Harpreet Randhawa, Mikita Belausau

Professor Chang

Software Methodology

22 September 2021

GUI Test Design Document

|  |  |  |  |
| --- | --- | --- | --- |
| **Java class:** public class MainController | | | |
| **Method signature:** void addStudentToRoster(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Should be able to add international students to the roster | User selects international student residency and adds a student to roster.  Test Input: User inputs name, selects major, selects international residency and selects if this student is studying abroad on GUI then clicks add student. | Expected result: “student has been added to roster”  The output is “student has been added to roster” so this passes. |
| 2 | Should be able to add resident students | User selects resident student residency and adds a student to roster.  Test Input: User inputs name, selects major, selects resident residency on GUI then clicks add student. | Expected result: “student has been added to roster”  The output is “student has been added to roster” so this passes. |
| 3 | Should be able to add tristate Students | User selects triState student residency and adds a student to roster.  Test Input: User inputs name, selects major, selects tristate, selects NY or CT residency on GUI then clicks add student. | Expected result: “student has been added to roster”  The output is “student has been added to roster” so this passes. |
| 4 | When you add an international student if they have less than 12 credits it should tell you that you cannot | User enters a name, selects a major, selects international student and puts in 5 credits  Test input: 11,12 | The output is “international students must enroll with 12 credits or more” which is the expected result for input of 11.  The output for 12 is “international student is added to roster” |
| 5 | Adding an international student with anything other than 12 credits should tell you that you cannot | User enters a name, selects a major, selects international, and selects study abroad, puts in x credits  Test input: 11,12,13 | The output is “International students who are studying abroad need to have exactly 12 credits” for inputs 11 and 13 which is expected. For 12 credits it outputs “international student added to roster” which is the expected output. |
| 6 | Make sure grow works | Add 5 students | All of the outputs should be “student added to roster”  Output: “student added to roster x5” |
|  | | | |
| **Method signature:** void deleteStudentFromRoster(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Should be able to delete international students from the roster | User inputs the name of a tristate student and selects their major and pressed delete from roster button.  Test Input: User inputs name of existing international student, selects that existing students major then clicks remove student. | Expected result: “student has been removed from roster”  The output is “student has been removed from roster” so this passes. |
| 2 | Should be able to delete resident students from the roster | User inputs the name of a tristate student and selects their major and pressed delete from roster button.  Test Input: User inputs name of existing international student, selects that existing students major then clicks remove student. | Expected result: “student has been removed from roster”  The output is “student has been removed from roster” so this passes. |
| 3 | Should be able to remove tristate students | User inputs the name of a tristate student and selects their major and pressed delete from roster button.  Test Input: User inputs name of existing tristate student, selects that existing students major then clicks remove student. | Expected result: “student has been removed from roster”  The output is “student has been removed from roster” so this passes. |
| 4 | Students that aren’t in the roster should not be deleted | Try to delete a student that is not part of the roster | Expected: “Student is not in the roster”  Output: “Student is not in the roster” |
| **Method signature:** public boolean checkIfValidStudentInput(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Must enter a name to perform all operations on views that have a name field. | If user doesn’t input a name it should print an error message.  Test input: don’t enter a name and press add student/delete student etc. | Expected Result: “Please enter a name for the student”  The output is “please enter a name for the student” |
| 2 | Student name should be comprised of only letters and spaces. | If user inputs a name that has anything but letters and spaces it will print an error message.  Test input: enter a name like john2doe and press add student/delete student. | Expected result: “Please use only characters for student name”  The output is “Please use only characters for student name” |
| 3 | User must select a major to perform any operations that include a student | If user doesn’t select a major it will print an error message if you are trying to do student related operations other than printing.  Test input: enter a name John Doe, then don’t select a major and press add student/delete student. | Expected result: “Please select a major.”  Output: “Please select a major.” |
| 4 | User must select a form of residency to perform student operations for adding | If user doesn’t select a residency it will print an error message.  Test input: Enter a name John Doe, then select a major CS, and press add student. | Expected Result: “Please select a form of residency.”  The output is “Please select a form of residency” |
| 5 | Student credit hours need to be between 3 and 24 credits inclusively. | If user inputs credit hours that are less than 3 or more than 24 it should return “credit hours cannot be more than 24” or “credits hours must be greater than 2”  Test input: Enter all student data and enter in 2, 25, 3, then switch majors and enter 24 for student cre dits. | Expected result: “Please use only credit hours greater than 2, Please use only credit hours less than 25, Student added, Student added.”  Output: “Please use only credit hours greater than 2, Please use only credit hours less than 25, Student added, Student added.” |
| 6 | Entering in anything but a number into student credit hours will result in a NumberFormatException | If User enters in anything but a number into student credit hours it will print an error message.  Test input: Enter in all student info and for credit hours enter in “Char{=…” or leave it blank etc. | Expected result: “You must use only credit hours between 2 and 24”  Output: “you must use only credit hours between 2 and 24.” |
| **Method signature:** void setStudyAbroadStatus(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Must set a student who is not studying abroad to studying abroad, set his credits to 12 and recalculate his tuition | Add an international student who is not studying abroad with 12 credits his tuition should be 35k+ if you calculate it, then for that same student make sure to set his study abroad status using the button | Expected: Adds the student and resets his credits to 12 and his tuition will go from 35k+ -> 5k+  Output: International student is now a study abroad international student with the correct tuition input and the correct credit amount. |
| 2 | If a student is already studying abroad setting his status will return that he’s already studying abroad | Try to set a study abroad student to study abroad | Expected: “Student is already studying abroad”  Output: “Student is already studying abroad” |
| 3 | Must return an error message if the student trying to be set to study abroad is not an international student | Try to set study abroad for a student who is not international | Expected: “Student is not an international student”  Output: “Student is not an international student” |
| 4 | Must return an error message if the student you are trying to set to study abroad is not in the roster | Try to set study abroad for a student who is not in the roster | Expected: “International student not found”  Output: “International student not found” |
| **Method signature:** void performNotStudyAbroad(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Must be able to set study abroad students to not study abroad | Takes an international student that is studying abroad and sets their status to not studying abroad | Expected: Adds the student and resets his credits to 12 and his tuition will go from 35k+ -> 5k+  Output: International student is now a study abroad international student with the correct tuition input and the correct credit amount. |
| 2 | If an international student is already studying abroad then it will return student is already studying abroad | Try to set notStudyAbroad for a student who is already not studying abroad | Expected: “Student is not studying abroad”  Output: “Student is not studying abroad” |
| 3 | If the student is not an international student then performNotStudyAbroad should provide an error message | Set notStudyAbroad for a student who is not an international student | Expected: “student is not an international student”  Output: “student is not an international student” |
| 4 | If the student is not in the roster it must display an error message if you performNotStudyAbroad | Set notStudyAbroad for a student not in the roster | Expected: “International student not found”  Output: “International student not found” |
| **Method signature:** void displayRoster(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Must print student roster | Add a bunch of students to the roster like John Doe, CS, Resident, 15  John Doe, ME, International study abroad, 12  John Doe, IT, TriState NY, 15  John Doe, BA, International, 13 | Expected: John Doe:CS:15 credit hours:tuition due:0.00:total payment:0.00:last payment date: --/--/--:resident  John Doe:ME:12 credit hours:tuition due:0.00:total payment:0.00:last payment date: --/--/--:non-resident:international:study abroad  John Doe:IT:15 credit hours:tuition due:0.00:total payment:0.00:last payment date: --/--/--:non-resident:(tri-state):NY  John Doe:BA:13 credit hours:tuition due:0.00:total payment:0.00:last payment date: --/--/--:non-resident:international  Output: Matches the above. |
| 2 | Must tell the user that no students are in the roster if they try to print an empty roster | Print without adding any students | Expected: “No student in roster”  Output: “No student in roster” |

|  |  |  |  |
| --- | --- | --- | --- |
| **Method signature:** void displayRosterByName(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Must print student roster by name order | Add a bunch of students to the roster like Zombie, CS, Resident, 15  Zombie, ME, International study abroad, 12  Apple, IT, TriState NY, 15  Apple, BA, International, 13 | Expected: For the names to be apple – apple – zombie – zombie instead of how they were originally inputed with all of the correct sentence structure.  Output: Matches the above. |
| 2 | Must tell the user that no students are in the roster if they try to print an empty roster | Print without adding any students | Expected: “No student in roster”  Output: “No student in roster” |
| **Method signature:** void displayRosterByPayment(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Must print student roster by order in which they paid and if they didn’t pay they must not be included in the list | Add a bunch of students to the roster like Zombie, CS, Resident, 15  Zombie, ME, International study abroad, 12  Apple, IT, TriState NY, 15  Apple, BA, International, 13  Make apple pay on 8/23/2021, Make Zombie pay on 8/27/2021, and make the others not pay | Expected: Only display apple-zombie in that order because apple paid earlier than zombie, of course with the correct sentence syntax and with their payment dates shown.  Output: Matches the above. |
| 2 | Must tell the user that no students are in the roster if they try to print an empty roster | Print without adding any students | Expected: “No student in roster”  Output: “No student in roster” |
| **Method signature:** General GUI interaction for tab1 | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Not allowed to have multiple majors, residencies | Radio button group for both of these things on tab1 | Doesn’t allow you to selected more than one major or residency. |
| 2 | Only tristate can select NY or CT in residency | Try to press ny or ct without tristate being highlighted | Hides NY and CT at the start and its only enabled if you actively press tristate |
| 3 | Only international students can select study abroad status | Try to press study abroad without pressing international | Hides study abroad unless international is also pressed. |
| **Method signature:** General GUI interaction for tab4 | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Be able to press display roster, display roster by name, display roster by payment date | They all press just fine | Expected: Pressable  Output: Pressable |
| **Method signature:** General GUI interaction for tab2 | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Be able to press financial aid, update financial aid, and payment | They all press and work correctly | Expected: Pressable  Output: Pressable |
| 2 | Financial aid, update financial aid, and payment are all disabled at the start | They are all disabled like how they should be | Expected: Disabled  Output: Disabled |
| 3 | When clicking on the yes full time resident button, the three buttons become enabled | All buttons become enabled and no longer disabled | Expected: Enabled  Output: Enabled |
| 4 | When clicking on the no full time resident button, the three buttons become enabled | The financial aid and update financial aid buttons don’t become enabled, but the payment button becomes enabled | Financial aid and update financial aid buttons are disabled. Payment button is enabled |
| **Method signature:** General GUI interaction for tab3 | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | The calculate student and calculate roster buttons are disabled at the start | They are all disabled like how they should be | Expected: Disabled  Output: DEisabled |
| 2 | Be able to press calculate student and calculate roster buttons | They all press and work correctly | Expected: Pressable  Output: Pressable |
| 3 | When clicking on the option 1 radio button the calculate student should become enabled and the calculate roster should stay disabled | Calculate student button becomes enabled and the calculate roster button stays disabled | Calculate student: Enabled  Calculate roster: Disabled  Correct output |
| **Method signature:** void calculateEntireRosterTuition(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | First check the edge case where there are no students in the roster and test that it will not calculate since the roster is empty | Testing the calculate roster button when the roster has no students inside | The expected result should be that no calculation is performed. This is indeed what happens with the output “Roster is empty nothing to calculate” |
| 2 | Check to see if the roster will be calculated if there is only 1 student in the roster | Testing the calculate roster button when the roster contains 1 student | The expected output should be that the calculate roster should still be calculated for the 1 student in which it is. It properly calculates the student with an output of “Roster tuition calculated” |
| 3 | Check to see if the roster will be calculated for many students in the roster | Testing the calculate roster button when the roster contains greater than 1 student | The expected output should be that the calculate roster should still be calculated for all the students given in the roster. It does properly calculate all the students with the output of “Roster tuition calculated” |
| 4 | Check to see if the roster will be calculated for a roster that has been filled, but then all deleted afterwards | Testing the calculate roster button when the roster contains greater than 1 student, but then all being deleted afterwards | The expected result should be that no calculation is performed. This is indeed what happens with the output “Roster is empty nothing to calculate” |
| **Method signature:** void calculateOneStudentTuition(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Check to see if the calculate student will calculate a student properly with the proper credits and requirements | Testing the calculate student to properly calculate their tuition | As expected, the student’s tuition is properly calculated with an output of “Student’s tuition was calculated successfully” |
| 2 | Check to see if the calculate student will calculate a student who has already been calculated once | This to check if the calculation will be done over and over again for the same student in which it should only calculate it once | The student’s tuition is not calculated again as the student’s tuition has already been calculated once. With an output of “Students tuition already calculated” |
| 3 | Check to see if the calculate student will work with the name text box having the incorrect name | This will be testing if the calculate student function is realize that the given name is not in the roster | As expected, the button recognizes that the student is not in the roster due to the incorrect name and therefore returns “Student does not exist in the roster” |
| 4 | Check to see if the calculate student will work with the incorrect major being chose | This will be testing if the calculate student function realizes that the given major is not correct | As expected, the button recognizes that the student is not in the roster due to the incorrect major and therefore returns “Student does not exist in the roster” |
| 5 | Check to see if the calculate student will work if the name box is empty | This will be testing if the calculate student function realizes that the given name is empty | As expected, the button recognizes that the name is empty and therefore returns “Enter a name for the student” |
| 6 | Check to see if the calculate student will work if none of the major options are chosen | This will be testing if the calculate student function is realizes that no major was chosen | Returns “Please select a major” |
| **Method signature:** void calculatePayment(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Check to see if the payment button works if the tuition for the given student has not been calculated yet | Testing if the button will work if the tuition has not been calculated yet | Returns ‘Student’s tuition must be calculated first” |
| 2 | Check to see if the payment button works if the payment text box is empty | Testing if the button will work if the payment text box is empty | Returns “Please only use numbers in the payment amount text field” |
| 3 | Check to see if the payment button works if the payment text box has a negative number | Testing if the button will work if the payment text box has a negative number | Returns “Please enter a non-negative payment amount” |
| 4 | Check to see if the payment button works if the payment text box has a 0 | Testing when the payment text box contains a 0 | Returns “Please enter payment amount greater than zero” |
| 5 | Check to see if the payment button works the payment date is not selected | Testing when the payment date is empty | Returns “Must choose a payment date” |
| 6 | Check to see if the payment button works when everything is chosen properly with a proper student | Testing to see if the button works with the proper inputs | Returns “Tuition updated” |
| 7 | Check to see if the payment button works when the given student is not a valid student | Testing to see if the button detects that the student is not proper | Returns “Student not found” |
| 8 | Check to see if the payment button works when the given major is not a valid major | Testing to see if the button detects that the major is not proper | Returns “Student not found” |
| 9 | Check to see if the given payment amount is not more than the tuition due | The tuition due must be greater because you cannot pay more than what is due | Returns “Cannot make a payment greater than the tuition due” |
| **Method signature:** void financialAid(ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Check to see if the financial aid button works when the given major is not a valid major | Testing to see if the button detects that the major is not proper | Returns “Student not found” |
| 2 | Check to see if the financial aid button works when the given student is not a valid student | Testing to see if the button detects that the student is not proper | Returns “Student not found” |
| 3 | Check to see if the financial aid button works when everything is chosen properly with a proper student | Testing to see if the button works with the proper inputs | Returns “Tuition updated” |
| 4 | Check to see if the financial aid button works the payment date is not selected | The button should still work because the financial aid does not require a payment date | Returns “Tuition updated” |
| 5 | Check to see if the financial aid button works if the payment text box has a 0 | Testing when the payment text box contains a 0 | Returns “Please enter payment amount greater than zero” |
| 6 | Check to see if the financial aid button works if the payment text box has a negative number | Testing if the button will work if the payment text box has a negative number | Returns “Please enter a non-negative payment amount” |
| 7 | Check to see if the financial aid button works if the payment text box is empty | Testing if the button will work if the payment text box is empty | Returns “Please only use numbers in the payment amount text field” |
| 8 | Check to see if the financial aid button works if the tuition for the given student has not been calculated yet | Testing if the button will work if the tuition has not been calculated yet | Returns ‘Student’s tuition must be calculated first” |
| 9 | Check to see if the financial aid amount given is not more than tuition due | The tuition due must be greater because the aid cannot cover more than the given tuition due | Returns “Cannot make a payment greater than the tuition due” |
| 10 | Check to see if the resident student is part time or not | Part time students that are residents cannot receive financial aid | Returns “Parttime student doesn’t qualify for the reward” |
| 11 | Check to see if the financial aid amount is greater than 10,000 | Cannot have a financial aid amount greater than 10,000 | Returns “Financial aid cannot exceed 10,000 dollars” |
| 12 | Check to see if the financial aid button will work if the financial aid amount text field is left blank | Must return a statement to indicate that the financial aid amount text field is not empty. | Returns “Payment amount not found” |
| **Method signature:** void updateFinancialAid (ActionEvent event) | | | |
| **Test Case #** | **Requirement** | **Test Description and Input Data** | **Expected result/output** |
| 1 | Check to see if the update financial aid button works when the given major is not a valid major | Testing to see if the button detects that the major is not proper | Returns “Student not found” |
| 2 | Check to see if the update financial aid button works when the given student is not a valid student | Testing to see if the button detects that the student is not proper | Returns “Student not found” |
| 3 | Check to see if the update financial aid button works when everything is chosen properly with a proper student | Testing to see if the button works with the proper inputs | Returns “Tuition updated” |
| 4 | Check to see if the update financial aid button works the payment date is not selected | The button should still work because the financial aid does not require a payment date | Returns “Tuition updated” |
| 5 | Check to see if the update financial aid button works if the tuition for the given student has not been calculated yet | Testing if the button will work if the tuition has not been calculated yet | Returns ‘Student’s tuition must be calculated first” |
| 6 | Check to see if the financial aid button works if the payment text box is empty | Testing if the button will work if the payment text box is empty | Returns “Please only use numbers in the payment amount text field” |
| 7 | Check to see if the update financial aid button works if the payment text box has a negative number | Testing if the button will work if the payment text box has a negative number | Returns “Please enter a non-negative payment amount” |
| 8 | Checks to see if the student has the financial aid first before updating the financial aid | Updating financial aid cannot be done if the financial aid is not there in the first place | Returns “Must have financial aid done first in order to update financial aid” |
| 9 | Check to see if the update financial aid amount given is not more than tuition due | The tuition due must be greater because the aid cannot cover more than the given tuition due | Returns “Cannot make a payment greater than the tuition due” |
| 10 | Check to see if the financial aid amount is greater than 10,000 | Cannot have a financial aid amount greater than 10,000 | Returns “Financial aid cannot exceed 10,000 dollars” |
| 11 | Check to see if the financial aid button will work if the financial aid amount text field is left blank | Must return a statement to indicate that the financial aid amount text field is not empty. | Returns “Payment amount not found” |