|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Time / compareTo | | | | |
| Test Case # | Requirement | Test Description and Input Data | Expected Result / Output | Actual Result / Output |
| 1 | The method recognizes the same time given. | Create a time instance and compare it to itself.  Test data: “9:00” compareTo “9:00” | 0 | 0 |
| 2 | The method recognizes a future time by at least 1 hour. | Create two time instances, one of which is 1 hour later than the other.  Test data: “10:00” compareTo “9:00” | 1 | 1 |
| 3 | The method recognizes a future time by at least 1 minute. | Create two time instances, one of which is 1 minute later than the other.  Test data: “9:01” compareTo “9:00” | 1 | 1 |
| 4 | The method recognizes a past time. | Create two time instances, one of which earlier than the other by 1 minute.  Test data: “8:59” compareTo “9:00” | -1 | -1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Timeslot / compareTo | | | | |
| Test Case # | Requirement | Test Description and Input Data | Expected Result / Output | Actual Result / Output |
| 1 | The method recognizes two same timeslots, which have the same date and time. | Create a timeslot instance and compare it to itself.  Test data: “10/09/2000, 10:00” compareTo “10/09/2000, 10:00” | 0 | 0 |
| 2 | The method recognizes when a timeslot is 1 day later than the other. | Create two timeslot instances and compare the earlier one to the later one.  Test data: “10/09/2000, 10:00” compareTo “10/10/2000, 10:00” | -1 | -1 |
| 3 | The method recognizes when a timeslot is 1 day earlier than the other. | Create two timeslot instances and compare the later one to the earlier one.  Test data: “10/09/2000, 10:00” compareTo “10/08/2000, 10:00” | 1 | 1 |
| 4 | The method recognizes when a timeslot is 15 mins later than the other. | Create two timeslot instances and compare the earlier one to the later one.  Test data: “10/09/2000, 10:00” compareTo “10/09/2000, 10:15” | -1 | -1 |
| 5 | The method recognizes when a timeslot is 15 mins earlier than the other. | Create two timeslot instances and compare the earlier one to the later one.  Test data: “10/09/2000, 10:00” compareTo “10/09/2000, 09:45” | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Patient / compareTo | | | | |
| Test Case # | Requirement | Test Description and Input Data | Expected Result / Output | Actual Result / Output |
| 1 | The method recognizes the same person | Create an instance of a patient and compare it to itself.  Test data: “Jah, Caffie-Speed, 5/6/2001” compareTo “Jah, Caffie-Speed, 5/6/2001” | 0 | 0 |
| 2 | The method recognizes two different people. | Create two unique instances of a patient and compare them.  Test data: “Lia, Shekaran, 5/17/2000” compareTo “Jah, Caffie-Speed, 5/6/2001” | -1 | -1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date / isValid | | | | |
| Test Case # | Requirement | Test Description and Input Data | Expected Result / Output | Actual Result / Output |
| 1 | The method does not accept dates with numbers out of bounds. | Create an instance of a date with invalid months, days, and years.  Test data: “0/00/1000” | False | False |
| 2 | The method does not accept dates with the day > 30 | Create an instance of a date with an invalid day, but all else valid.  Test data: “1/5000/2020” | False | False |
| 3 | The method does not accept a date with the month > 12 | Create an instance of a date with an invalid month, but all else valid.  Test data: “13/02/2022” | False | False |
| 4 | The method does not accept dates with year > 9999 | Create an instance of a date where the year is invalid, but all else is valid.  Test data: “1/01/10000” | False | False |
| 5 | The method does not accept a February date with day > 28 (non-leap) | Create an instance of a date where the days are out of bounds in February.  Test data: “2/31/2023” | False | False |
| 6 | The method accepts the last February day on a leap year. | Create an instance of a date where the date is the last day of a leap year February.  Test data: “2/29/2020” | True | True |
| 7 | The method accepts a date within the bounds. | Create an instance of a date where the date is within bounds and is valid.  Test data: “2/10/2015” | True | True |