MediForm Deliverable 5

Leader: Nic

Minutes: Paul

Tim

4/23/2021

|  |  |  |  |
| --- | --- | --- | --- |
| Notes |  | | |
| Input | Expect Output | Actual Output | Note |
| “This is the first note” | “This is the first note” added to the note String. The updated note String should then be displayed in the TextArea above. | “This is the first note” was added to note String. The updated note String is then displayed in the TextArea above. | This is the base case, where the system works as intended. |
| “” (Adding an empty note) | This would create a one line break between the previous and next note. | A single line break was created between the last and next note. | This might be better implemented to not add anything and inform the user that the field was empty. |
| “What happens when I use \r\n in a note” | The string would read the “\r\n” and place the “in a note” on a new line. | The “\r\n” was just added to the string and the whole note was added to a single line. | This is not actually an error but simply how the string builder operates. However this is still an unexpected result. |

|  |  |  |  |
| --- | --- | --- | --- |
| OrderMeds |  | | |
| Input | Expect Output | Actual Output | Note |
| 5 | 5 should be added to the respective medicine variable. | 5 is added to the respective medicine variable. | This is the base case, where the system works as intended. |
| -6 | -6 should be added to the respective medicine variable. | -6 is added to the respective medicine variable. | This is not an error as there are not any checks for negative numbers. This could be used as a method for reducing the number of medicines if necessary, or could be removed for clarity. |
| “test” | Produce an error on the command line. | This produces a Number Format Exception in the terminal, and the window does not change. | A check for non-int values could be implemented for more visual clarity for the user. |
| 16546516846 | Produce an error on the command line. | This produces a Number Format Exception in the terminal, and the window does not change. | In normal usage a number this large would likely never be entered, but the system is not able to handle a value this large as it is outside the range of the int data type. |

Patient ID Test:

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expect Output | Actual Output | Entered ID |
| Correct Patient ID | Brings me to the next screen. | Brings me to the next Screen | “6074fba91477040008b0c736” |
| Incorrect Patient ID | Display “Error patient ID doesn’t exist” | Displays “Error patient ID doesn’t exist” | “6074fba91477040008b0c73” |
| No Patient ID | Displays “Error patient ID field is Empty” | Displays “Error patient ID field is Empty” | I did not enter and ID |

Billing Test:

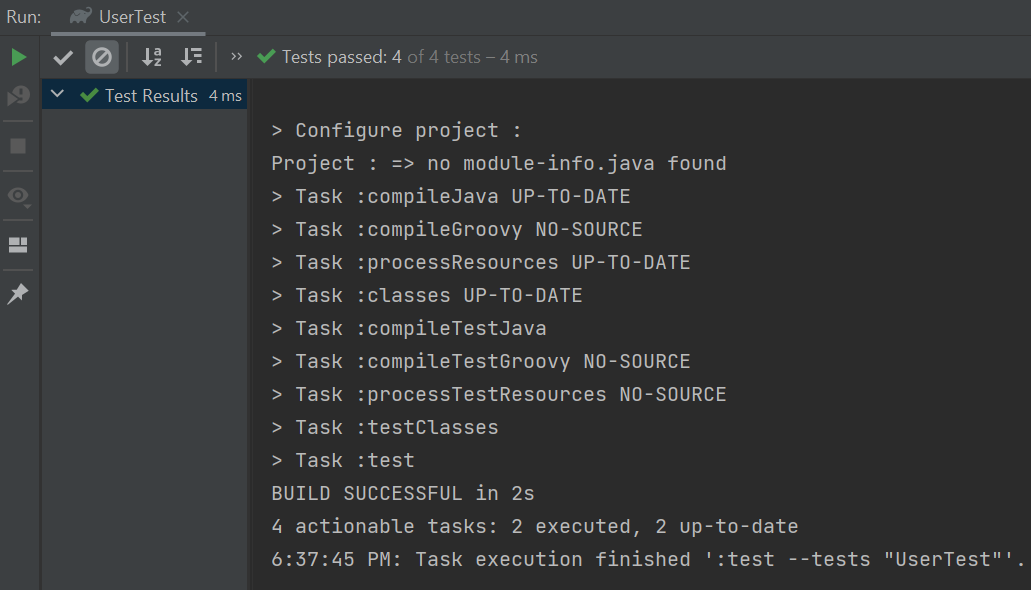
|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expect Output | Actual Output | Type numbers added |
| Billing That works | it compiles the bill correctly | it compiles the bill correctly | Positive |
| Billing that Has Negative | Error “Bill is Negative | Continues to Billing screen with -Total cost. | Negative |
| Billing With High Number | Error “Bill exceed max value” | Makes the billing with a negative number. | Large Positive Numbers |

|  |  |  |  |
| --- | --- | --- | --- |
| Add New Patient (input) | Expected Output | Actual Output | Notes |
| First Name: “John”  Last Name: “Smith”  Date of Birth: “05/12/1976”  Address: “23 Long St.”  City: “Toms River”  State: “New Jersey”  Phone Number:  “732-897-5643”  Insurance Provider: “Cigna”  Primary Physician: “Nancy Wells”  Emergency Contact Name: “Randy”  Emergency Contact Phone Number: “732-656-7865”  Reason for Visit: “chest pain”  Date of Visit: “05/17/1976”  Insurance ID: “F45838729321”  Current Medication(s): “finasteride”  Medical History: “leg surgery” | User should be brought back to the main menu screen while the new patient information is saved into the database. | The actual output matches the expected output exactly. | This is the base case for registering a new patient and it works how it should. |
| First Name: “John”  Last Name: “ ” (empty text field/ empty content)  Date of Birth: “05/12/1976”  Address: “23 Long St.”  City: “Toms River”  State: “New Jersey”  Phone Number:  “732-897-5643”  Insurance Provider: “Cigna”  Primary Physician: “Nancy Wells”  Emergency Contact Name: “Randy”  Emergency Contact Phone Number: “732-656-7865”  Reason for Visit: “chest pain”  Date of Visit: “05/17/1976”  Insurance ID: “F45838729321”  Current Medication(s): “finasteride”  Medical History: “leg surgery” | The user will be alerted when the “submit” button is clicked and user will not be able to continue, nor will the information be saved to the database until the user enters an input. | The user is not alerted and is still brought back to the main menu while the information is still saved to the database regardless of the empty field. | An exception could be added to make sure the user is alerted when a field is empty. |
| First Name: “John”  Last Name: “Smith”  Date of Birth: “05/12/2024”  Address: “23 Long St.”  City: “Toms River”  State: “New Jersey”  Phone Number:  “732-897-5643”  Insurance Provider: “Cigna”  Primary Physician: “Nancy Wells”  Emergency Contact Name: “Randy”  Emergency Contact Phone Number: “732-656-7865”  Reason for Visit: “chest pain”  Date of Visit: “05/17/1976”  Insurance ID: “F45838729321”  Current Medication(s): “finasteride”  Medical History: “leg surgery” | User will be alerted when the “submit” button is clicked saying the “date of birth” entered is invalid. The user will not be able to continue to the main menu nor will the information save to the database until the user enters a valid “date of birth.” | The user is not alerted and is still brought back to the main menu while the information is still saved to the database regardless of the invalid birth date. | An exception can be added to make sure that inputted values are valid and realistic. |

|  |  |  |  |
| --- | --- | --- | --- |
| Nurse MedForm (entering vitals) | Expected Output | Actual Output | Notes |
| Height: “167.4”  Weight: “146.8”  Temperature: “98.6”  Blood Pressure: “98/64”  Pulse Rate: 78bpm | When clicking submit, the vitals will be saved to the database and the nurse user will be returned to the nurse main menu. Prior to inputting the vitals on the screen, the patient’s first name, last name, and assigned physician will be displayed in the correct text fields. | The actual output matches the expected output exactly. | This is the base case for entering vitals and it works how it should. |
| Height: “-167.4”(negative value)  Weight: “146.8”  Temperature: “98.6”  Blood Pressure: “98/64”  Pulse Rate: 78bpm | The nurse user will be alerted when clicking the submit button and will not be able to continue until the nurse enters a realistic height value. | The nurse user is not alerted and the vitals are still saved to the database. The nurse is also still brought to the main menu. | An exception can be added to make sure realistic and valid values are entered before the user can continue. |
| Height: “ ”(empty value)  Weight: “146.8”  Temperature: “98.6”  Blood Pressure: “98/64”  Pulse Rate: 78bpm | The nurse user will be alerted when clicking the submit button and will not be able to continue until the nurse enters a height value for the patient. | The nurse user is not alerted and the vitals are still saved to the database as is. The nurse is also still brought back to the main menu. | An exception can be added to make sure the user fills all required vital fields before continuing. |

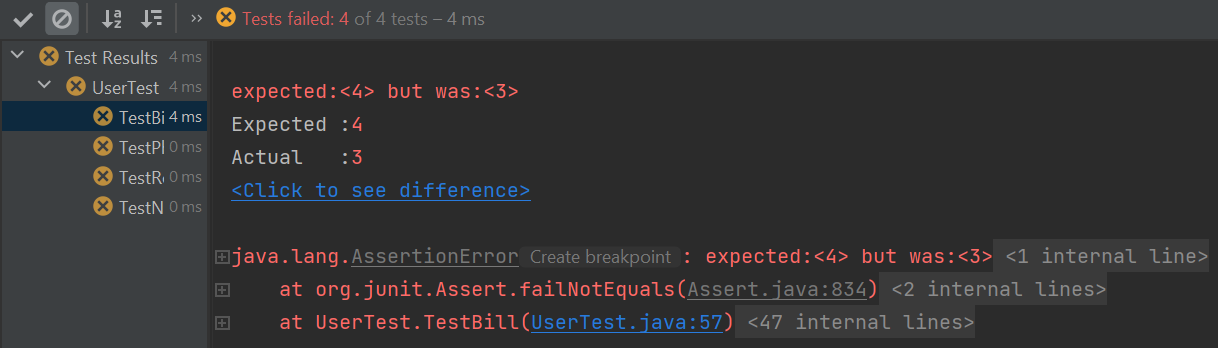
JUnit Testing:

All Test On User Class:



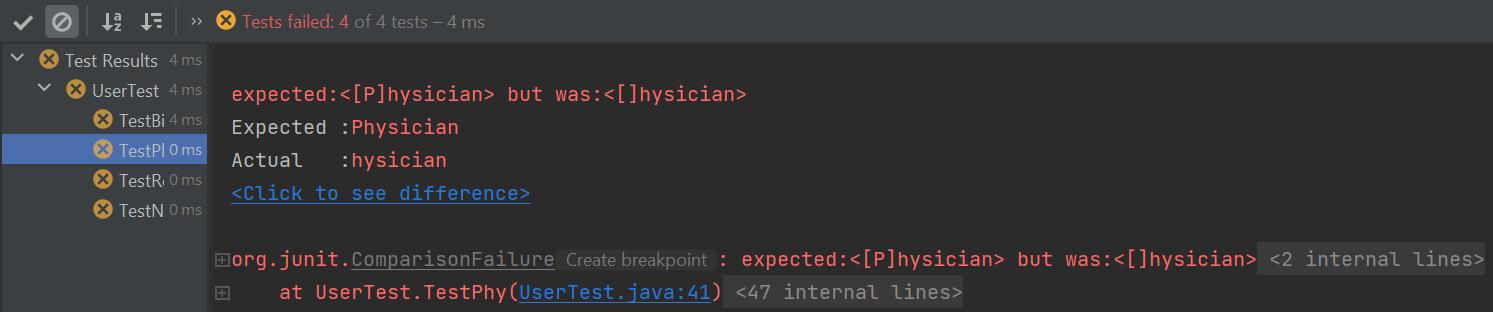
All tests PASS all spelling is correct for each testing phase.

Test For Billing Fails:



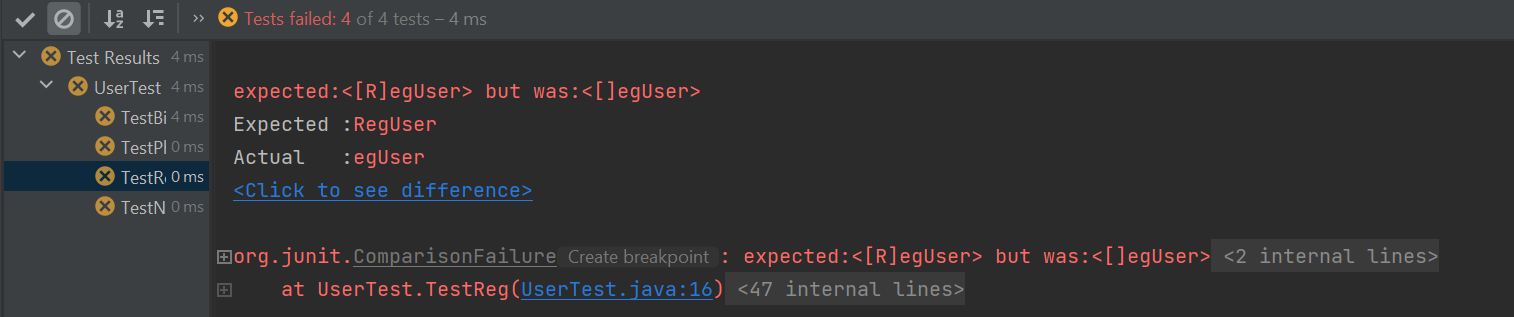
This test failed because the roleID is wrong.

Test For Physician Fails:



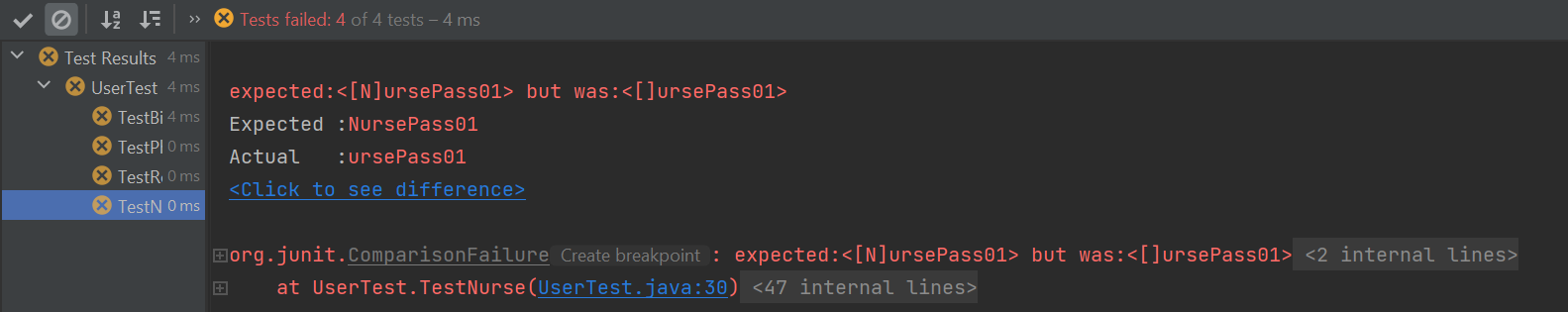
This test failed because the Name was Wrong.

Test For Registration Fails:



This test failed because the Username was spelled incorrectly.

Test For Nurse Fails:



This test failed because the password was incorrect.