

- 1) Source code is found in the 'Expedia' project; test code is found in the 'ExpediaTest' project.
- 2) Expedia
 - a. Booking
 - b. Car
 - c. Flight
 - d. Hotel
 - e. User
- 3) A Flight can be created; it requires a start date, end date and a number of miles, all of which are saved into their respective properties of the class. When creating a flight, the start date must be before the end date and the number of miles must be greater than 0. Flight has the ability to check equality of two flights.
- 4) BookingTest, CarTest, FlightTest, HotelTest, UserTest
- 5) TestTahtUserInitializes, TestTahtUserHasZeroFrequentFlierMilesOnInit, TestThatUserCanBookEverything, TestThatUserHasFrequentFlierMilesAfterBooking, TestThatUserCanBookAFlight, TestThatUserCanBookAHotelAndCar, TestThatUserHasCorrectNumberOfFrequentFlyerMilesAfterOneFlight
- 6) Catch, False, Greater
- 7) Catch (verifies an exception was thrown), False (verifies 'false' is returned), Greater (verifies the first argument is greater than the second argument)
- 8) AreEqual verifies that the two objects are equal while AreSame verifies the two objects refer to the same object.
- 9) TestThatHotelInitializes verifies that a hotel object was created by checking if it is null after its creation.
- 10) \$45 times the number of nights user will stay in the hotel
- 11) The three tests verify that the dollar amount for a 1, 2, and 10 day stay in a hotel is correct.
- 12) We do not include a null check in all tests because that is repeating code; once a test is written to test a functionality that test does not need to be repeated throughout the code.
- 13) The Hotel class throws an exception when the length of stay is less than or equal to 0.
- 14) [ExpectedException(typeof(OutOfMemoryException))]