TJ Durnford
Andrew Ringquist
George Kunthra
Devin Roche
Ben Rieckers

Schedule Builder Business Function Model

The Schedule Builder application will help Gonzaga students create a class schedule for a semester by taking a set of user selected courses and displaying it on an adjustable user interface.

This application will be achieved by having select users type a select number of courses into the application. The test will determine internal processing outcomes such as accuracy of schedule, randomness of times/professors, and the application's speed. The test will also determine certain response outcomes such as ease of use and user success rate.

When setting up a test, the tester must be able to specify:

- The kind of test such as A/B testing or student demographics testing.
- The selected students to test the Schedule Builder Application.
- Section details from course database.

For each test, response and user activity must be recorded. Response activity needs to include the following outcomes: application's response time in generation of schedule, accuracy of schedule, and ability to continue to generate random schedules with varying professors and times. User activity needs to include the following outcomes: ease of use, clicks, complaints, success rate, search, and save.

The tests must be able to gather and record section details. Section details must be able to pull section detail and include the following: section data, location, time, and professors.

The test must allow the program to be flexible with all possible inputs or however the user modifies their desired schedule.

After these test runs, we will be able to use the data to build a model for future users. Our team will record what users selected for number of classes each day, professors, times, and locations. Using this information, we can determine the most popular options that users select. We hope in the future that our application can use this data to generate better schedules for future users.

Through these test runs of the application, we hope to take all of the recorded activity and be able to apply this data in future modifications of the application. Some possible future modifications include: a sub-program which would automate the registration process by entering the user's desired CRN numbers into Zagweb's registration system on the night of the user's registration period. A class and teacher rating/review scale in order to assist future users' in their class search. A database of popular teachers and times from previous users in order to have the application include these possible teachers and times when generating a schedule.

In addition to these extensions, the Schedule Builder application will open up opportunities for similar campaigns such as a daily planner application and a workout planner application which are both based on similar systems. The data we hope to collect through the test runs of the Schedule Builder application will also show our team what aspects of the application do not work well and need to be terminated for future modifications.

Schedule Builder Application Business Function Model Outline

- Target test campaign
    - Identify what to test
    - Identify who is to test
    - Identify section details
- Identify campaign activity
    - Identify response activity
        - Examples: application's response time, accuracy of schedule, random schedule generator, speed
    - Identify user activity
        - Examples: ease of use, clicks, complaints, success rate, search, and save.
- Identify Section details
    - Pull section data
    - Pull class time
    - Pull class professor
    - Pull class location
- Build Predictive Models
    - Collect all response data
    - Select an algorithm with this data
    - Build a model for better schedules
    - Validate Model
- Future modifications
    - Apply what we have learned in the past
    - Potential future modifications
    - Terminate ineffective features