| **Class Name:** TuitionManagerController  **Method Signature:** checkValidAddStudentInput() {} //*Validates if the current user input can successfully create a student.* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | The user is required to input the student’s first name. | * User leaves the first name text field blank. * Test input: “” | “Enter a first name for the Student.” |
| 2 | The user is required to input the student’s last name. | * User leaves the last name text field blank. * Test input: “” | “Enter a last name for the Student.” |
| 3 | The user is required to input the date value for the student. | * User leaves the date picker field empty * Test input: “” | “Enter a date of birth.” |
| 4 | The user is required to select a major for the student. | * User does not pick a major * Test input: “” | “Please select a major” |
| 5 | The user is required to select a residency for the student. | * User does not pick a residency * Test input: “” | “Please select a form of residency.” |
| 6 | The user is required to input the student’s credits completed. | * User leaves the credits completed text field blank. * Test input: “” | “Enter credits completed.” |
| 7 | The user is supposed to input only digit in the credits completed text field. | * User enters a non-digit entry in the credits completed text field. * Test input: “AAA” | “Please only use digits for credit completed” |
| 8 | The user is supposed to input only positive digit in the credits completed text field. | * User enters a negative entry in the credits completed text field. * Test input: “-12” | “Please use positive digits for credit completed” |
| 9 | If the user selects residency as tristate, they are also supposed to select NY or CT button as tristate. | * User does not select NY or CT buttons * Test input: “” | “Please select a state” |

| **Class Name:**TuitionManagerController  **Method Signature:** loadRosterFromFile(){} //*loads student objects into the roster from a txt file* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | User is supposed to load a txt file. To read in a list of students to register. | * User selects the “Load from File '' button and selects the text file. * Test input: Select “Load from file button. Select studentList.txt file. | “Roaster loaded from file studentList.txt” |
| 2 | User is supposed to load a txt file. To read in a list of students to register. | * User selects the “Load from File '' button and selects the text file. * Test input: Select “Load from file button. Select studentList1.txt file. | “File studentList1.txt file not found” |

| **Class Name:**TuitionManagerController  **Method Signature:** addStudentToRoster(){} //*Adds a student to the roster based on user input specifications.* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | User enters a new student with proper inputs. | * User enters the students First name, Last name, Date of birth, selects major, enters credit completed, and selects the residency status. * Test input:   + FirstName: “Emma”   + LastName: “Miller”   + Date of birth: 2/28/2003   + Major: CS   + Credits Completed: “15”   + Status: “Non-Resdient”, “Tristate”, “NY” | “Emma Miller 2/28/2003 was added to the roster.” |
| 2 | User enters the same student from test case one with proper inputs. | * User enters the students First name, Last name, Date of birth, selects major, enters credit completed, and selects the residency status. * Test input:   + FirstName: “Emma”   + LastName: “Miller”   + Date of birth: 2/28/2003   + Major: CS   + Credits Completed: “15”   + Status: “Non-Resdient”, “Tristate”, “NY” * Test input: Select “Load from file button. Select studentList1.txt file. | “Emma Miller 2/28/2003 was already in roster” |

| **Class Name:** TuitionManagerController  **Method Signature:** checkValidEnrollStudentInput() {} //*Validates if the current user input can successfully create a student.* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | The user is required to input the student’s first name. | * User leaves the first name text field blank. * Test input: “” | “Enter a first name for the Student.” |
| 2 | The user is required to input the student’s last name. | * User leaves the last name text field blank. * Test input: “” | “Enter a last name for the Student.” |
| 3 | The user is required to input the date value for the student. | * User leaves the date picker field empty * Test input: “” | “Enter a date of birth.” |
| 4 | The user is required to input the student’s credits completed. | * User leaves the credits completed text field blank. * Test input: “” | “Enter credits completed.” |
| 5 | The user is supposed to input only digits in the credits completed text field. | * User enters a non-digit entry in the credits completed text field. * Test input: “AAA” | “Please only use digits for credit completed” |
| 6 | The user is supposed to input only positive digits in the credits completed text field. | * User enters a negative entry in the credits completed text field. * Test input: “-12” | “Please use positive digits for credit completed” |

| **Class Name:**TuitionManagerController  **Method Signature:** enrollStudent(){} //*Adds a student to enrollment based on user input specifications.* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | Enrolling students with proper credits and input. | * User enters the students First name, Last name, Date of birth, and credits enrolling, * Test input:   + FirstName: “Emma”   + LastName: “Miller”   + Date of birth: 2/28/2003   + Credits: “16” | “Emma Miller 2/28/2003 enrolled 16 credits” |
| 2 | Every student that is Non-Resident being enrolled must have a minimum of 3 credits and maximum of 24 | * User enters the students First name, Last name, Date of birth, and credits enrolling less than 3. * FirstName: “Emma” * LastName: “Miller” * Date of birth: 2/28/2003 * Credits: “2” | “(Non-Resident) 1: invalid credit hours.” |
| 3 | Every student that is Resident being enrolled must have a minimum of 3 credits and maximum of 24 | * User enters the students First name, Last name, Date of birth, and credits enrolling more than 24. * FirstName: “John” * LastName: “Doe” * Date of birth: 1/2/2007 * Credits: “30” | “(Resident) 30: invalid credit hours.” |
| 4 | Every student that is a International student must have a minimum credit of 12 and a maximum credit 24 | * User enters the students First name, Last name, Date of birth, and credits enrolling less than 3. * FirstName: “Bill” * LastName: “Scanlan” * Date of birth: 5/1/1999 * Credits: “8” | “(International student) 8: invalid credit hours.” |
| 5 | Every student that is a International student and studying abroad must have a minimum credit of 3 and a maximum credit of 12 | * User enters the students First name, Last name, Date of birth, and credits enrolling more than 12 * FirstName: “April” * LastName: “March” * Date of birth: 3/31/2000 * Credits: “13” | “(International student: study abroad) 13: invalid credit hours.” |

| **Class Name:**TuitionManagerController  **Method Signature:** dropStudent(){} //*Drop student from enrollment list* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | Dropping student from enroll list if the student is already enrolled with proper input | * User enters the students First name, Last name, Date of birth, and credits enrolling, * Test input:   + FirstName: “John”   + LastName: “Doe”   + Date of birth: 1/2/2007   + Credits: “16” | “John Doe 1/2/2007 dropped.” |
| 2 | Dropping a student from the enrollment list if the student is not in the enrollment list. | * User enters the students First name, Last name, Date of birth, and credits enrolling, * Test input:   + FirstName: “John”   + LastName: “Doe”   + Date of birth: 1/2/2007   + Credits: “16” | “John Doe 1/2/2007 is not enrolled” |

| **Class Name:**TuitionManagerController  **Method Signature:** deleteStudentFromRoster(){} //*Remove student from roster list* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | Remove student from roster list if the student is already registered with proper input. | * User enters the students First name, Last name, Date of birth, and credits completed, * Test input:   + FirstName: “John”   + LastName: “Doe”   + Date of birth: 1/2/2007   + Major: EE   + Credits: “16”   + Status: Resident | “John Doe 1/2/2007 removed from the roster.” |
| 2 | Removing a student from the roster list if the student is not in the roster list. Method will say that the student is not in the roster | * User enters the students First name, Last name, Date of birth, and credits completed, * Test input:   + FirstName: “John”   + LastName: “Doe”   + Date of birth: 1/2/2007   + Credits: “16” | “John Doe 1/2/2007 is not in the roster” |

| **Class Name:**TuitionManagerController  **Method Signature:** changeMajor(){} //*change major of student in the roster class* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | Change the major of a student that is already registered in the roster. | * User enters the students First name, Last name, Date of birth, Select Major button, enter credits completed, and select the status type. We will change his major from EE to CS. * Test input:   + FirstName: “John”   + LastName: “Doe”   + Date of birth: 1/2/2007   + Major: CS   + Credits: “16”   + Status: Resident | “John Doe 1/2/2007 major has been changed!” |
| 2 | Change the major of a student that is not registered in the roster. Assume the student from the test case above has been removed from the roster. | * User enters the students First name, Last name, Date of birth, Select Major button, enter credits completed, and select the status type. We will change his major from EE to CS. * Test input:   + FirstName: “John”   + LastName: “Doe”   + Date of birth: 1/2/2007   + Major: CS   + Credits: “16”   + Status: Resident | “John Doe 1/2/2007 is not in the roster” |

| **Class Name:** TuitionManagerController  **Method Signature:** checkValidScholarStudentInput() {} //*Validates if the current user input can successfully create a student.* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | The user is required to input the student’s first name. | * User leaves the first name text field blank. * Test input: “” | “Enter a first name for the Student.” |
| 2 | The user is required to input the student’s last name. | * User leaves the last name text field blank. * Test input: “” | “Enter a last name for the Student.” |
| 3 | The user is required to input the date value for the student. | * User leaves the date picker field empty * Test input: “” | “Enter a date of birth.” |
| 4 | The user is required to input a scholarship amount. | * User leaves the scholarship field empty * Test input: “” | “Please enter scholarship amount” |
| 5 | The user is also required to input a scholarship amount and it is supposed to be a digit. | * User enters the scholarship amount non-digit * Test input: “AAA” | “Please enter digits for scholarship amount” |

| **Class Name:** TuitionManagerController  **Method Signature:** awardScholarship() {} //*Validates if the current user input can successfully create a student.* | | | |
| --- | --- | --- | --- |
| **Test Case #** | **Requirement** | **Test description and Input Data** | **Expected result/output** |
| 1 | The user can award up to 10000 to resident students. | * User enters first name, last name, date of birth, and scholarship amount. * Test input:   + FirstName: “John”   + LastName: “Doe”   + Date of birth: 1/2/2007   + Amount: 10000 | “John Doe 1/2/2007 :scholarship amount updated” |
| 2 | The user can not award scholarships to any student of 0 or less than 0 amount. | * User enters first name, last name, date of birth, and scholarship amount. * Test input:   + FirstName: “John”   + LastName: “Doe”   + Date of birth: 1/2/2007   + Amount: 0 | Invalid scholarship amount: 0 |
| 3 | User cannot award scholarship to student who is not enrolled | * User enters first name, last name, date of birth, and scholarship amount. * Test input:   + FirstName: “Jane”   + LastName: “Doe”   + Date of birth: 5/1/1996   + Amount: 10000 | “An error has occurred while processing the scholarship award: Cannot invoke "com.example.project3.EnrollStudent.getProfile()" because "enrollment" is null” |
| 4 | If a student is Non-Resident, they cannot get a scholarship. | * User enters first name, last name, date of birth, and scholarship amount. * Test input:   + FirstName: “April”   + LastName: “March”   + Date of birth: 3/31/2000   + Amount: 1000 | “April March 3/31/2000 is not a resident student and is not eligible for a scholarship.” |
| 5 | If a student is Part-time, they cannot get a scholarship. Assume test case 1 student is now Part-time. | * User enters first name, last name, date of birth, and scholarship amount. * Test input:   + FirstName: “John”   + LastName: “Doe”   + Date of birth: 1/2/2007   + Amount: 10000 | “Part-time students are not eligible for scholarship” |