will be redirected to the enrolled_courses page.

'Members' will return a list of objects from the auth_user table.

'Rows-coursework' will return a list of members that have matching previous coursework with the current user.

(in members.html)

Clicking on a specific "profile" button will take you to that specific

Clicking on the user's "profile" button will take you to the current user's profile. (see "Known Problems").

def statistics():

This will return the statistics of the class to the admin

If no course is selected, will tell the user that they have not selected a course

Otherwise gets the course ID from the current user and outputs the course

This function will return a list of members in the course, a list of project ID's in the course, and a list of members that are not in projects