**Columns of Grids**

**Raw Supplies**

frmSupplier

|  |  |  |
| --- | --- | --- |
| List of Suppliers | **Column Name** | **Database or Equivalent** |
| Supplier Name | supplier - supplierName |
| Supplier Description | supplier - supplierDetails |
| Contact Details | supplier - contactDetails |
| Address | supplier - address |

|  |  |  |
| --- | --- | --- |
| Items Sold by Supplier | **Column Name** | **Database or Equivalent** |
| Category Name | Supply\_category - categoryName |
| Supply Name | Supply\_items - supplyName |

|  |  |  |
| --- | --- | --- |
| Sold by Selected Supplier   * Categories | **Column Name** | **Database or Equivalent** |
| Category Name | Supply\_category - categoryName |

|  |  |  |
| --- | --- | --- |
| Sold by Selected Supplier   * Items | **Column Name** | **Database or Equivalent** |
| Supply Name | Supply\_items - supplyName |
| Supply Description | Supply\_items - supplyDescription |

frmAddSupplierItems

|  |  |  |
| --- | --- | --- |
| Items Sold by Supplier | **Column Name** | **Database or Equivalent** |
| Category Name | Supply\_category - categoryName |
| Supply Name | Supply\_items - supplyName |

frmAddSupplyCategory

|  |  |  |
| --- | --- | --- |
| Supply Categories | **Column Name** | **Database or Equivalent** |
| Category Name | Supply\_category - categoryName |
| Category Description | Supply\_category - categoryName |
| Type of Measure | Supply\_category - categoryName |

frmSupplyItems

|  |  |  |
| --- | --- | --- |
| Supply Items | **Column Name** | **Database or Equivalent** |
| Supply Name | Supply\_items - supplyName |
| Category Name | Supply\_category - categoryName |
| Supply Description | Supply\_items - supplyDescription |
| Measurement | Supply\_items – if Area{concat(measureA,measureB)}  Else  {measureA} |
| Unit Measure | Supply\_items - unitMeasure |
| Purchase Price per Unit | Supply\_items – unitPurchasePrice |
| Quantity Left | DOUBLE  Length {  Quantity Left = [1-(Total Lengths used Stock Out/[Total StockIn Amount in Length-Total Damage Items in Length StockOut])]\*(Total StockIn Quantity)  }  Area{  Get Area first (LxW)  Area2=measureA\*measureB  Quantity Left = [1-(Total Areas2 Stock Out/Total StockIn Amount in Area2- [Total Damage Items in Area2 StockOut])]\* (Total StockIn Quantity)  }  Weight{  Quantity Left = [1-(Total Weights Stock Out/[Total StockIn Amount in Weights-Total Damage Items in Weight StockOut])]\* (Total StockIn Quantity)  }  Volume{  Quantity Left = [1-(Total Volume Stock Out/[Total StockIn Amount in Volume -Total Damage Items in Volume StockOut])]\* (Total StockIn Quantity)  }  Whole{  Quantity Left = Total StockIn Quantity-(Total Stockout Quantity)  }  Note:  TOTAL STOCK OUT INCLUDES frame\_materials, jOrderDetails, damaged\_materials  MeasureA   * LengthMeasurement * WeightMeasurement * TotalStockOutQuantity(1 ang value nila and cannot use decimal) * Whole pieces   MeasureA\*MeasureB   * Area2 Measurement |
| Quantity Left in Measurement | Length {  Quantity Left = (LengthMeasurement\* Total StockIn Quantity) – (Total Lengths Used in StockOut)  }  Area{  Get Area first (LxW)  Area2=measureA\*measureB  Quantity Left = (Area2Measurement\* Total StockIn Quantity) – (Total Area2 Used in StockOut)  }  Weight{  Quantity Left = (Weight Measurement\* Total StockIn Quantity) – (Total Weight Used in StockOut)  }  Volume{  Quantity Left = (Volume Measurement\* Total StockIn Quantity) – (Total Volume Used in StockOut)  }  Whole{  Total StockIn Quantity-(Total Stockout Quantity)  }  Note:  TOTAL STOCK OUT INCLUDES frame\_materials, jOrderDetails, damaged\_materials  MeasureA   * LengthMeasurement * WeightMeasurement * TotalStockOutQuantity(1 ang value nila and cannot use decimal)   MeasureA\*MeasureB   * Area2 Measurement |

frmSupplyStockIn

|  |  |  |
| --- | --- | --- |
| Select Supplier | **Column Name** | **Database or Equivalent** |
| Supplier Name | supplier - supplierName |
| Supplier Description | supplier - supplierDetails |
| Contact Details | supplier – Contact Details |

|  |  |  |
| --- | --- | --- |
| Stock In Records | **Column Name** | **Database or Equivalent** |
| Supply Name | Supply\_details - supplyName |
| Supplied by | Supplier - supplierName |
| Date Delivered | Supply\_details – delivery\_date |
| Calculated Total Purchase Price | Supply\_details – priceRawTotal |
| Actual Total Purchase Price | Supply\_details - pricePurchaseTotal |

frmStockInSupplyEdit

|  |  |  |
| --- | --- | --- |
| Select Supplier | **Column Name** | **Database or Equivalent** |
| Supplier Name | supplier - supplierName |
| Supplier Description | supplier - supplierDetails |
| Contact Details | supplier – Contact Details |

frmSupplyDamage

|  |  |  |
| --- | --- | --- |
| Stock Out Records | **Column Name** | **Database or Equivalent** |
| Supply Name | supplier - supplierName |
| Supply Category | supplier - supplierDetails |
| Measurement | Damaged\_materials – if Area{concat(measureA,’x’,measureB)}  Else  {measureA} |
| Unit Measure | Damaged\_materials – unit measure |
| Quantity Stocked Out | Damaged\_materials – totalQuantityStockedOut |
| Date Stocked Out | Damaged\_materials – date\_created |
| Date Modified | Damaged\_materials – date\_modified |
| Cost of Stock Out | Purchase Unit Cost from frame\_list\*Stocked Out Quantity |

TO CALCULATE for TXTBOX OF (CALCULATE STOCK OUT QUANTITY)

DOUBLE

Length {

Quantity = Total Quantity Left-([1-(Total Lengths damaged StockOut/Total StockIn Amount in Length)]\* Total Quantity Left)

}

Area{

Get Area first (LxW)

Area2=measureA\*measureB

Quantity = Total Quantity Left-([1-(Total Areas2 damaged Stock Out/Total StockIn Amount in Area2)]\* Total Quantity Left)

}

Weight{

Quantity Left = Total Quantity Left - ([1-(Total Weights damaged Stock Out/Total StockIn Amount in Length)]\* Total Quantity Left

}

Whole{

Total damaged Stockout Quantity

}

Note:

MeasureA

* LengthMeasurement
* WeightMeasurement
* TotalStockOutQuantity(1 ang value nila and cannot use decimal)
* Whole pieces

MeasureA\*MeasureB

* Area2 Measurement

**Frame Inventory**

frmFrameList

|  |  |  |
| --- | --- | --- |
| List of Frames | **Column Name** | **Database or Equivalent** |
| Frame Name | Frame\_list - frameName |
| Dimension | Frame\_list - dimension |
| Description | Frame\_list – frameDescription |
| Quantity Left | Quantity Left = Total of frameStock\_In Quantity – Total of sOrder\_Details Quantity |
| Sales Price per Unit | Frame\_list – unitSalesPrice |

|  |  |  |
| --- | --- | --- |
| Select Supplies Used by Selected Frame | **Column Name** | **Database or Equivalent** |
| Supply Name | Supply\_items - supplyName |
| Supply Category | Supply\_category - categoryName |
| Measurements | Frame\_materials – if(Area, concat(measureADeduction,’x’,  measureBDeduction))  else measureA |
| Unit Measure | Frame\_materials - unitMeasure |
| Price per Unit Measure | *See FrmFrameCreate* |
| Raw Cost | *See FrmFrameCreate* |
| Total Raw Cost based on Quantity Left | Raw Cost\*Quantity |

frmFrameCreate

|  |  |  |
| --- | --- | --- |
| Select Supplies Used by Selected Frame | **Column Name** | **Database or Equivalent** |
| Supply Name | Supply\_items - supplyName |
| Supply Category | Supply\_category - categoryName |
| Measurements | Frame\_materials – if(Area, concat(measureADeduction,’x’,  measureBDeduction))  else measureA |
| Unit Measure | Frame\_materials - unitMeasure |
| Price per Unit Measure | DOUBLE  ***Based on selected Unit Measure from frmFrameAddSupplies = convert through if statements, if measurement is not the same as supply\_items unitMeasure. Must be lowest whole number unit measure which is 1.***  Length {  Price/Based on selected Unit Measure = unitPurchasePrice/  Based on selected Unit Measure from frmFrameAddSupplies Usage  }  Area{  Get Area first (LxW)  Area2=measureA\*measureB  Price/ Based on selected Unit Measure = unitPurchasePrice/ Based on selected Unit Measure from frmFrameAddSupplies Usage Area2}  Weight{  Price/ Based on selected Unit Measure = unitPurchasePrice/  Based on selected Unit Measure from frmFrameAddSupplies Usage  }  Whole{  Price/ Based on selected Unit Measure = supply\_items.unitPurchasePrice}  Note:  MeasureA   * LengthMeasurement * WeightMeasurement * TotalStockOutQuantity(1 ang value nila and cannot use decimal) * Whole pieces   MeasureA\*MeasureB  Area2 Measurement |
| Raw Cost | Length {  Raw Cost = Price per Unit Measure\*Length Usage  }  Area{  Get Area first (LxW)  Area2=measureA\*measureB  Raw Cost = Price per Unit Measure\*Area2 Usage  }  Weight{  Raw Cost = Price per Unit Measure\*Weight Usage  }  Whole{  Raw Cost = Price per Unit Measure\*Quantity Usage  }  Note:  MeasureA   * LengthMeasurement * WeightMeasurement * TotalStockOutQuantity(1 ang value nila and cannot use decimal)   MeasureA\*MeasureB   * Area2 Measurement |

frmFrameEdit

|  |  |  |
| --- | --- | --- |
| Select Supplies Used by Selected Frame | **Column Name** | **Database or Equivalent** |
| Supply Name | Supply\_items - supplyName |
| Supply Category | Supply\_category - categoryName |
| Measurements | Frame\_materials – if(Area, concat(measureADeduction,’x’,  measureBDeduction))  else measureA |
| Unit Measure | Frame\_materials - unitMeasure |
| Price per Unit Measure | See FrmFrameCreate |
| Raw Cost | See FrmFrameCreate |

frmFrameAddSuppliesUsed

|  |  |  |
| --- | --- | --- |
| List of Supplies | **Column Name** | **Database or Equivalent** |
| Supply Name | supplier - supplierName |
| Supply Category | supplier - supplierDetails |
| Measurement | Supply\_items – if Area{concat(measureA,measureB)}  Else  {measureA} |

frmFrameEditSuppliesUsed

|  |  |  |
| --- | --- | --- |
| List of Supplies | **Column Name** | **Database or Equivalent** |
| Supply Name | supplier - supplierName |
| Supply Category | supplier - supplierDetails |
| Measurement | Supply\_items – if Area{concat(measureA,measureB)}  Else  {measureA} |

frmFrameStockInAdd

|  |  |  |
| --- | --- | --- |
| Stock In Record of Frame | **Column Name** | **Database or Equivalent** |
| Date Stocked In | FrameStock\_In – dateStockIn |
| Stocked In Quantity | FrameStock\_In - stockInQuantity |
| Date Modified | FrameStock\_In - dateModified |
| Stocked in By | Employee – concat(lastname, ‘, ’, firstname) |
| Cost of Stock In | Raw Cost from FrmFrameCreate\*Stocked In Quantity |

frmFrameStockInUpdate

Retrieve data from selected item in frmFrameStockInAdd

frmFrameStockInAdd

|  |  |  |
| --- | --- | --- |
| Stock In Record of Frame | **Column Name** | **Database or Equivalent** |
| Date Stocked In | FrameStock\_In – dateStockIn |
| Stocked In Quantity | FrameStock\_In - stockInQuantity |
| Date Modified | FrameStock\_In - dateModified |
| Stocked in By | Employee – concat(lastname, ‘, ’, firstname) |
| Cost of Stock In | Raw Cost from FrmFrameCreate\*Stocked In Quantity |

frmFrameDamageItems

|  |  |  |
| --- | --- | --- |
| Stock In Record of Frame | **Column Name** | **Database or Equivalent** |
| Date Stocked In | FrameStock\_In – dateStockIn |
| Stocked Out Quantity | FrameStock\_In - stockInQuantity |
| Date Modified | FrameStock\_In - dateModified |
| Cost of Stock Out | Sales Unit Cost from frame\_list\*Stocked Out Quantity |

frmFrameArchivedList

|  |  |  |
| --- | --- | --- |
| Archived Frame List | **Column Name** | **Database or Equivalent** |
| Frame Name | Frame\_list - frameName |
| Dimension | Frame\_list - dimension |
| Description | Frame\_list – frameDescription |