|  |  |  |
| --- | --- | --- |
| **Test Number: 1** | | **Number of Tests**: 100 |
| **Test Definition:** | | |
| * This test case tests the generation of initial population with a determined population size | | |
| **Expected Value** | | |
| * Initial population must be generated with given population size | | |
| **Test Steps** | | |
| * Population takes population size and connection variables of mySql DB * Population.java generates and returns the population | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

**3GAG’S SENIOR PROJECT TEST DOCUMENT**

**Algorithm Tests**

|  |  |  |
| --- | --- | --- |
| **Test Number: 2** | | **Number of Tests**: 100 |
| **Test Definition:** | | |
| * This test case tests the ‘TournamentSelection’ function | | |
| **Expected Value** | | |
| * Selects an individual/chromosome within the population | | |
| **Test Steps** | | |
| * Randomly picks 3 individual within population * Selects the one with highest fitness | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 3** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case tests the ‘UniformCrossover’ function | | |
| **Expected Value** | | |
| * Creation of a brand new individual | | |
| **Test Steps** | | |
| * Parent1 and parent2 exchanges genes * A child must be produced with same number of genes. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 4** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case tests the ‘Mutation’ function | | |
| **Expected Value** | | |
| * Mutates a gene of an individual with a little probability. | | |
| **Test Steps** | | |
| * A random number between [0 1] generated * If generated number is less than 0.001, then target gene mutated to a new one | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 5** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case tests the ‘getBestTimeTable’ function | | |
| **Expected Value** | | |
| * Selection of an individual with best fitness | | |
| **Test Steps** | | |
| * Takes population * Then, returns an individual with best fitness value | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 6** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case tests the ‘Elitisism’ function * Elitisism ensures the conservation of elite individual. | | |
| **Expected Value** | | |
| * Elite individual must be transferred to the next generation. | | |
| **Test Steps** | | |
| * Elite individual of current generation copied into new generation | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 7** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests the creation of an Individual/Timetable | | |
| **Expected Value** | | |
| * Timetable’s courses/genes must be assigned sequentially. | | |
| **Test Steps** | | |
| * Function creates 1xN matrix(a chromosome/individual) * Fills every gene with a course object | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 8** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘setter’ functions of algorithm | | |
| **Expected Value** | | |
| * Value must be assigned to object succesfully | | |
| **Test Steps** | | |
| * Assign value to target object | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 9** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘getter’ functions of algorithm | | |
| **Expected Value** | | |
| * Requested value must be called succesfully | | |
| **Test Steps** | | |
| * Calls the requested value from the target object | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 10** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘FindFittest’ function | | |
| **Expected Value** | | |
| * This function must find the fittest individual within the current generation | | |
| **Test Steps** | | |
| * Takes the population * Iterates through each individual to find the fittest and returns the fittest individual. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 11** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests creation of an AVLTree | | |
| **Expected Value** | | |
| * Objects must be properly assigned to the AVLTree | | |
| **Test Steps** | | |
| * İnserts an object into AVLTree * Checks for violations and corrects if any | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 12** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘LecturerTree’ | | |
| **Expected Value** | | |
| * Must create a balanced binary tree of lecturers | | |
| **Test Steps** | | |
| * Takes all lecturer objects * Calls fillLecturerTree function and adds lecturers to newly created tree | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 13** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘fillLecturerTree’ | | |
| **Expected Value** | | |
| * Must add a lecturer object to lecturerTree | | |
| **Test Steps** | | |
| * Takes a lecturer object * Calls AVLTree function and adds lecturer to target tree | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 14** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘TimetableTree’ | | |
| **Expected Value** | | |
| * Must create a balanced binary tree of timetables | | |
| **Test Steps** | | |
| * Takes all Timetable objects * Calls fillTimetableTree function and adds timetables to newly created tree | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 15** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘fillTimetableTree’ | | |
| **Expected Value** | | |
| * Must add a timetable to timeTableTre | | |
| **Test Steps** | | |
| * Takes a timetable object * Calls AVLTree function and adds Timetable to target tree | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 16** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘CheckLecturerPreferences’ function | | |
| **Expected Value** | | |
| * If assigned course, day and hour combination overlaps with lecturer preferences, related hard or soft constraint value must be incremented | | |
| **Test Steps** | | |
| * Checks lecturer hashmap to find overlaps according to a lecturer’s preference * If there is overlapping increases the related hard/soft constraint. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 17** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘generateKeyForHashMaps’ | | |
| **Expected Value** | | |
| * Must allocate a space in memory for specified combination | | |
| **Test Steps** | | |
| * Has a list of keys * Generates a key string for given combination and adds it to the list of keys | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 18** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘ModifyHashMaps’ function | | |
| **Expected Value** | | |
| * Must check and modify hashmap for constraint values | | |
| **Test Steps** | | |
| * Takes the key holder * If it detects overlapping modfies the respected constraint value | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 19** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘handleHashMaps’ function | | |
| **Expected Value** | | |
| * Must handle soft and hard constraints of a timetable | | |
| **Test Steps** | | |
| * Calls generateKeyForHashmap function * Returns hard and soft constraints of the target timetable. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 20** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘insert’ function of AVLTree | | |
| **Expected Value** | | |
| * Must insert specified object with cost and variable to the target tree | | |
| **Test Steps** | | |
| * Defines and matches the tree using variable * Inserts the element using the cost, calls fix and balance functions to handle violations | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 21** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘balance’ function of AVLTree | | |
| **Expected Value** | | |
| * A balanced binary tree is the expected value. | | |
| **Test Steps** | | |
| * Checks whether the target tree is balanced or not * If it is not balanced, calls fix function up on nodes that causes violation | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 22** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘fix’ function of AVLTree | | |
| **Expected Value** | | |
| * Must fix the violations | | |
| **Test Steps** | | |
| * It does the left and right rotations to maintain the balance | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 23** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘find\_leastValue’ function | | |
| **Expected Value** | | |
| * Must return the node with leastValue | | |
| **Test Steps** | | |
| * Takes the target tree * Iterates through the node with less value till it reaches the leaf node | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 24** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘updateHeight’ function | | |
| **Expected Value** | | |
| * Must update height when a modification done | | |
| **Test Steps** | | |
| * Checks the target node and update its height if necessary | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 25** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests whether ‘course’ object constructed correctly or not | | |
| **Expected Value** | | |
| * A course object must constructed with given values | | |
| **Test Steps** | | |
| * Takes these values to construct the course object:   courseID, capacity, groupofCourse, departmentId, roomType, Duration, hour & day  section, heavyCourses, lecturerId, yearofCourse | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

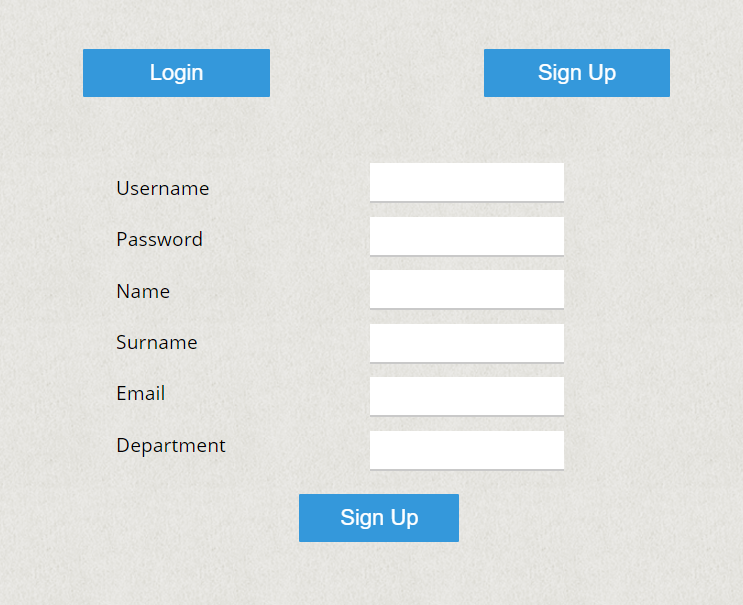
|  |  |  |
| --- | --- | --- |
| **Test Number: 26** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests whether ‘lecturer’ object constructed correctly or not | | |
| **Expected Value** | | |
| * A lecturer object must constructed with given values | | |
| **Test Steps** | | |
| * Takes these values to construct a lecturer object:   lecturerId, lecturername, lecturerPreferences | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 27** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests whether ‘room’ object constructed correctly or not | | |
| **Expected Value** | | |
| * A room object must constructed with given values | | |
| **Test Steps** | | |
| * Takes these values to construct a room object:   roomId, roomCapacity, roomType | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number: 27** | | **Number of Tests**: 100 |
| **Test Definition** | | |
| * This test case, tests ‘EvolvePopulation’ function | | |
| **Expected Value** | | |
| * Evolves population to next generation | | |
| **Test Steps** | | |
| * Calls geneticAlgorithm functions   Selection, Crossover, Mutation, Elitisism sequentially. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

**WEB SERVER TESTS**

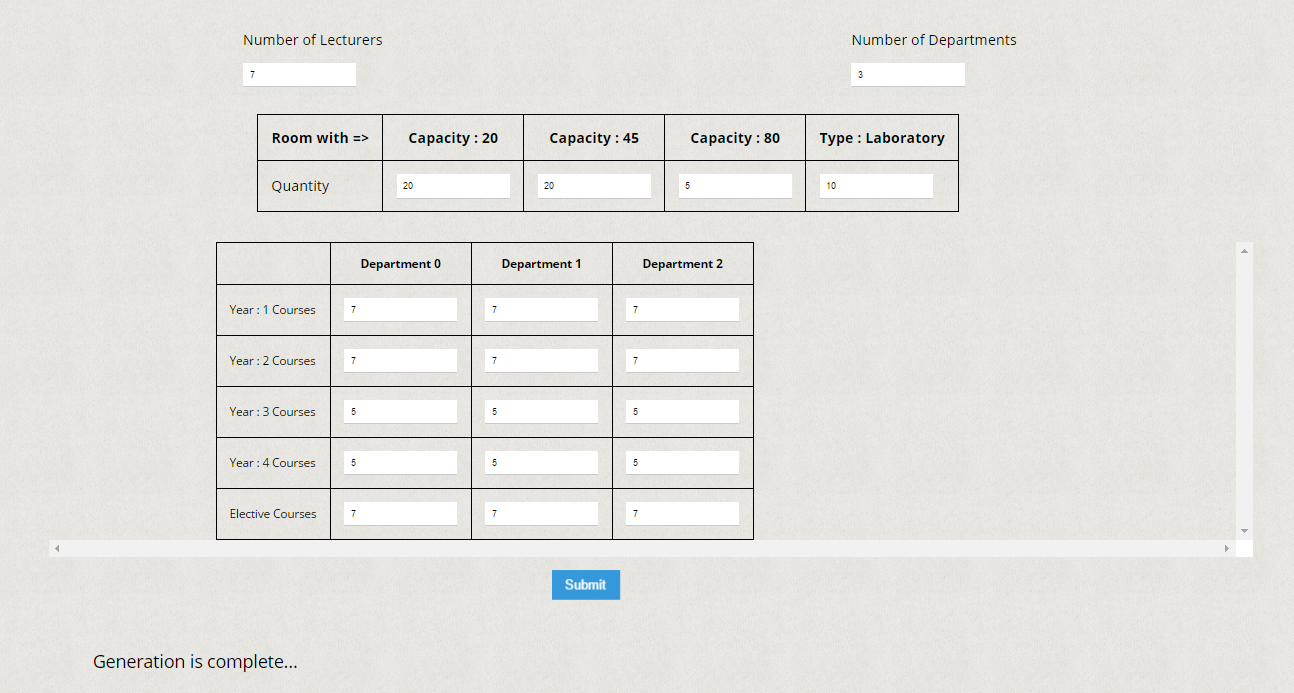
|  |  |
| --- | --- |
| **Test Number : 1** | |
| **Test Definition** | |
| This test case verifies to create new account. | |
| **Expected Value** | |
| User should be able to create new account by sign up button. | |
| **Test Steps** | |
| * Click the sign up button. * Fill the necessary informations. * Click the sign up button again which is appeared at the bottom of page. | |
| **Test Result** | X  Failed  Succesful |
| **If test failed please give more information and expectations** | |



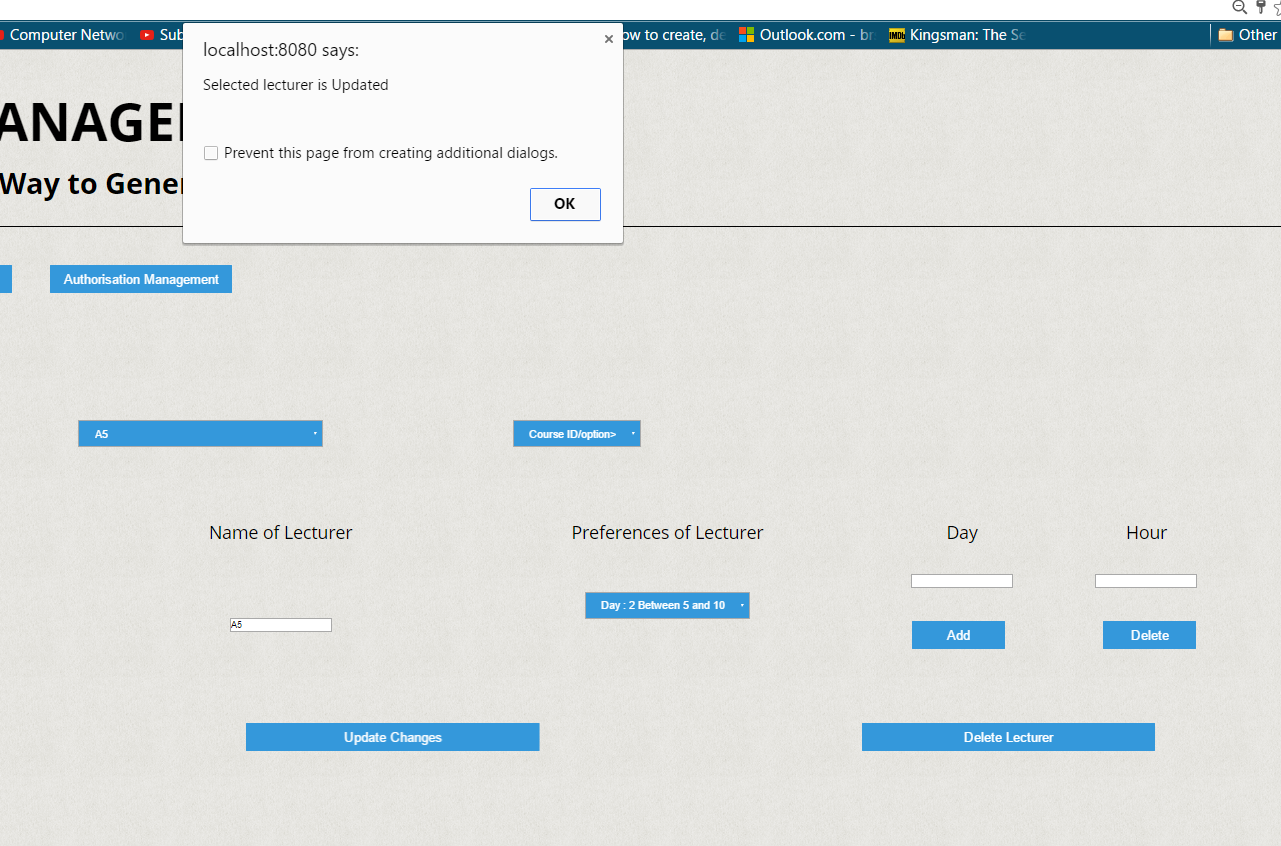
|  |  |
| --- | --- |
| **Test Number : 2** | |
| **Test Definition** | |
| This test case verifies to login process | |
| **Expected Value** | |
| User should be able to login to the system | |
| **Test Steps** | |
| * Click the login button. * Fill the necessary informations. * Click the login button again which is appeared at the bottom of page. | |
| **Test Result** | X  Failed  Succesful |
| **If test failed please give more information and expectations** | |



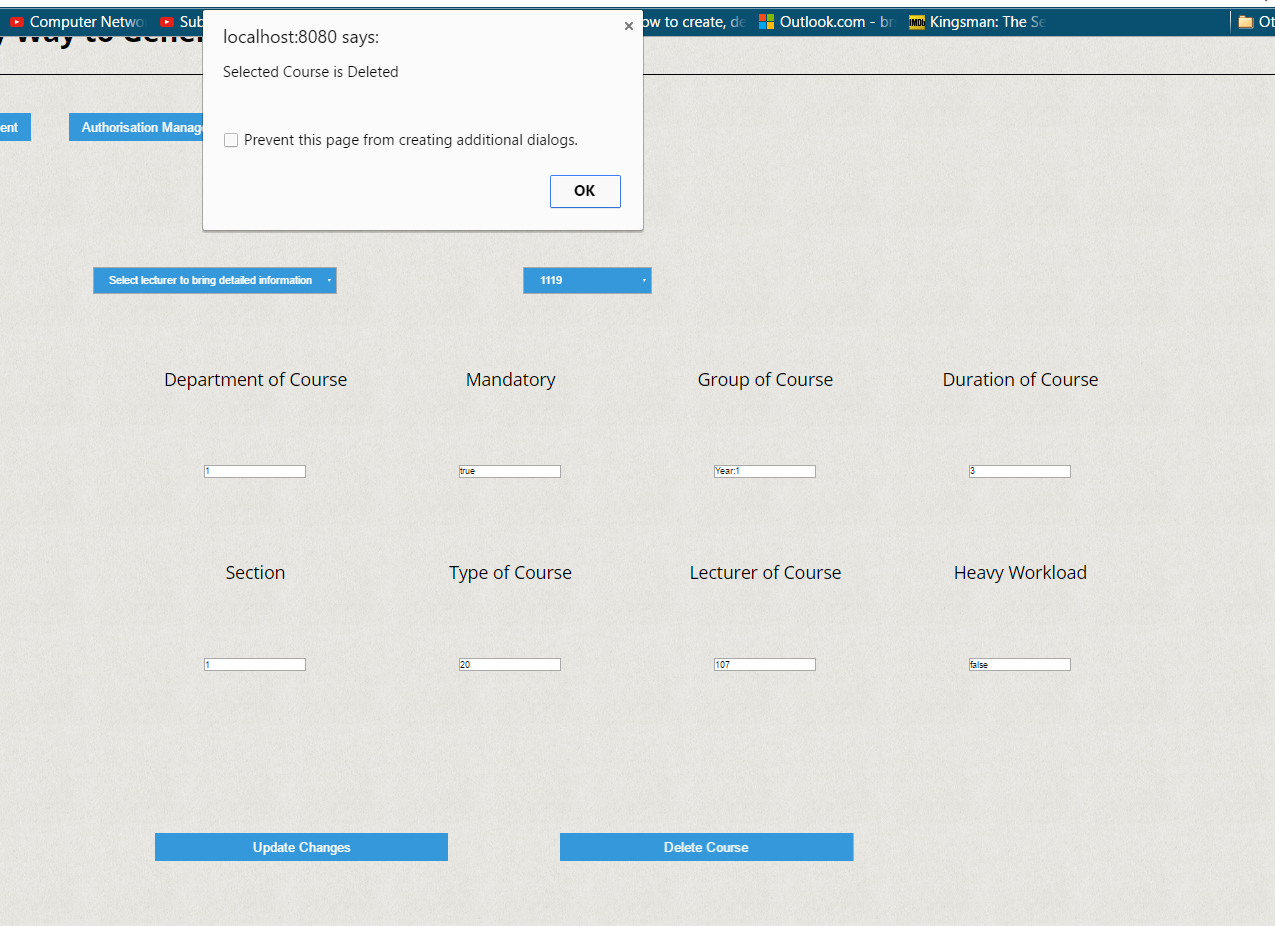
|  |  |
| --- | --- |
| **Test Number : 3** | |
| **Test Definition** | |
| This test case verifies to automatic data creation | |
| **Expected Value** | |
| Server should return “Generation is complete...” | |
| **Test Steps** | |
| * Click the Data Management button. * Create Data by Excel, Create Data Automatically and Manage Data Fields should appear. Choose create data automatically button. * The program needs user to fill the specific informations. Fill the necessary parts “number of lecturer” etc. * Click submit button * Server returns “Generation is complete...” | |
| **Test Result** | X  Failed  Succesful |
| **If test failed please give more information and expectations** | |



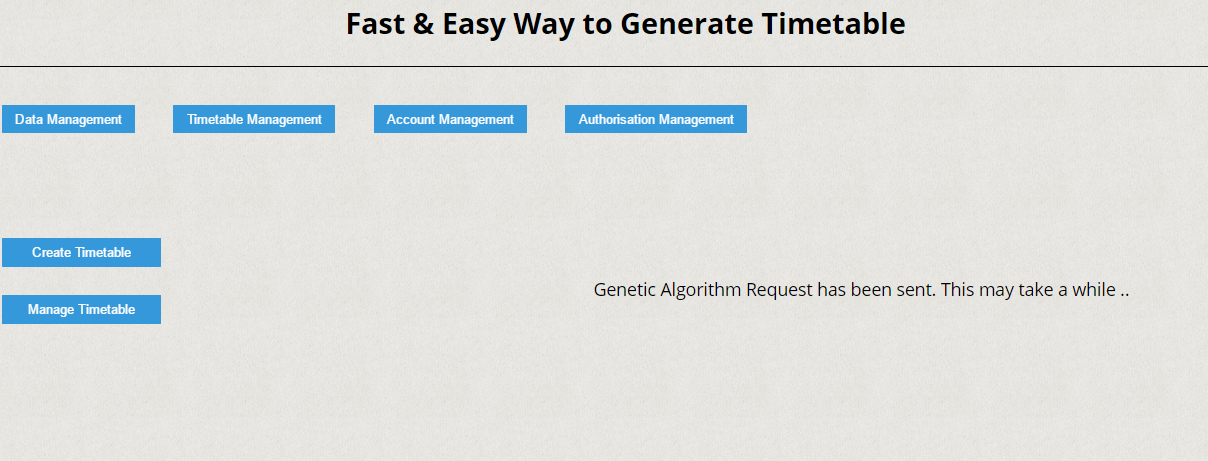
|  |  |  |
| --- | --- | --- |
| **Test Number : 4** | | Number of tests: 20 |
| **Test Definition** | | |
| This test case verifies to updating changes on generated data | | |
| **Expected Value** | | |
| Server should return “Update is complete” | | |
| **Test Steps** | | |
| * Click the Data Management button. * Create Data by Excel, Create Data Automatically and Manage Data Fields should appear. Choose Manage data button. * Server brings the list of lecturers and courses. Select one of them to make changes. * Server brings every detail of selected course or lecturer. Make changes and press update button. * Server notifies the update is succesful. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |



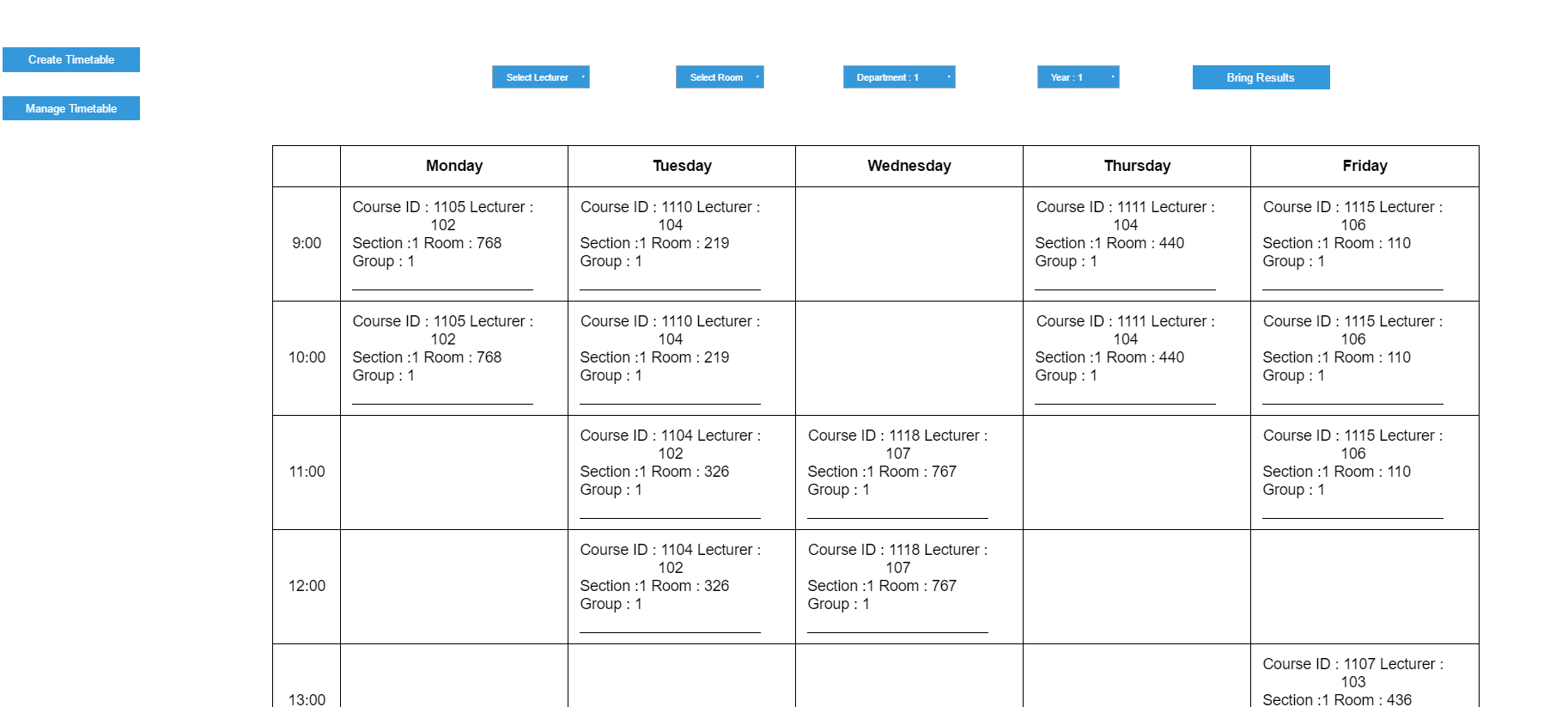
|  |  |  |
| --- | --- | --- |
| **Test Number : 5** | | Number of tests: 20 |
| **Test Definition** | | |
| This test case verifies to deletion process of data | | |
| **Expected Value** | | |
| Server should return “Selected course/lecturer is deleted” | | |
| **Test Steps** | | |
| * Click the Data Management button. * Create Data by Excel, Create Data Automatically and Manage Data Fields should appear. Choose Manage data button. * Server brings the list of lecturers and courses. Select one of to delete. * Press Delete button. * Server notifies the deletion is succesful. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |



|  |  |  |
| --- | --- | --- |
| **Test Number : 6** | | Number of tests: 20 |
| **Test Definition** | | |
| This test case verifies to sending request of timetable creation | | |
| **Expected Value** | | |
| Server should return “Genetic Algorithm Request has been sent. This may take a while...” | | |
| **Test Steps** | | |
| * Click the Timetable management button. * “Create timetable” and “Manage timetable” buttons must appear. Click “Create timetable” button. * Server notifies user about request. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |



|  |  |  |
| --- | --- | --- |
| **Test Number : 7** | | Number of tests: 20 |
| **Test Definition** | | |
| This test case verifies to bring result of generated timetable. | | |
| **Expected Value** | | |
| Server should return the timetable according to the specifications | | |
| **Test Steps** | | |
| * Click the Timetable management button. * “Create timetable” and “Manage timetable” buttons must appear. Click “Manage Timetable ” button. * Server returns the list of lecturers, departments, classrooms and groups. * Select the options and click “Bring Results” button. * Server receive options and returns result if this result exists. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |



|  |  |  |
| --- | --- | --- |
| **Test Number : 8** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies bringcourseServlet.java | | |
| **Expected Value** | | |
| Servlet should bring the course informations according to the parameters. | | |
| **Test Steps** | | |
| * Servlet receives parameters from client. * Servlet returns course information if course exists. If course object does not exist, then servlet notifies user. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 9** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies bringdataServlet.java | | |
| **Expected Value** | | |
| Servlet should list course and lecturer names and send to client. | | |
| **Test Steps** | | |
| * Server receives request and launches bringdataServlet * This servlet takes informations from database,list informations and send these informations to client . | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 10** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies bringlecturerServlet.java | | |
| **Expected Value** | | |
| Servlet should send the lecturer informations according to the parameters. | | |
| **Test Steps** | | |
| * Servlet receives parameters from client. * Servlet connects to database and takes the corresponding informations. * Servlet returns lecturer informations if lecturer exists. If lecturer does not exist, then servlet notifies user. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 11** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies bringresultSevlet.java | | |
| **Expected Value** | | |
| Servlet should send the lecturer informations according to the parameters. | | |
| **Test Steps** | | |
| * Servlet receives parameters from client. * Servlet connects to database and takes the corresponding informations. * Servlet returns the part of result according to the parameters. If result doesnot exist,then servlet notifies user. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 12** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies DataGenSevlet.java | | |
| **Expected Value** | | |
| Servlet should create test data according to the parameters. | | |
| **Test Steps** | | |
| * Servlet receives parameters from client. * Servlet starts to create test data according to the parameters. * Servlet connects to database and stores test data. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 13** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies deletecourseSevlet.java | | |
| **Expected Value** | | |
| Servlet should delete the specified course object from database. | | |
| **Test Steps** | | |
| * Servlet receives the id of course object from client. * Servlet connects to the database and deletes the specific course object. * Servlet notifies user about deletion process. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 14** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies deletelecturerSevlet.java | | |
| **Expected Value** | | |
| Servlet should delete the specified lecturer object from database. | | |
| **Test Steps** | | |
| * Servlet receives the id of lecturer object from client. * Servlet connects to the database and deletes the specific lecturer object. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 15** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies ExecuteGeneticAlgorithmSevlet.java | | |
| **Expected Value** | | |
| Servlet should run ExecuteGeneticAlgorithm thread. | | |
| **Test Steps** | | |
| * Server receives request and launches ExecuteGeneticAlgorithmServlet. * Servlet creates and runs ExecuteGeneticAlgorithm thread. * Servlet notifies the user. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 16** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies loginSevlet.java | | |
| **Expected Value** | | |
| Servlet handles login process. | | |
| **Test Steps** | | |
| * Servlet receives username and password as parameter. * Servlet connects to database and validates username & password. * Servlet gives authorisation according to validation process. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 17** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies ManageTimetableSevlet.java | | |
| **Expected Value** | | |
| Servlet sends information of departments, courses, rooms, lecturers. | | |
| **Test Steps** | | |
| * Server receive request and starts ManageTimetableServlet. * Servlet connects database and takes list of departments, courses, rooms, lecturers. * Servlet sends lists. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 18** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies signupSevlet.java | | |
| **Expected Value** | | |
| Servlet handles sign up process. | | |
| **Test Steps** | | |
| * Servlet receives user informations as parameter. * Servlet connects to database and validates username.If username does not exist, it creates new user. * Servlet notifies the user. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 19** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies updatecourseSevlet.java | | |
| **Expected Value** | | |
| Servlet updates course according to parameters. | | |
| **Test Steps** | | |
| * Servlet receives course id and other parameterers. * Servlet connects to database and validates course id. * If course id is valid, servlet updates changes on this course. * Servlet notifies user about process. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

|  |  |  |
| --- | --- | --- |
| **Test Number : 20** | | Number of tests: 5 |
| **Test Definition** | | |
| This test case verifies updatelecturerServlet.java | | |
| **Expected Value** | | |
| Servlet updates lecturer according to the parameters | | |
| **Test Steps** | | |
| * Servlet receives lecturer id and other parameterers. * Servlet connects to database and validates lecturer id. * If lecturer id is valid, servlet updates changes on this lecturer. * Servlet notifies user about process. | | |
| **Test Result** | X  Failed  Succesful | |
| **If test failed please give more information and expectations** | | |

**Encountered Problems**

We have encountered with these problems ;

* Local optimum problem : Basically, this problem is choosing the wrong path to solve the problem. This wrong path causes a blockage and effects the performance of algorithm dramatically. We tried to implement the “Asynchronous Island Model” to solve this problem. However we were lack on funds and we couldn’t find hardware support.
* Fitness Calculation : Fitness calculation is heart of our system and it effects the performance of our algorithm negatively. We created hashmap system to handle this calculation effectively.
* Hardware problems : Our web server runs on personal computer. However the genetic algorithm consumes more cpu power and memory space than regular request.Therefore our computer failed to meet multiple genetic algorithm request at the same time. We implemented ajax functions to reduce workload of web server. It is succesful to handle regular requests with genetic algorithm request. Unfortunately web server still has a low limit of request capacity.