Testing Document and Specification
Test Specification

Alternate Project
Group 8
CS 451

Adam Flowerday	Lake Gatton	Elizabeth Nastoff
Thom Martin	Ryan Moody	Zunair Ahmed

Test Case ID:	1.1.1
Title:	Authenticating the User to Allow Site Access
Feature/Subfeature:	System User Validation/Accessibility to main UI
Purpose:	To ensure main site can be accessed by allowing only credentials set by Gina
Initial Conditions:	User must be on the "login" page. The page will display entry fields for username and password, both of which need to be filled with valid user information to allow access.
Test Data:	Test Data will include invalid usernames like abba1 and fake@notreal.phony and invalid passwords as in 123456789 and abcde. Valid username and test password will also be used here, but not explicitly shared for security reasons.
Test Actions:	1. Go to the main website 2. Click in the "Email" field 3. Enter username 4. Press "tab" or click on "Password" field 5. Enter password 6. Click "Login" button
Expected Results:	After Step 2, the auto placement title of "Email" should disappear and a flashing cursor should take its place. After step 4, the auto placement title of "Password" should disappear and a flashing cursor should take its place, but the "username" field should retain the entered information. Step 5 should yield "security dots" in place of the text being entered by the user, and step 6 should direct the user to the "index" or "main" program page if credentials are correct. The test should yield an "Incorrect Username/Password" message if the information entered is incorrect.

Test Case ID:	1.1.2
Title:	Forgot Password
Feature/Subfeature:	Using Email to Change Password
Purpose:	To ensure that the user is able to continue accessing the program if the password is forgotten.
Initial Conditions:	User is on the "Login" page and cannot remember the password needed to securely log on.
Test Data:	Will include submitting an incorrect username (abcd) and incorrect email (a@b.c) as well as entering the correct email address. Further testing would include verification of the email and follow-up actions such as sending a "reset password" email or embedded form for password reset.

Test Actions:	1. Click on the “forgot password” link under the input fields 2. Click in “email” input field 3. Enter email 4. Click submit 5. Click link to change password in received email 6. Click “new password” field 7. Enter new password 8. Click “confirm password” field or press tab 9. Enter confirmed password 10. Click “submit” button 11. Repeat test case ID 1.1.1 (to confirm success of 1.1.2)
Expected Results:	After step 1, the tester should be redirected to the “forgot password” html page. Once step 4 is completed a confirmation message should appear and a “password reset” email would automatically be sent to the correct email address. The tester would be redirected to a “password change” page after step 5. Step 10 would prompt a change in the database, updating the password field to the new password.

Test Case ID:	2.1.1
Title:	Addition of requests
Feature/Subfeature:	Requesting schedule according to time constraint
Purpose:	To ensure that valid users are able make requests to have their class schedules added according to their respective time constraints
Initial Conditions:	User would be on the homepage also confirming the fact that they are a authorized user
Test Data:	Will include the entry of a class schedule into a particular time slot with the instructor name and class date and time. Furthermore, a note can be added in the request to indicate a reason as to why an instructor might need a particular time slot
Test Actions:	1. From the “home” page select scheduling requests 2. After clicking on add enter the instructor name 3. Enter class name and timing with the days 4. Select “Submit” button
Expected Results:	After step 4, a thank you message should appear indicating that the request has been submitted successfully

Test Case ID:	2.1.2
Title:	Edit requests
Feature/Subfeature:	Editing the schedule after it being submitted
Purpose:	To ensure that valid users are able edit requests to have their class schedules added according to their respective time constraints
Initial Conditions:	User would be on the homepage also confirming the fact that they are a authorized user
Test Data:	Will include the editing of a class schedule into a particular time slot with the instructor name and class date and time. Furthermore, a note can be added in the request to indicate a reason as to why an instructor might need a particular time slot so that the adjustment can be made and faculty can be notified
Test Actions:	1. From the “home” page select scheduling requests 2. After clicking on edit enter the instructor name 3. Enter class name and timing with the days 4. Select “Update” button
Expected Results:	After step 4 , system should prompt a message indicating that the edit requested has been submitted successfully and is now being reviewed

Test Case ID:	2.1.3
Title:	Approve Requests
Feature/Subfeature:	Approving the requests after they have been submitted
Purpose:	To ensure that the administrators of the system can approve the edited and added requests
Initial Conditions:	The administrator will have access to this from the homepage confirming that they are a authorized personnel
Test Data:	Will include the approving of a class schedule in a time slot and day provided by the professor. Approval and denial on the request can be determined by the administrator
Test Actions:	1. From the “home” page select scheduling requests 2. After clicking on approve see the requests 3. Approve or deny the requests according to availabilities 4. Select “Update” button
Expected Results:	After step 4 , system should prompt a message indicating that the approved request has been processed successfully

Test Case ID:	5.1.1
Title:	Instructors Page
Feature/Subfeature:	Add Instructors Button/DropDown Menu
Purpose:	To Allow the User to add additional instructors in order to place them into the schedule generator
Initial Conditions:	User must be on the “Instructors” page
Test Data:	Test Data includes invalid Email entries [GinaAtGmail.com] and valid Name [Gina, Bob], Class [CS 451, SE Capstone], and Email [Gina@gmail.com, Gina@umsystem.edu] entries
Test Actions:	1. Hover cursor over “Add Instructors” button 2. Left click on the button 3. Left click the “Name” field on the drop-down menu 4. Enter name of instructor 5. Left click the “Class” field 6. Enter class name 7. Left click the “Email” field 8. Enter email 9. Left click “submit” button 10. Left click “Add Instructors” button again 11. Left click “Cancel” button on drop-down menu
Expected Results:	After step 2, a drop-down menu should appear with fields to be filled out. All fields must be properly filled out before the “Submit” button can be pushed. The “Email” field must contain an @ for the address. Steps 9 & 11 should bring the user back to the original “Instructors” Page

Test Case ID:	5.1.2
Title:	Instructors Page Top-bar Menu Buttons
Feature/Subfeature:	Navigational Top-Bar
Purpose:	To ensure that the Navigational Top-Bar at the top of the Instructors Page correctly navigates the user to their desired page
Initial Conditions:	The User must be on the Instructors Page
Test Data:	No data is needed to complete these tests

Test Actions:	1. Select the “Home” button at the top of the page 2. Navigate back to the “Instructors” page 3. Select the “Generate Schedule” button at the top of the page 4. Navigate back to the “Instructors” page 5. Select the “Logout” button at the top of the page
Expected Results:	Step 1 should bring the user to the home page “Index”. Step 3 should bring the user to the “Generate Schedule” page. Step 5 should bring the user to the original login page and require the user to login again in order to access the Scheduling System

Test Case ID:	6.1.1
Title:	Generate Schedule Page Menu and Button
Feature/Subfeature:	Select Schedule Drop-down Menu and Generate Button
Purpose:	To ensure the correctness of the Select Schedule Menu and Generate button on the “Generate Schedule” Page
Initial Conditions:	The User must be on the “Generate Schedule” Page
Test Data:	No data is needed to complete these tests
Test Actions:	1. Click the “Generate” button 2. Click the “Select Schedule” drop-down 3. Select a schedule from the list 4. Click the “Generate” button
Expected Results:	Step 1 should yield a warning telling the user to select a schedule before hitting the “Generate” button. Step 2 should show a list of all the current schedules. Step 4 should navigate the user to the “Schedule” page.

Test Case ID:	6.1.2
Title:	Generate Schedule Page Top-Bar Navigational Buttons
Feature/Subfeature:	Top-Bar Navigational Buttons
Purpose:	To ensure the correctness of the Top-Bar Navigational Buttons
Initial Conditions:	The User must be on the “Generate Schedule” Page
Test Data:	No data is needed to complete these tests

Test Actions:	1. Select the “Home” button at the top of the page 2. Navigate back to the “Instructors” page 3. Select the “Instructors” button at the top of the page 4. Navigate back to the “Generate Schedule” page 5. Select the “Logout” button at the top of the page
Expected Results:	Step 1 should bring the user to the home page “Index”. Step 3 should bring the user to the “Instructors” page. Step 5 should bring the user to the original login page and require the user to login again in order to access the Scheduling System

Test Case ID:	7.0.0
Title:	Connecting django to local Mysql Server
Feature/Subfeature:	Middleware
Purpose:	To connect the djago to a local server so we can pull from the server.
Initial Conditions:	User must be attempting to connect to the site
Test Data:	try to open one of the url pages
Test Actions:	1.turn on the server through the command term. 2. type in the desired url 3. connect to web page
Expected Results:	The html page linked to the url to pop up.

Test Case ID:	7.0.1
Title:	Connect the html file to the correct url
Feature/Subfeature:	Middleware
Purpose:	Connect the html files to the proper url so we can access the program
Initial Conditions:	User must be attempting to connect to the site
Test Data:	try to open one of the url pages
Test Actions:	1.turn on the server through the command term. 2. type in the desired url 3. connect to correct html page and be able to switch between the desired pages
Expected Results:	The html page linked to the url to pop up.

Test Case ID:	7.0.2
Title:	Have the css file format the html page correctly
Feature/Subfeature:	Middleware
Purpose:	to make the html page look like our test page
Initial Conditions:	User must be attempting to connect to the site
Test Data:	try to open one of the url pages
Test Actions:	1.turn on the server through the command term.

	2. type in the desired url 3. connect to web page 4. see the desired format of the page
Expected Results:	The html page linked to the url to pop up and be in the correct format.

Test Case ID:	7.0.3
Title:	Connect the AI portion to the rest
Feature/Subfeature:	Middleware
Purpose:	allow the user to use the ai schedule maker
Initial Conditions:	User must be connected to the site and clicking the great schedule button
Test Data:	Scheduler button click
Test Actions:	1. Turn on the server through the command term. 2. Type in the desired url 3. Connect to web page 4. Go the the scheduler page 5. Click the make new schedule button 6. See desired schedule
Expected Results:	A AI generated schedule to be generated and display on screen

Test Case ID:	8.0.1
Title:	Output completed schedule
Feature/Subfeature:	schedule generating
Purpose:	To provide each course with an assigned professor, day, and time slot
Initial Conditions:	User must have already provided a completed list of professors and courses on the “instructors page” then navigate to the “generate schedule” page
Test Data:	A list of professors and their assigned courses
Test Actions:	1. Go to “instructors” page 2. Enter all Professors and their courses 3. Navigate to the “generate schedule” page 4. click “create schedule”
Expected Results:	An easy to read schedule will be shown listing each course and their assigned professor, day, and time. For every professor/course pair provided, there must be an assigned day and time

Test Case ID:	8.0.2
Title:	Schedule preferences and restrictions

Feature/Subfeature:	schedule generating
Purpose:	To create a schedule that considers specific day and time preferences for certain professors and courses.
Initial Conditions:	User must have already provided a completed list of professors and courses on the “instructors page”, additionally inputting what days and times professors prefer not to have class at. Then navigate to the “generate schedule” page .
Test Data:	A list of professors and their assigned courses along with day/time preferences.
Test Actions:	5. Go to “instructors” page 6. Enter all Professors, their courses, and preferences 7. Navigate to the “generate schedule” page 8. click “create schedule”
Expected Results:	An easy to read schedule will be shown listing each course and their assigned professor, day, and time. For every professor/course pair provided, there must be an assigned day and time. Most preferences should be met. However, not every single one necessarily will be due to other conflicts and priority..