Date	25/10/2025
Team ID	NM2025TMID04337
Project Name	MedicalInventoryManagementSystem
Maximum Mark	10Marks

1. Introduction

ThisdocumentdescribesthesystemdesignfortheMedicalInventoryManagementSystem using Salesforce concepts and best practices.

Data Model

Maincustom objects:

• Medicine:fieldsincludeName,SKU,BatchNumber,ExpiryDate,QuantityOnHand, ReorderLevel,UnitPrice.
• Supplier:Name,Contact,Lead Time.
• PurchaseOrder:PONumber,Supplier,OrderDate,Status.
• StockTransaction:Type(In/Out),Quantity,RelatedMedicine,Date,Reference(POor Issue).

ObjectRelationships

• Supplier(Master)→PurchaseOrder(Detail)
• Medicine→StockTransaction(Lookup)
• PurchaseOrder→StockTransaction(Lookupforreceivedgoods)

PageLayoutand UI

Use simple Lightning Record Pages with clear related lists: for Medicine page show Stock TransactionsandrecentPurchaseOrders.CreateaLightningAppforPharmacistwithquick actions: 'Log Stock In', 'Log Stock Out', 'Create Purchase Order'.

AutomationDesign

Flows:

• StockUpdateFlow:whenaStockTransactioniscreated,update Medicine.Quantity_On_Hand.
• LowStockAlertFlow:scheduledflowtofindMedicinesbelowReorderLevelandcreate Tasks/Notifications.
• ExpiryAlertFlow:scheduledflowtofinditemsnearingexpiryandnotifyPharmacists.

Securityand Access

Profilesandpermissionsets:Pharmacist(editstock),Admin(fullaccess),Supplier(limited access via community if needed). Use field-level security for sensitive fields.

IntegrationandDataImport

UseDataLoaderforinitialimportofmedicinemasterdata.Forfuture,considerAPIsto integrate with procurement or ERP systems.

```
//TriggerName:StockTransactionTrigger

//Object:Stock_Transactionc

//Purpose:ToupdateMedicinestockquantitywheneveraStockTransactionis inserted.


triggerStockTransactionTriggeronStock_Transactionc(afterinsert,afterupdate,after delete) {


  //MaptostoreMedicineIDsandtotalquantitychange

  Map<Id,Decimal>medicineStockMap=newMap<Id,Decimal>();


  // Handle Insert and Update Events

  if(Trigger.isInsert||Trigger.isUpdate){

    for(Stock_Transactionc st : Trigger.new){

      if(st.Medicinec!=null&&st.Quantityc!=null){

        //Iftransactiontypeis"In"->addstock,"Out"->reduce stock

        DecimalqtyChange=(st.Transaction_Typec=='In')?st.Quantityc:- st.Quantityc;

        medicineStockMap.put(st.Medicinec,

          medicineStockMap.containsKey(st.Medicinec)

          ?medicineStockMap.get(st.Medicinec)+qtyChange

          :qtyChange);

      }

    }
```

```
}

//HandleDeleteEvents if(Trigger.isDelete){
   for(Stock_Transactionc st : Trigger.old){
      if(st.Medicinec!=null&&st.Quantityc!=null){
         //Revertstockbasedondeleted record
         DecimalqtyChange=(st.Transaction_Typec=='In')?-st.Quantityc: st.Quantityc;
         medicineStockMap.put(st.Medicinec,
            medicineStockMap.containsKey(st.Medicinec)
            ?medicineStockMap.get(st.Medicinec)+qtyChange
            :qtyChange);
      }
   }
}

// Update Medicine records with new stock quantities
List<Medicinec>medicinesToUpdate=newList<Medicinec>(); for(Id
medId : medicineStockMap.keySet()){
   Medicinecmed=newMedicinec(Id=medId);
   med.Available_Stockc=(med.Available_Stockc==null?0:med.Available_Stockc)
+ medicineStockMap.get(medId);
   medicinesToUpdate.add(med);
}

if(!medicinesToUpdate.isEmpty()){
```

```
        update medicinesToUpdate;
    }
}
```

Diagrams(TextDescription)

ERDiagram:Supplier—PurchaseOrder—StockTransaction—Medicinelinks.ProcessFlow: ReceiveGoods→LogStockIn→Flowupdatesstock→Dashboard refreshes.

Outcome

Designaimsforclarity,simplicity,andmaintainability.Thechosenmodelsupports scalability for multiple sites.