

# Testing Document and Specification Test Plan

Alternate Project  
Group 8  
CS 451

Adam Flowerday  
Thom Martin

Lake Gatton  
Ryan Moody

Elizabeth Nastoff  
Zunair Ahmed

## Introduction

This document outlines the test plan for the University of Missouri-Kansas City Automated Scheduling System. Per this project's Requirements Document, this system should enable Gina Campbell to enter instructor schedule requests to create semester schedules. This system will also act as a database for UMKC instructors and their schedule preferences. The tests discussed in this document will verify whether the software for the Scheduling system meets the requirements as outlined in the Requirements Document.

It should be acknowledged that there was an option, during the development of this project, to cover a large portion of the functional requirements with the javascript/HTML code that controls the user interface portion of this project. This route was not taken as a main concern of the ongoing use of this program is security of sensitive information: teacher names and locations, as well as possible medical information that would need to be stored very securely. The initial hope was that, by using python and Django together, the overall program would have a high ease-of-use and ease-of-access, and would also have a strong security protocol in place to protect this sensitive data. This goal was ultimately overshadowed by the difficulty encountered in setting up a Django server effectively for the project's needs, and testing will be performed on the available code.

The testing routine would test the user login, front end navigation to each page, information submission and retrieval to and from the database, and accuracy/usability of the automatic schedule algorithm. The results of this testing procedure would enable the creators of this system to gauge project success as outlined in the Macro Project Plan.

## Terminology

Throughout this document the terms user, instructor, system, site and username/password will be used frequently therefore, formal definitions will be given.

System- the databases and code that support the backend of the website

User - Gina Campbell

Instructor – General term for anyone included in or added to the schedule

Site- the front-end website

Username/Password - unique identifiers that authenticate and validates a user - username is designed to be an email address, specifically Gina Campbell's official UMKC email

## **Items Tested**

Documented tests will be based on the project requirements outlined in the Requirements Document and will include:

- Ability for user to securely login to the site via username and password
- Ability for user to submit a new instructor to the corresponding database
- Ability for user to submit a new schedule request to the corresponding database
- Ability for user to edit an existing schedule request to the corresponding database
- Ability for user to approve an existing schedule request in the system
- Ability for user to generate a schedule

## **Items Not Tested**

There are features in this project that could not be tested, largely due to overall issues with integration between the front end and back end. As the actual functionality of the project depends heavily on this relationship, testing will largely be of the individual groups (front-end and back-end).

- Login verification (sending Username/Password to database to compare input with secure login information and returning the appropriate function)
- Adding instructor and request information from HTML user-interface to the SQL back-end database
- Retrieving instructor and request information from SQL back-end database using HTML user-interface command
- Utilizing database entries to generate a working semester schedule

## **Approach**

The overall method to this testing procedure is manual system testing. This will largely amount to local testing of the front-end and back-end programs separately. Each process will be tested individually for clarity; the processes themselves will be simplified and incomplete versions of the final product but will be documented for posterity and consistency.

## Item Pass/Fail Criteria

As tests are being performed on what can be called a simplified form of the intended code, success and failure will be more relative terms as relayed in this document. The user interface tests will not be able to perform any “get” or “show” commands to add, edit or delete information from the databases nor is there any way for the user to trigger the scheduling algorithm. This means each test result, by default, would be a failure. Separating the front-end and back-end tests will at least allow the creators to review the code locally and measure the results accordingly.

Pass criteria will be based on each test, but generally speaking the front-end testing will involve interacting with the user interface: full inter-connectivity of the html sites (can reach virtually any page through any other page, excluding the “forget password” page and embedded content), and correct display of embedded content with user trigger. Back-end testing will be more complex but will be outlined for the individual tests.

Incorrect output and redirecting will all be considered failures in this testing. Failures of full-stack integration will be detailed in the test reports.

## Test Deliverables

Test deliverables include:

- Test Plan
- Test Specifications
  - outlines specific test cases and expected results of each test
- Test Reports
  - Incidents
  - Defects
  - Changes.

## Testing Tasks

The following list the testing deliverables and the activities required to produce the deliverable.

Deliverables	Activities
Test Plan	<ul style="list-style-type: none"><li>● Analyze Requirements for System Features</li><li>● Determine Testable/Non-Testable Features</li><li>● Develop Approach/Method for testing</li><li>● Outline Major Testing Obstacle(s)</li></ul>
Test Specifications	<ul style="list-style-type: none"><li>● Analyze Requirements</li><li>● Define Test Cases for Testable Features as Outlined by the Test Plan</li></ul>

Test reports	<ul style="list-style-type: none"> <li>● Implement Test Cases as Outlined by the Test Specifications</li> <li>● Document Incidents and Defects</li> <li>● Determine Severity of Incidents and Defects</li> <li>● Determine Changes that Need to be Made to System</li> </ul>
--------------	--