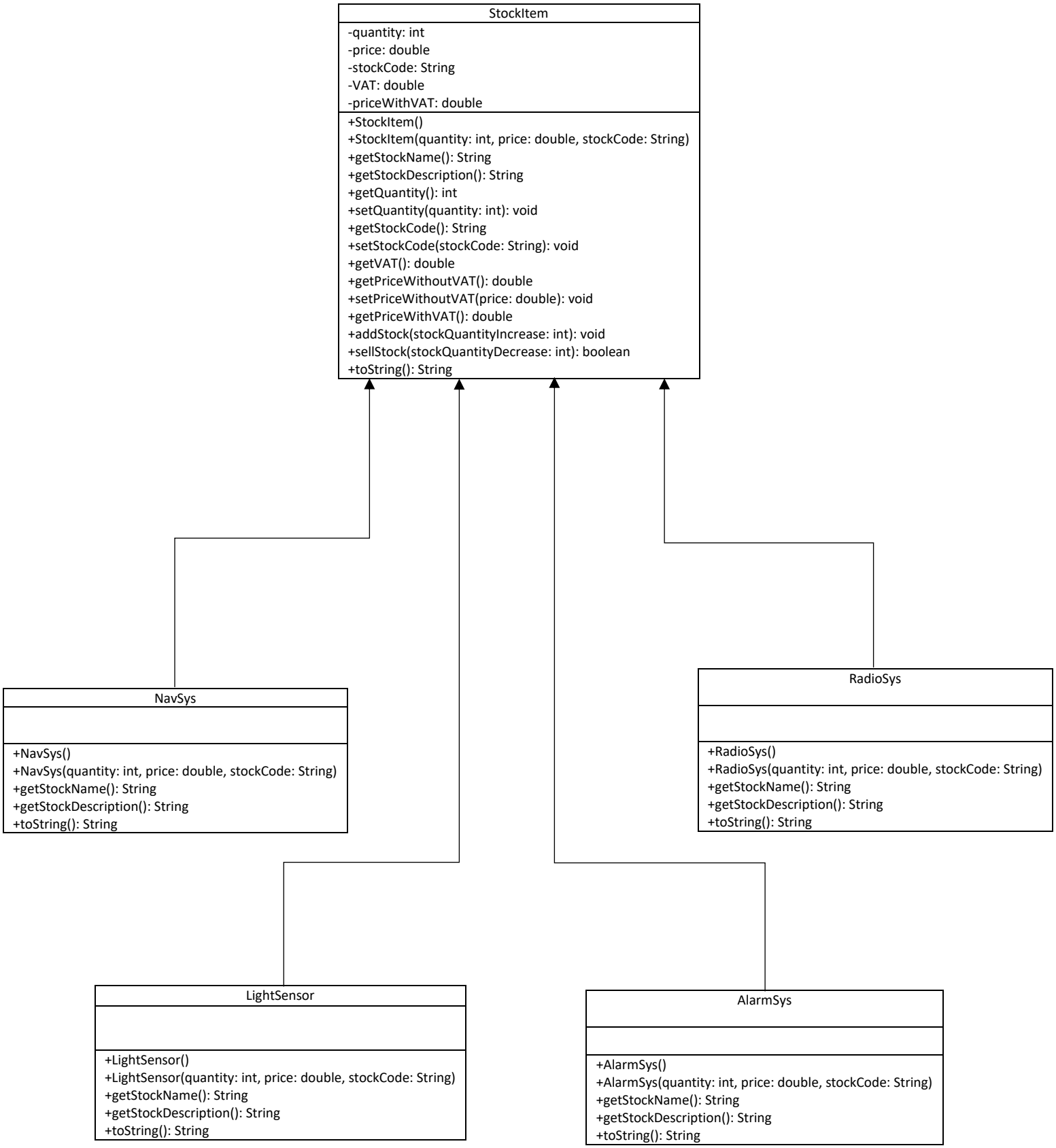
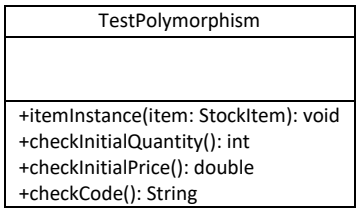
UML diagram for car parts shop





Test cases

*Note: "**NavSys**" class will be used as example in the test case scenarios below*

Test case	Purpose	Expected result
quantity = -1	Test an initial stock quantity which is negative	Error message
quantity = 0	Test an initial stock quantity which is 0	Error message
quantity = 101	Test an initial stock quantity which exceeds the maximum stock quantity	Error message
quantity = "k"	Test an initial stock quantity which is a string	Error message
quantity = 88.88	Test an initial stock quantity which is a float/double	Error message
quantity = 10	Test an initial stock quantity which is within the valid range [1, 100]	No error message/Valid input
price = -1	Test an initial item price which is negative	Error message
price = 0	Test an initial item price which is 0	Error message
price = "k"	Test an initial price which is a string	Error message
price = 99.99	Test an initial item price which is positive	No error message/Valid input
stockCode = "NS101"	Test an initial stock code	No error message/Valid input
After a valid initialisation NavSys Class(current stock info) quantity = 10 price = 99.99 stockCode = "NS101"		
increaseQuantity = -1	Test increasing the stock by a negative value	Error message/no stock increase quantity = 10
increaseQuantity = 0	Test increasing the stock by 0	Error message/no stock increase quantity = 10

increaseQuantity = 91	Test increasing the stock by a value which makes the stock exceed its maximum quantity	Error message/no stock increase quantity = 10
increaseQuantity = 101	Test increasing the stock by a value which makes the stock exceed its maximum quantity	Error message/no stock increase quantity = 10
IncreaseQuantity = “k”	Test increasing the stock by a string value	Error message/no stock increase quantity = 10
IncreaseQuantity = 88.88	Test increasing the stock by a float/double value	Error message/no stock increase quantity = 10
increaseQuantity = 20	Test increasing the stock by a valid value	No error message/Valid stock increase quantity = 30
After a valid stock increase NavSys Class(current stock info) quantity = 30 price = 99.99 stockCode = “NS101”		
decreaseQuantity = -1	Test decreasing the stock by a negative value	Error message/no stock reduction quantity = 30
decreaseQuantity = 0	Test decreasing the stock by 0	Error message/no stock reduction quantity = 30
decreaseQuantity = 31	Test decreasing the stock by a value greater than the current stock quantity	Error message/no stock reduction quantity = 30
decreaseQuantity = “k”	Test decreasing the stock by a string value	Error message/no stock reduction quantity = 30
decreaseQuantity = 88.88	Test decreasing the stock by a float/double value	Error message/no stock reduction quantity = 30
decreaseQuantity = 15	Test decreasing the stock by a valid value	No error message/Valid stock reduction quantity = 15

After a valid stock reduction NavSys Class(current stock info) quantity = 15 price = 99.99 stockCode = ”NS101”		
price = -100	Test setting a new price which is negative	Error message/no price update price = 99.99
price = 0	Test setting a new price which is 0	Error message/no price update price = 99.99
price = “k”	Test setting a new price which is a string value	Error message/no price update price = 99.99
price = 120	Test setting a new valid price	No error message/Valid price update price = 120
After setting new price NavSys Class(current stock info) quantity = 15 price = 120 stockCode = ”NS101”		