

Data Structures and Algorithms

Final Project

Team OG

Team Members: Hassan Abbasi, 05468

Project: Semester Schedule Builder

Objective

Prior to the beginning of every semester, students are required to manually formulate their schedules according to the courses they are required to take and their personal preferences. However, due to the poor design of the ERP system, it takes a long time to manually check all possible schedule combinations.

Moreover, coming up with alternate schedules in case of unavailability of classes makes the process even more tiresome. Taking all of this into consideration I have decided to build a WEB API which will allow students to select the courses they would like to take and receive a valid list of schedules in the form of a JSON response.

Outcomes

Send Get Request to get list of courses (1)

<http://therovecompany.com:8080/MyHabibScheduler/HabibCS>

Send Get Request with parameter codes to get a list of valid schedules (2)

<http://therovecompany.com:8080/MyHabibScheduler/HabibCS?codes=>

Sample Run:

User goes to (1) to retrieve the list of offered courses for the respective semester.

He then copies all the course codes (not the NBR codes but the course codes).

Finally, the user enters the copied codes as plain text after (2) and obtains all possible schedule combinations, containing the courses he has selected. Screenshots of the process have been attached below:

(1)

```

1 // 20200624124854
2 // http://therovecompany.com:8080/MyHabibScheduler/HabibCS
3
4 [
5 {
6   "name": "CS",
7   "courses": [
8     {
9       "name": "Computer Science Freshman Seminar",
10      "code": "CS 100"
11    },
12    {
13      "name": "Programming Fundamentals",
14      "code": "CS 101"
15    },
16    {
17      "name": "Data Structures and Algorithms",
18      "code": "CS 102"
19    },
20    {
21      "name": "Object Oriented Programming and Design Methodologies",
22      "code": "CS 224"
23    },
24    {
25      "name": "Operating Systems",
26      "code": "CS 232"
27    },
28    {
29      "name": "Introduction to Computational Social Science",
30      "code": "CS 262"
31    }
32  ]
33 }
34 ]

```

(2) URI: "<http://therovecompany.com:8080/MyHabibScheduler/HabibCS?codes=CS 100, CS 102, CS 242>"

As per the example, the courses selected are CS 100 (freshman seminar), CS 102 (Data Structures & Algorithms) and CS 242 (Object Oriented Programing and Design Methodologies)

```

1 // 20200624125627
2 // http://therovecompany.com:8080/MyHabibScheduler/HabibCS?codes=CS%20100,%20CS%20102,%20CS%20242
3
4 [
5 {
6   "Tu": [
7     {
8       "code": "1089",
9       "startTime": "11:30 AM",
10      "endTime": "12:45 PM",
11      "instructor": " Staff Staff"
12    }
13  ],
14   "Mo": [
15     {
16       "code": "1079",
17       "startTime": "05:30 PM",
18       "endTime": "06:20 PM",
19       "instructor": " Staff Staff"
20    }
21  ],
22   "Th": [
23     {
24       "code": "1089",
25       "startTime": "11:30 AM",
26       "endTime": "12:45 PM",
27       "instructor": " Staff Staff"
28    }
29  ],
30   "We": [

```

The partially displayed list of schedules above can be accessed at the following URI:

<http://therovecompany.com:8080/MyHabibScheduler/HabibCS?codes=CS%20%20100,%20CS%20%20102,%20CS%20%20242>

Code Documentation

The project contains a package named *com.roveapps.hScheduler*. This package contains all the relevant classes for creating a schedule. The project's main package is *com.roveapps*. It contains the sub package and relevant classes for creating a Java Servlet.

HScheduler:

Please find details of the classes utilized by the program to build and display schedules.

Class: Caching Service

- **Functionality:** Stores the document received from the Habib server and on requesting of document, provides a valid cached copy from storage, if one is available.

Class: HTML Parser

- **Functionality:** Parses the HTML document and extracts all data using JSOUP. The data is used to create objects of time Courses, Class etc.

Class: Networking Service

- **Functionality:** Connects to the Habib Server to authenticate the user and fetch authentication token. The authentication token is then persisted and used to fetch other information from the server such as the list of courses and the respective classes available.

Class: Subroutines

- **Functionality:** Contains global helper functions and properties such as conversion from `String[]` to `String`.

Class: Scheduler

- **Functionality:** Contains functions to create a graph, traverse it and evaluate if it is valid or not.

Class: User Service

- **Functionality:** Contains function to check if user has successfully authenticated on the server.

Object: Course

- Properties:
 - o List of type Class
 - o String name
 - o String code

Object: Class

- Properties:
 - o String Array of Instructors
 - o String Array of Rooms
 - o Dictionary of Timings
 - o String code

Object: Timing

- Properties:
 - o Date StartTime
 - o Date EndTime

Main Package:

Class: HabibCS

- Functionality: Creates a servlet and checks GET request params

Class: JsonObjects

- Functionality: Contains all objects used in the JSON. Objects contains only relevant properties.

Class: Main

- Functionality:
 - o Initializes the Networking Singleton
 - o Fetches list of course
 - o Initializes Scheduler with required courses and parses the data returned by the Scheduler into human readable schedules.

Complexity Analysis

The complexity analysis of the code is difficult and requires the use of probability. This is because it has a lower bound of $n!$. However, it is more efficient than that as it uses BackTracking to reject paths which are not valid. This reduces the overall complexity, however to determine the factor by which it is reduced, the use of probability is required.