Hyeon Oh

**Project 3 Test Specifications**

**Class Name: TuitionManagerController**

**Method Signature: public void onAddButtonClick() {} //adds student to roster**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result / Output |
| 1 | A student that will be added to the roster should have a first and last name, valid birth date, and valid credit completed. If the student is valid, it will be added to the roster and display on the GUI that the student has been added successfully. | * Fill in information of a student with a first and last name, valid birthday, and valid credits * Test data:   First Name: Bryan  Last Name: Oh  Date of Birth: 6/28/2000  Major: CS  Credits Completed: 78  Status: Resident | Bryan Oh 6/28/2000 added to the roster. |
| 2 | A student that will be added to the roster should have a first and last name, valid birth date, and valid credit completed. If the student does not have a first name, he/she will be considered invalid and will output an error message. | * Fill in information of a student with every form filled except for first name * Test data:   First Name:  Last Name: Oh  Date of Birth: 6/28/2000  Major: CS  Credits Completed: 78  Status: Resident | Date Missing: First Name is Empty! |
| 3 | A student that will be added to the roster should have a first and last name, valid birth date, and valid credit completed. If the student does not have a last name, he/she will be considered invalid and will output an error message. | * Fill in information of a student with every form filled except for last name * Test data:   First Name: Bryan  Last Name:  Date of Birth: 6/28/2000  Major: CS  Credits Completed: 78  Status: Resident | Data Missing: Last Name is Empty |
| 4 | A student that will be added to the roster should have a first name, last name, valid birth date, and valid credits completed. If the student has an empty field in credits completed, he/she will be considered invalid and will output an error message. | * Fill in information of a student that has an empty form of credits completed. * Test data:   First Name: John  Last Name: Smith  Date of Birth: 6/28/2000  Major: CS  Credits Completed:  Status: Resident | Data Missing: Credits Completed is Empty! |
| 5 | A student that will be added to the roster should have a first name, last name, valid birth date, and valid credits completed. If the student has an invalid number of credits completed, he/she will be considered invalid and will output an error message. | * Fill in information of a student with an invalid number of credits completed. * Test data:   First Name: John  Last Name: Smith  Date of Birth: 4/13/2003  Major: CS  Credits Completed: -4  Status: Resident | Credits Invalid: -4 is negative |
| 6 | A student that will be added to the roster should have a first and last name, valid birth date, and valid credits completed. The program should be able to group radio buttons correctly. For Non-Resident students living in the Tri-State, the option to pick whether they live in NY or CT should be enabled while the option to study abroad should be disabled. | * Fill in information of a non-resident student that resides in a tri-state. * Test data:   First Name: John  Last Name: Park  Date of Birth: 2/23/2003  Major: CS  Credits Completed: 60  Status: Non-Resident , Tri-State, NY | John Park 2/23/2003 added to the roster. |
| 7 | A student that will be added to the roster should have a first and last name, valid birth date, and valid credits completed. The program should be able to group radio buttons correctly. For Non-Resident students that are international students, the option to pick whether they live in NY or CT should be disabled while the option to pick study abroad should be enabled. | * Fill in information of a non-resident student that is an international student studying abroad. * Test data:   First Name: John  Last Name: Lee  Date of Birth: 2/23/2003  Major: CS  Credits Completed: 60  Status: Non-Resident, International, Study Abroad | John Lee 2/23/2003 added to the roster. |
| 8 | A student that will be added to the roster should have a first and last name, valid birth date, and valid credits completed. The student should have a valid date of birth to be added to the roster. If the date of birth is empty, there will be an error message. | * Fill in information of a student with an empty date of birth. * Test Data:   First Name: John  Last Name: Lee  Date of Birth:  Major: CS  Credits Completed: 70  Status: Resident | Data Missing: No date selected! |
| 9 | A student that will be added to the roster should have a first and last name, valid birth date, and valid credits completed. The student should have a valid date of birth to be added to the roster. If the date of birth is invalid, there will be an error message. | * Fill in information of a student with an invalid date of birth. * Test Data:   First Name: John  Last Name: Lee  Date of Birth: 3/2/2017  Major: CS  Credits Completed: 70  Status: Resident | Invalid Date: Student is younger than 16 years old |
| 10 | A student that will be added to the roster should have a first and last name, valid birth date, and valid credits completed. The student should have a valid date of birth to be added to the roster. If the credits completed is not a number, there will be an error message. | * Fill in information of a student with an invalid form of credits completed. * Test Data:   First Name: John  Last Name: Lee  Date of Birth: 3/2/2000  Major: CS  Credits Completed: Hi  Status: Resident | Credits Invalid: Hi is not an integer. |
| 11 | A student that will be added to the roster should have a first and last name, valid birth date, and valid credits completed. The student should not be added if he/she already exists in the roster. If the student already exists in the roster, there will be an error message. | * Fill in information of a student twice. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 70  Status: Resident | John Park 3/2/2000 is already in the roster. |

**Class Name: TuitionManagerController**

**Method Signature: public void onDeleteButtonClick() {} //deletes a student from the roster**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | A student that will be deleted from the roster should already exist in the roster. Furthermore, the forms should be filled out correctly so that the student is found for this method to work. | * Fill in an information of a student that already exists and delete him/her. First add a student then delete. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 70  Status: Resident | John Park 3/2/2000 removed from the roster. |
| 2 | A student that will be deleted from the roster should already exist in the roster. If the form is filled out incorrectly and doesn’t match the student, there will be an error message. If the student’s first name form is empty, there will be an error message. | * Fill in information of a student with no first name form filled out. * Test Data:   First Name:  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 70  Status: Resident | Data Missing: First Name is Empty! |
| 3 | A student that will be deleted from the roster should already exist in the roster. If the form is filled out incorrectly and doesn’t match the student, there will be an error message. IF the student’s last name form is empty, there will be an error message. | * Fill in information of a student with no last name form filled out. * Test Data:   First Name: John  Last Name:  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 70  Status: Resident | Data Missing: Last Name is Empty |
| 4 | A student that will be deleted from the roster should already exist in the roster. If the form is filled out incorrectly and doesn’t match the student, there will be an error message. If there is an empty date, there will be an error message. | * Fill in information of a student with an empty date form filled * Test Data:   First Name: John  Last Name: Park  Date of Birth:  Major: CS  Credits Completed: 70  Status: Resident | Data Missing: No date selected! |
| 5 | A student that will be deleted from the roster should already exist in the roster. If the form is filled out incorrectly and doesn’t match the student, there will be an error message. If the date is invalid, there will be an error message. | * Fill in information of a student with an invalid date form filled. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2017  Major: CS  Credits Completed: 70  Status: Resident | Invalid Date: Student is younger than 16 years old |
| 6 | A student that will be deleted from the roster should already exist in the roster. If the form is filled out incorrectly and doesn’t match the student, there will be an error message.  If the student doesn’t exist in the roster, there will be an error message. | * Fill in information of a student that does not exist in the roster. * Test Data:   First Name: James  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 70  Status: Resident | James Park 3/2/2000 is not in the roster. |

**Class Name: TuitionManagerController**

**Method Signature: public void onChangeButtonClick() {} //changes a student’s major**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | A student’s major that will be changed must already exist on the roster and must have a first and last name, and valid birth date. | * Fill in information of a student and then change his/her major. * Test Data:   Add  First Name: Bryan  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Change ->  First Name: Bryan  Last Name: Park  Date of Birth: 3/2/2000  Major: ITI | Bryan Park 3/2/2000 major changed to ITI |
| 2 | A student’s major that will be changed must already exist on the roster and must have a first and last name, and valid birth date.  If the student’s major to be changed does not a first name form filled, there will be an error message. | * Fill in information of a student with no first name form filled * Test data:   First Name:  Last Name: Park  Date of Birth: 3/2/2000  Major: CS | Data Missing: First Name is Empty! |
| 3 | A student’s major that will be changed must already exist on the roster and must have a first and last name, and valid birth date.  If the student’s major to be changed does not a last name form filled, there will be an error message. | * Fill in information of a student with no last name form filled * Test data:   First Name: Bryan  Last Name:  Date of Birth: 3/2/2000  Major: CS | Data Missing: Last Name is Empty! |
| 4 | A student’s major that will be changed must already exist on the roster and must have a first and last name, and valid birth date.  If the student’s date of birth is empty, there will be an error message. | * Fill in information of a student’s major to be changed but have the date form empty. * Test Data:   First Name: Bryan  Last Name: Park  Date of Birth:  Major: CS | Data Missing: No date selected! |
| 5 | A student’s major that will be changed must already exist on the roster and must have a first and last name, and valid birth date.  If the student’s major to be changed does not have a valid birth date, there will be an error message. | * Fill in information of a student with an invalid birth date. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2017  Major: CS | Invalid Date: Student is younger than 16 years old. |
| 6 | A student’s major that will be changed must already exist on the roster and must have a first and last name, and valid birth date.  If a student does not exist on the roster, then there will be an error message. | * Fill in information of a student that does not exist in the roster. * Test Data:   First Name: Leon  Last Name: Lee  Date of Birth: 3/2/2000  Major: CS | Leon Lee 3/2/2000 is not in the roster. |

**Class Name: TuitionManagerController**

**Method Signature: public void onLoadFileButtonClick() {} //loads a file**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | A file name can be loaded. For example, studentList.txt can be loaded to the program.  The form must be filled correctly and file must be in the same folder as the project folder. | * Fill out the text field with the correct name of the file and extension, in our case it is .txt * Test Data:   Text File: studentList.txt | Students loaded to the roster. |
| 2 | A file name can be loaded. For example, studentList.txt can be loaded to the program.  The form must be filled correctly and file must be in the same folder as the project folder. If the form is empty, then there will be an error message. | * Leave the form empty for load * Test Data:   Text File: | Data Missing: Filename is Empty! |
| 3 | A file name can be loaded. For example, studentList.txt can be loaded to the program.  The form must be filled correctly and file must be in the same folder as the project folder. If the form has an invalid name of a file that does not exist, then there will be an error message. | * Fill in invalid information of a file that does on exist. * Test Data:   Text File: oompa | Error: File not Found  Please enter valid textfile (Ex: studentList.txt) |

**Class Name: TuitionManagerController**

**Method Signature: public void onEnrollButtonClick() {} //enrolls a student**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | A student must exist and be in the roster to be enrolled. A student must have a valid number of credits enrolled for this method to work. | * First add a student to the roster and then enroll that student * Test Data:   Add a student to the first.  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 15  Status: Resident  Then on Enroll/Drop tab,  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: 15 | John Park 3/2/2000 enrolled 15 credits |
| 2 | A student must exist and be in the roster to be enrolled. If first name form is empty, then there will be an error message. | * Fill in a student to be enrolled with no first name * Test Data:   First Name:  Last Name: Park  Date of Birth: 3/2/2000  Credits: 15 | Data Missing: First Name is Empty! |
| 3 | A student must exist and be in the roster to be enrolled. If the last name form is empty, then there will be an error message. | * Fill in a student to be enrolled with no last name * Test Data:   First Name: John  Last Name:  Date of Birth: 3/2/2000  Credits: 15 | Data Missing: Last Name is Empty! |
| 4 | A student must exist and be in the roster to be enrolled. If the date of birth is empty, then there will be an error message. | * Fill in a student to be enrolled with no date of birth * Test Data:   First Name: John  Last Name: Park  Date of Birth:  Credits: 15 | Data Missing: No Date selected! |
| 5 | A student must exist and be in the roster to be enrolled. If the credits form is empty, then there will be an error message. | * Fill in a student to be enrolled with an empty credits form. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: | Data Missing: Credits form is Empty! |
| 6 | A student must exist and be in the roster to be enrolled. If the credits enrolled is an invalid amount, then there will be an error message. | * Fill in a student to be enrolled with invalid credits enrolled number. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: 2 | (Resident) 2: Invalid credit hours. |
| 7 | A student must exist and be in the roster to be enrolled. If the credits form has an invalid data type, there will be an error message. | * Fill in a student to be enrolled with invalid data type in credits form field. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: Hi | Credits Enrolled is not an integer |
| 8 | A student must exist and be in the roster to be enrolled. International students must enroll at least 12 credits if they are not studying abroad. | * Fill in an international student that is not studying abroad to be enrolled with invalid credit amount * Test Data:   First Name: John  Last Name: Lee  Date of Birth: 3/2/2000  Credits: 8 | (International Student) 8: invalid credit hours. |
| 9 | A student must exist and be in the roster to be enrolled. International students in the study abroad program can only have a maximum of up to 12 credits enrolled. | * Fill in information of an international student that has an invalid credits enrolled amount of over 12 credits * Test Data:   First Name: John  Last Name: Yang  Date of Birth: 3/2/2000  Credits: 13 | (International student study abroad) 13: invalid credit hours. |
| 10 | A student must exist and be in the roster to be enrolled. If the student does not exist in the roster, there will be an error message in the program. | * Fill in information of a student that does not exist to be enrolled. * Test Data:   First Name: Eric  Last Name: Park  Date of Birth: 3/2/2000  Credits: 15 | Cannot enroll: Eric Park 3/2/2000 is not in the roster. |

**Class Name: TuitionManagerController**

**Method Signature: public void onDropButtonClick() {} //drops a student from enrollment**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | A student in the enrollment list can be dropped if the student exists and is on the list. The student’s name and date of birth is taken into matter when dropping. | * First, add a student to the roster. Then add a student to be enrolled. Then drop that student. * Test Data:   To add:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 15  Status: Resident  To Enroll:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: 15  To Drop:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: | John Park 3/2/2000 dropped. |
| 2 | A student in the enrollment list can be dropped if the student exists and is on the list. The student’s name and date of birth is taken into matter when dropping.  If the student to be dropped has the first name field empty, then there will be an error message. | * Fill in information of a student to be dropped that has an empty first name field. * Test Data:   First Name:  Last Name: Park  Date of Birth: 3/2/2000 | Data Missing: First Name is Empty! |
| 4 | A student in the enrollment list can be dropped if the student exists and is on the list. The student’s name and date of birth is taken into matter when dropping.  If the student to be dropped has the last name field empty, then there will be an error message. | * Fill in information of a student to be dropped that has an empty first name field. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000 | Data Missing: Last Name is Empty! |
| 5 | A student in the enrollment list can be dropped if the student exists and is on the list. The student’s name and date of birth is taken into matter when dropping.  If the student to be dropped has an empty date of birth field, there will be an error message. | * Fill in information of a student to be dropped that has an empty date of birth field. * Test Data:   First Name: John  Last Name: Park  Date of Birth: | Data Missing: No date selected! |
| 6 | A student in the enrollment list can be dropped if the student exists and is on the list. The student’s name and date of birth is taken into matter when dropping.  If the student is not in the enrollment list, then that student cannot be dropped. | * Fill in an information of a student that is not enrolled. * Test Data:   First Name: Mike  Last Name: Lee  Date of Birth: 3/2/2000 | Mike Lee 3/2/2000 is not enrolled. |

**Class Name: TuitionManagerController**

**Method Signature: public void onUpdateScholarshipButtonClick() {} //updates scholarship**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | A student must exist for them to receive a scholarship. A maximum of $10,000 can be awarded to a resident student. | * First add a student to the roster. Then enroll the student, then update scholarship amount in the third tab. * Test Data:   To add student to roster:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 15  Status: Resident  Then enroll the student.  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: 15  Then update scholarship:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Amount: 1000 | John Park 3/2/2000 enrolled 15 credits. |
| 2 | A student must exist for them to receive a scholarship. A maximum of $10,000 can be awarded to a resident student. If a student to receive a scholarship has an invalid scholarship amount that is not within the range of 1-10000 in the form field, then there will be an error message. | * Fill in information of a student with invalid scholarship amount. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Amount: 10001 | 10001: invalid amount |
| 3 | A student must exist for them to receive a scholarship. A maximum of $10,000 can be awarded to a resident student. If the scholarship form field is empty, then there will be an error message. | * Fill in information of a student with an empty scholarship amount form field. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Amount: | Data Missing: Scholarship amount is Empty! |
| 4 | A student must exist for them to receive a scholarship. A maximum of $10,000 can be awarded to a resident student. If the student first name form field is empty, then there will be an error message. | * Fill in information of a student that is to receive a scholarship with an empty first name form field. * Test Data:   First Name:  Last Name: Park  Date of Birth: 3/2/2000  Amount: 1000 | Data Missing: First Name is Empty! |
| 5 | A student must exist for them to receive a scholarship. A maximum of $10,000 can be awarded to a resident student. If the student last name field is empty, then there will be an error message. | * Fill in information of a student that is to receive a scholarship with an empty last name form field. * Test Data:   First Name: John  Last Name:  Date of Birth: 3/2/2000  Amount: 1000 | Data Missing: Last Name is Empty! |
| 6 | A student must exist for them to receive a scholarship. A maximum of $10,000 can be awarded to a resident student. If the student’s date of birth form field is empty, there will be an error message. | * Fill in information of a student that is to receive a scholarship with an empty date of birth form field. * Test Data:   First Name: John  Last Name: Park  Date of Birth:  Amount: 1000 | Data Missing: No date selected! |
| 7 | A student must exist for them to receive a scholarship. A maximum of $10,000 can be awarded to a resident student. If the student does not exist, then there will be an error message. | * Fill in information of a student that does not exist. * Test Data:   First Name: Edward  Last Name: Lee  Date of Birth: 3/2/2000  Amount: 1000 | Edward Lee 3/2/2000 is not in the roster. |

**Class Name: TuitionManagerController**

**Method Signature: public void onNormalPrintButtonClick() {} //prints by profile**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | On Clicking Roster Print By Profile, the student roster will be sorted by last name, first name, and date of birth. | * First, load the studentList.txt file. Then print by profile. * Test Data:   Load from File:  Text File: studentList.txt | \*\* Student roster sorted by last name, first name, DOB \*\*  Roy Brooks 8/8/1999 (01:640 MATH SAS) credits completed: 111 (Senior)(non-resident)(tri-state:CT)  Roy Brooks 9/9/1999 (04:547 ITI SC&I) credits completed: 100 (Senior)(non-resident)  Carl Brown 10/7/2004 (01:198 CS SAS) credits completed: 75 (Junior)(resident)  Jane Doe 5/1/1996 (01:198 CS SAS) credits completed: 30 (Sophomore)(resident)  John Doe 1/20/2003 (33:136 BAIT RBS) credits completed: 60 (Junior)(non-resident)  John Doe 4/3/2003 (01:198 CS SAS) credits completed: 29 (Freshman)(non-resident)(international)  John Doe 12/20/2004 (14:332 EE SOE) credits completed: 45 (Sophomore)(non-resident)(tri-state:NY)  John Doe 1/2/2007 (14:332 EE SOE) credits completed: 20 (Freshman)(resident)  Duke Ellington 2/29/2004 (01:640 MATH SAS) credits completed: 46 (Sophomore)(non-resident)(tri-state:NY)  Kate Lindsey 7/15/2002 (04:547 ITI SC&I) credits completed: 59 (Sophomore)(non-resident)  Mary Lindsey 12/1/2001 (33:136 BAIT RBS) credits completed: 89 (Junior)(non-resident)(international)  April March 3/31/2000 (33:136 BAIT RBS) credits completed: 0 (Freshman)(non-resident)(international:study abroad)  Bill Scanlan 5/1/1999 (01:198 CS SAS) credits completed: 120 (Senior)(non-resident)(international)  Paul Siegel 5/1/2000 (01:198 CS SAS) credits completed: 90 (Senior)(resident)  \* End of Roster \* |
| 2 | On Clicking Roster Print By Profile, the student roster will be sorted by last name, first name, and date of birth.  If there is no roster, then there will be an error when trying to produce a print by roster. | * This time, no studentList.txt file will be loaded. * Test Data:   None | Roster is Empty! |

**Class Name: TuitionManagerController**

**Method Signature: public void onMajorPrintButtonClick() {} //prints the roster sorted by school, major**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | On clicking roster print by school and major, the student roster will be sorted by school and major. | * First, load the studentList.txt file. Then print by school and major. * Test Data:   Load From File:  Text File:  studentList.txt | \*\* Student roster sorted by school, major \*\*  John Doe 1/20/2003 (33:136 BAIT RBS) credits completed: 60 (Junior)(non-resident)  Mary Lindsey 12/1/2001 (33:136 BAIT RBS) credits completed: 89 (Junior)(non-resident)(international)  April March 3/31/2000 (33:136 BAIT RBS) credits completed: 0 (Freshman)(non-resident)(international:study abroad)  Carl Brown 10/7/2004 (01:198 CS SAS) credits completed: 75 (Junior)(resident)  Jane Doe 5/1/1996 (01:198 CS SAS) credits completed: 30 (Sophomore)(resident)  John Doe 4/3/2003 (01:198 CS SAS) credits completed: 29 (Freshman)(non-resident)(international)  Bill Scanlan 5/1/1999 (01:198 CS SAS) credits completed: 120 (Senior)(non-resident)(international)  Paul Siegel 5/1/2000 (01:198 CS SAS) credits completed: 90 (Senior)(resident)  Roy Brooks 8/8/1999 (01:640 MATH SAS) credits completed: 111 (Senior)(non-resident)(tri-state:CT)  Duke Ellington 2/29/2004 (01:640 MATH SAS) credits completed: 46 (Sophomore)(non-resident)(tri-state:NY)  Roy Brooks 9/9/1999 (04:547 ITI SC&I) credits completed: 100 (Senior)(non-resident)  Kate Lindsey 7/15/2002 (04:547 ITI SC&I) credits completed: 59 (Sophomore)(non-resident)  John Doe 12/20/2004 (14:332 EE SOE) credits completed: 45 (Sophomore)(non-resident)(tri-state:NY)  John Doe 1/2/2007 (14:332 EE SOE) credits completed: 20 (Freshman)(resident)  \* End of Roster \* |
| 2 | On clicking roster print by school and major, the student roster will be sorted by school and major. If there is no roster, there will be an error message. | * This time, no studentList.txt will be loaded * Test Data:   None | Roster is Empty! |

**Class Name: TuitionManagerController**

**Method Signature: public void onStandingPrintButtonClick() {} //student roster sorted by standing**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | On clicking roster print by standing, the student roster will be sorted by standing. | * First, load the studentList.txt file. Then print by school and major. * Test Data:   Load From File:  Text File:  studentList.txt | \*\* Student roster sorted by standing \*\*  April March 3/31/2000 (33:136 BAIT RBS) credits completed: 0 (Freshman)(non-resident)(international:study abroad)  John Doe 4/3/2003 (01:198 CS SAS) credits completed: 29 (Freshman)(non-resident)(international)  John Doe 1/2/2007 (14:332 EE SOE) credits completed: 20 (Freshman)(resident)  John Doe 1/20/2003 (33:136 BAIT RBS) credits completed: 60 (Junior)(non-resident)  Mary Lindsey 12/1/2001 (33:136 BAIT RBS) credits completed: 89 (Junior)(non-resident)(international)  Carl Brown 10/7/2004 (01:198 CS SAS) credits completed: 75 (Junior)(resident)  Bill Scanlan 5/1/1999 (01:198 CS SAS) credits completed: 120 (Senior)(non-resident)(international)  Paul Siegel 5/1/2000 (01:198 CS SAS) credits completed: 90 (Senior)(resident)  Roy Brooks 8/8/1999 (01:640 MATH SAS) credits completed: 111 (Senior)(non-resident)(tri-state:CT)  Roy Brooks 9/9/1999 (04:547 ITI SC&I) credits completed: 100 (Senior)(non-resident)  Jane Doe 5/1/1996 (01:198 CS SAS) credits completed: 30 (Sophomore)(resident)  Duke Ellington 2/29/2004 (01:640 MATH SAS) credits completed: 46 (Sophomore)(non-resident)(tri-state:NY)  Kate Lindsey 7/15/2002 (04:547 ITI SC&I) credits completed: 59 (Sophomore)(non-resident)  John Doe 12/20/2004 (14:332 EE SOE) credits completed: 45 (Sophomore)(non-resident)(tri-state:NY)  \* End of Roster \* |
| 2 | On clicking roster print by standing, the student roster will be sorted by standing. If there is no roster, there will be an error message. | * This time, no studentList.txt will be loaded * Test Data:   None | Roster is Empty! |

**Class Name: TuitionManagerController**

**Method Signature: public void onRBSClick() {} //shows student in RBS**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | On clicking School/College dropdown menu and clicking RBS, the program will show the students that attend RBS. | * First, load the studentList.txt file. Then print by school and major. * Test Data:   Load From File:  Text File:  studentList.txt | \*\* Students in RBS \*\*  John Doe 1/20/2003 (33:136 BAIT RBS) credits completed: 60 (Junior)(non-resident)  Mary Lindsey 12/1/2001 (33:136 BAIT RBS) credits completed: 89 (Junior)(non-resident)(international)  April March 3/31/2000 (33:136 BAIT RBS) credits completed: 0 (Freshman)(non-resident)(international:study abroad)  \* End of List \* |
| 2 | On clicking School/College dropdown menu and clicking RBS, the program will show the students that attend RBS. If the roster is empty, then there will be an error message. | * This time, no studentList.txt will be loaded * Test Data:   None | Roster is Empty! No students in any school. |
| 3 | On clicking School/College dropdown menu and clicking RBS, the program will show the students that attend RBS. If there are no students in RBS, there will be an error message. | * Add a student to the roster who is not in RBS, then click print rbs. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 15  Status: Resident | No students in RBS! |

**Class Name: TuitionManagerController**

**Method Signature: public void onSOEClick() {} //prints students that are in SOE**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | On clicking School/College dropdown menu and clicking SOE, the program will show students that are in SOE. | * First, load the studentList.txt file. Then print by school and major. * Test Data:   Load From File:  Text File:  studentList.txt | \*\* Students in SOE \*\*  John Doe 12/20/2004 (14:332 EE SOE) credits completed: 45 (Sophomore)(non-resident)(tri-state:NY)  John Doe 1/2/2007 (14:332 EE SOE) credits completed: 20 (Freshman)(resident)  \* End of List \* |
| 2 | On clicking School/College dropdown menu and clicking SOE, the program will show students that are in SOE.  If the roster is empty, then there will be an error message. | * This time, no studentList.txt will be loaded * Test Data:   None | Roster is Empty! No students in any school. |
| 3 | On clicking School/College dropdown menu and clicking SOE, the program will show the students that are in SOE. If there are no students in SOE, there will be an error message. | * Add a student to the roster who is not in SOE, then print SOE. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 15  Status: Resident | No students in SOE! |

**Class Name: TuitionManagerController**

**Method Signature: public void onSASClick() {} //shows students that are in SAS**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | On clicking School/College dropdown menu and clicking SAS, the program will show students that are in SAS. | * First, load the studentList.txt file. Then print by school and major. * Test Data:   Load From File:  Text File:  studentList.txt | \*\* Students in SAS \*\*  Roy Brooks 8/8/1999 (01:640 MATH SAS) credits completed: 111 (Senior)(non-resident)(tri-state:CT)  Carl Brown 10/7/2004 (01:198 CS SAS) credits completed: 75 (Junior)(resident)  Jane Doe 5/1/1996 (01:198 CS SAS) credits completed: 30 (Sophomore)(resident)  John Doe 4/3/2003 (01:198 CS SAS) credits completed: 29 (Freshman)(non-resident)(international)  Duke Ellington 2/29/2004 (01:640 MATH SAS) credits completed: 46 (Sophomore)(non-resident)(tri-state:NY)  Bill Scanlan 5/1/1999 (01:198 CS SAS) credits completed: 120 (Senior)(non-resident)(international)  Paul Siegel 5/1/2000 (01:198 CS SAS) credits completed: 90 (Senior)(resident)  \* End of List \* |
| 2 | On clicking School/College dropdown menu and clicking SAS, the program will show students that are in SAS.  If the roster is empty, then there will be an error message. | * This time, no studentList.txt will be loaded * Test Data:   None | Roster is Empty! No students in any school. |
| 3 | On clicking School/College dropdown menu and clicking SAS, the program will show the students that are in SAS. If there are no students in SAS, there will be an error message. | * Add a student to the roster who is not in SOE, then print SOE. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: EE  Credits Completed: 15  Status: Resident | No students in SAS! |

**Class Name: TuitionManagerController**

**Method Signature: public void onSCIClick() {} //shows students in SCI**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Specification and Input Data | Expected Result/Output |
| 1 | On clicking School/College dropdown menu and clicking SCI, the program will show students that are in SCI. | * First, load the studentList.txt file. Then print by school and major. * Test Data:   Load From File:  Text File:  studentList.txt | \*\* Students in SC&I \*\*  Roy Brooks 9/9/1999 (04:547 ITI SC&I) credits completed: 100 (Senior)(non-resident)  Kate Lindsey 7/15/2002 (04:547 ITI SC&I) credits completed: 59 (Sophomore)(non-resident)  \* End of List \* |
| 2 | On clicking School/College dropdown menu and clicking SCI, the program will show students that are in SCI.  If the roster is empty, then there will be an error message. | * This time, no studentList.txt will be loaded * Test Data:   None | Roster is Empty! No students in any school. |
| 3 | On clicking School/College dropdown menu and clicking SCI, the program will show students that are in SCI.  If there are no students in SCI, there will be an error message. | * Add a student to the roster who is not in SCI, then print SCI. * Test Data:   First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 15  Status: Resident | No students in SC&I! |

**Class Name: TuitionManagerController**

**Method Signature: public void onEnrollmentPrintClick() {} //displays enrolled students**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | On clicking on the Enrollment dropdown menu and clicking print enrolled students, the program will show students that are enrolled. | * First add a student to the roster and then enroll him/her. Then click on the enrollment dropdown menu and click print enrolled students. * Test Data:   To add to roster:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 15  Status: Resident  To Enroll:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: 15 | \*\* Enrollment \*\*  John Park 3/2/2000: credits enrolled: 15  \* End of Enrollment \* |
| 2 | On clicking on the Enrollment dropdown menu and clicking print enrolled students, the program will show students that are enrolled. If there are no students enrolled, then there will be an error message. | * Simply launch the program * Test Data:   None | Enrollment is Empty! |

**Class Name: TuitionManagerController**

**Method Signature: public void onTuitionDueClick() {} //shows tuition due for students enrolled**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | On clicking on the Enrollment dropdown menu and clicking Print Tuition Dues, the program will show the tuition due for each student enrolled. | * First add a student to the roster and then enroll him/her. Then click on the enrollment dropdown menu and click print tuition dues. * Test Data:   To add to roster:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 15  Status: Resident  To Enroll:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: 15 | \*\* Tuition Due \*\*  John Park 3/2/2000 (Resident) enrolled 15 credits: tuition due: $15,804.00  \* End of Tuition Due \* |
| 2 | On clicking on the Enrollment dropdown menu and clicking print tuition dues, the program will show students that are enrolled. If there are no students enrolled, then there will be an error message. | * Simply launch the program * Test Data:   None | Enrollment is Empty! |

**Class Name: TuitionManagerController**

**Method Signature: public void onSemesterEndClick() {} //adds the enrolled credits to the credits completed in the roster and prints out students that have completed 120 credits or more**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Requirement | Test Description and Input Data | Expected Result/Output |
| 1 | On clicking on the Enrollment dropdown menu and clicking Semester End, the program will add the enrolled credits to the credits completed in to the roster and prints out students that have completed 120 credits or more. | * First add a student to the roster and then enroll him/her. Then click on the enrollment dropdown menu and click semester end. * Test Data:   To add to roster:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Major: CS  Credits Completed: 110  Status: Resident  To Enroll:  First Name: John  Last Name: Park  Date of Birth: 3/2/2000  Credits: 15 | Credit completed has been updated.  \*\* list of students eligible for graduation \*\*  John Park 3/2/2000 (01:198 CS SAS) credits completed: 125 (Senior)(resident)  \* end of list \* |
| 2 | On clicking on the Enrollment dropdown menu and clicking Semester End, the program will add the enrolled credits to the credits completed in to the roster and prints out students that have completed 120 credits or more. If there are no students enrolled, there will be an error message. | * Simply launch the program * Test Data:   None | Enrollment is Empty! |