SysML Diagrams: Tips, Types and More.

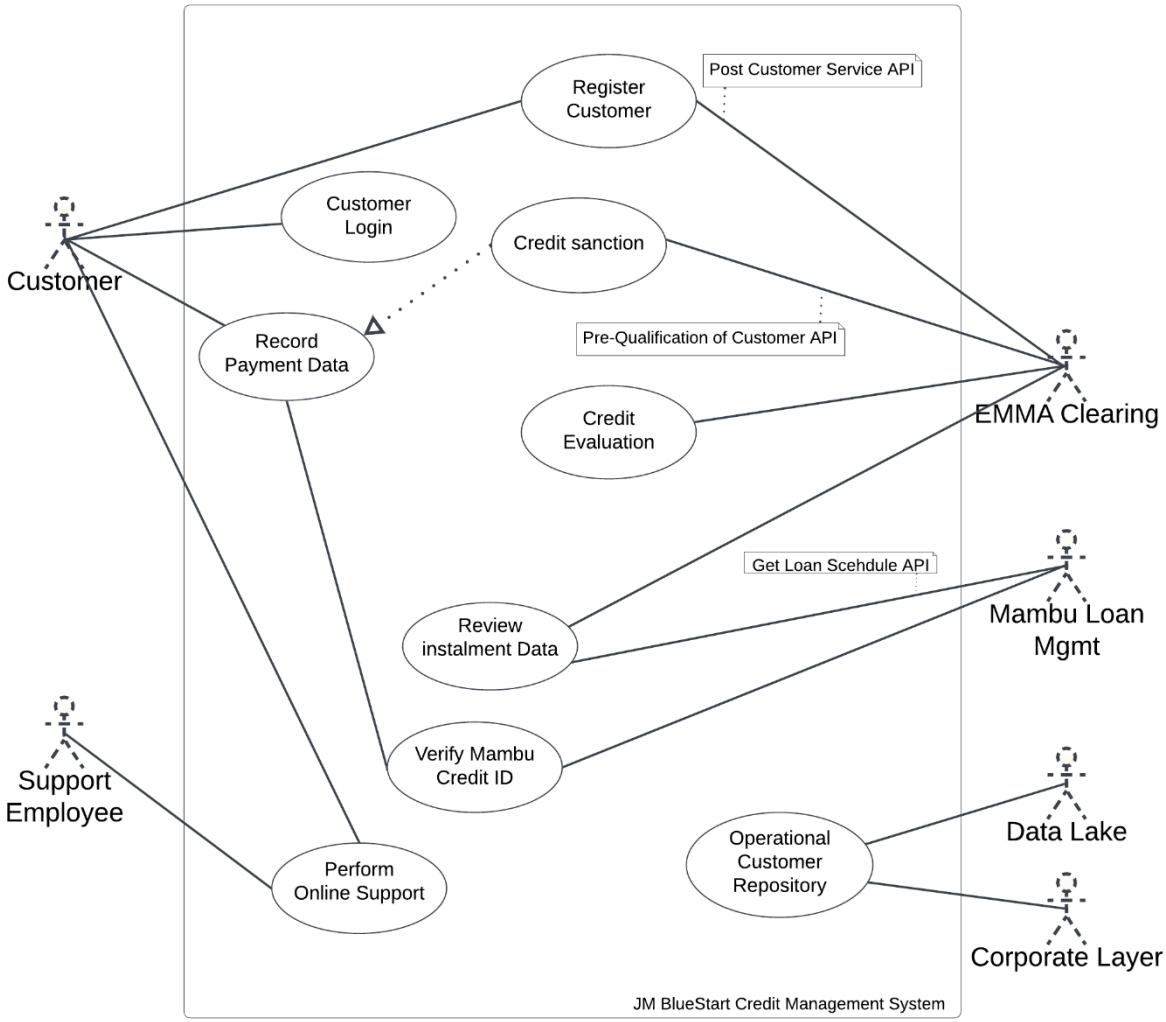
Systems Modeling Language (SysML) [OMG 2008a] aims to support the specification, analysis, design, and verification and complex system consisting of hardware, software, information, personnel, procedures, and organizations. Using the Modelling constructs of SysML, requirements for such systems as well as the behaviour and structure of these systems can be documented

SysML Diagram Types

Diagram Type	Description
Block definition diagram	A block is a modular unit of decomposition. It can represent for instance, a system, a subsystem, a hardware device, a software component, or a data element. The block definition diagram documents the structure of a system in terms of its building blocks, their structural and behavioural features, and their relationships (e.g. association, composition, and generalisation)
Internal block diagram	The internal block diagram documents the internal structure of a block in terms of its parts and connections among the parts.
Package Diagram	The package diagram is used to structure a system model by grouping model elements into packages and establishing relationships ammong the packages, or theelements within the packages
Parameteric diagram	The parametric diagram documents constraints on value properties of set of value properties. For instance, the physical law 'F=m*a' can be defined as a constraint that relates the value properties F (Force), m (mass), a (acceleration)
Activiity Diagram	The activity diagram documents activities, object/data flows and control flows among the activities. SysML extends the UML activity diagram. The extensions include, for instance, control operators which allow not only to enable but also to disable an activity, continuous obkect flows, and probabilities
Sequence Diagram	The sequence diagram documents interactions between actors and sustems or between parts of a ystem (i.e blocks)
State Machine diagram	The state machine diagram documents the state transitions and activities or actions that a system or a part of a system performs as reactions to events.
Use Case diagram	The use case diagram provides an overview of the scenarios documenting the usage of the system by its actors
Requirements diagram	The requirements diagram documents requirements and their relationships such as hierarchical stucture or refinement. In addition, it supports interrelation of requirements and other model elements which for instance, satisfy or verify a requirement

SysML defines the following relationships among requirements, or requirements between requirements and other model elements:

Requirements containment relationship: This relationship documents that a requirement called a sub requirement



Is part of another requirement (Called Compound requirement). A Sub requirement can be part of at most one compound requirement at a time. If a requirement is decomposed into a (set of) requirement (s), the containment relationship defines that, to satisfy compound requirement, all defined sub requirements must be satisfied.

Copy relationship: This relationship documents that a requirement (called a client requirement) is a copy of another requirement (Called a supplier requirement). The copy of a requirement is, for example, needed when a requirement is reused in another context (e.g. between different system versions or in product line development). The supplier requirement and the client requirement maintain a master-slave relationship which means that the name and the identifier of the client requirement are different from the original requirement, whereas the describing text is identical and cannot be edited. If the supplier requirement is decomposed into sub requirements, a copy of each sub requirement is made. Each sub requirement and its copy are associated by a copy relationship.

Derive relationship: This relationship documents that a requirement (the client) can be derived from another requirement (the supplier). For instance, a functional requirement may be derived from a goal, or a subsystem requirement may be derived from a system requirement.

Refine relationship: his relationship documents that a model element, for instance a use case, defines a requirement in more detail. The ‘refine’ relationship does not state whether the refinement is complete or partial Multiple model elements can be related to the same requirement by ‘refine’ relationships.

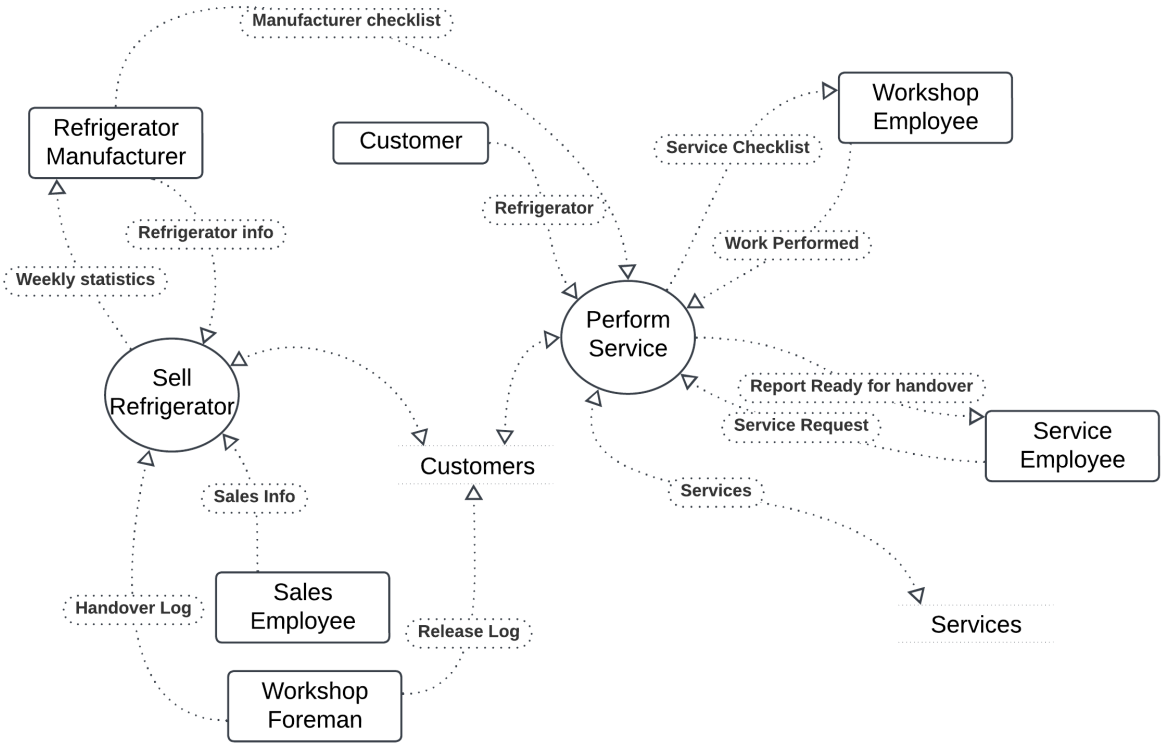
Satisfy relationship: This relationship documents that a model element of a design (the supplier). However, the “Satisfy” relationship does not document whether the model element satisfies the requirement completely or only partially. Multiple model elements can be related to the same requirement by satisfy relationships.

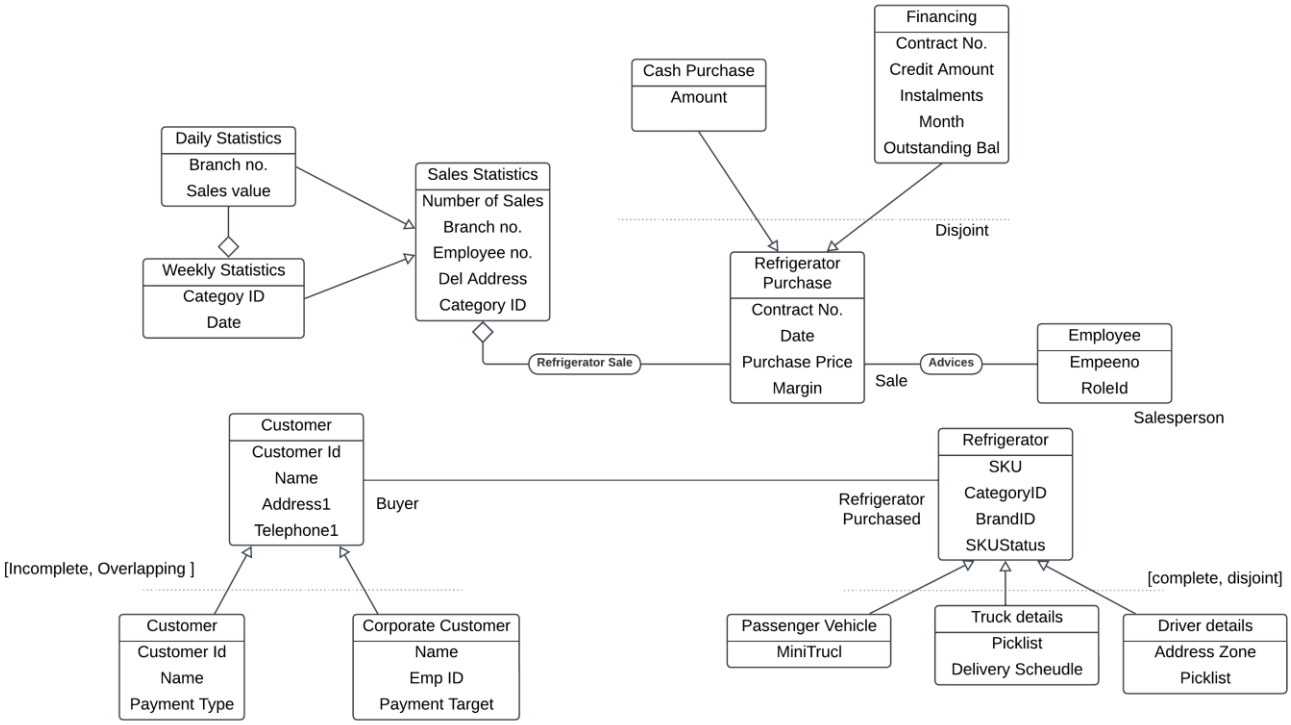
Verify relationship: This relationship relates a requirement (supplier) to a test case (client) that can be used to verify that the system satisfies the requirement. However, the ‘verify’ relationship does not document whether the client verifies the requirement completely or only partially.

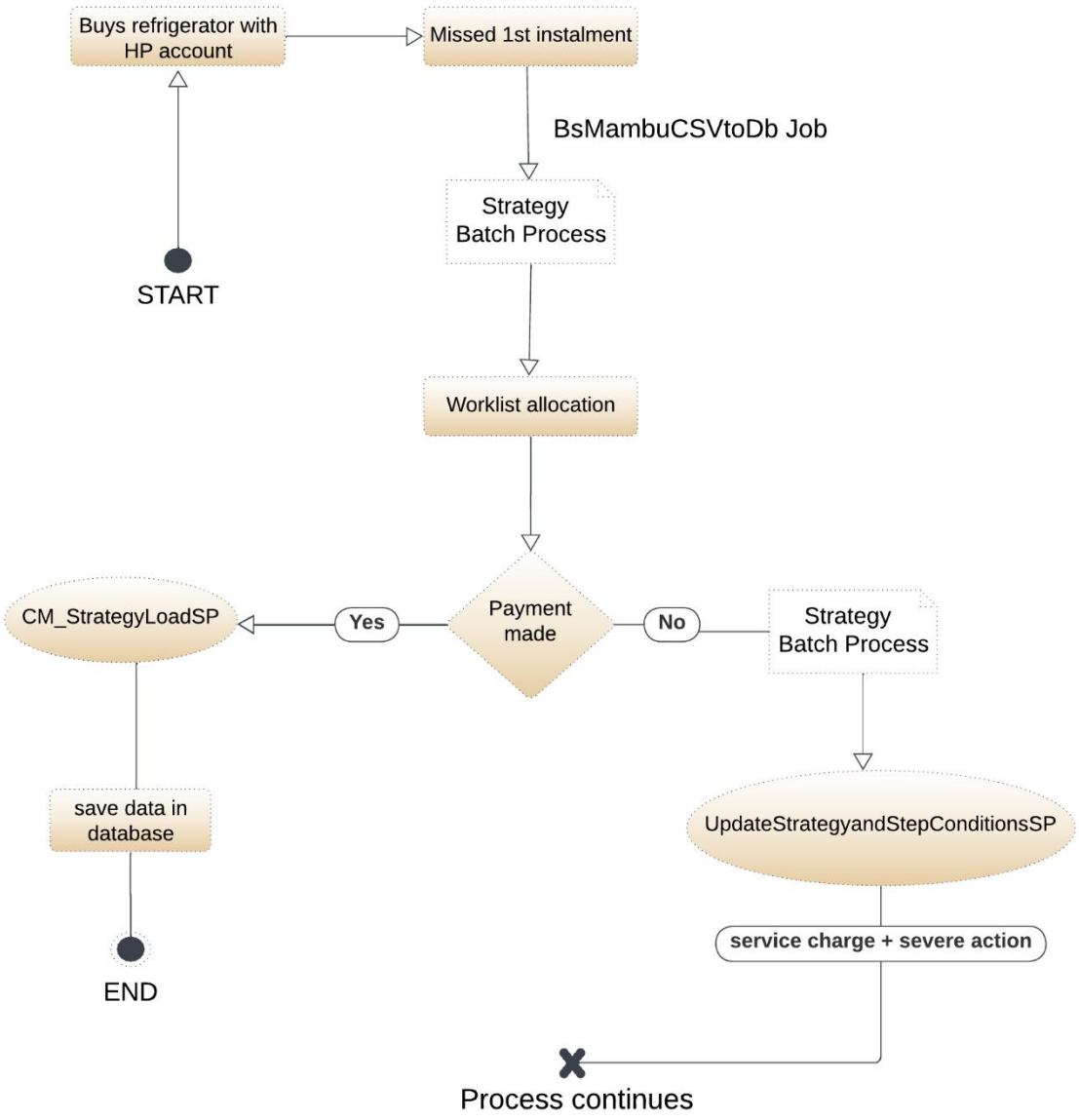
Trace relationship: This relationship is used to interrelate a requirement with any type of model element. It documents a generic relationship between between the requirement and the other model element.

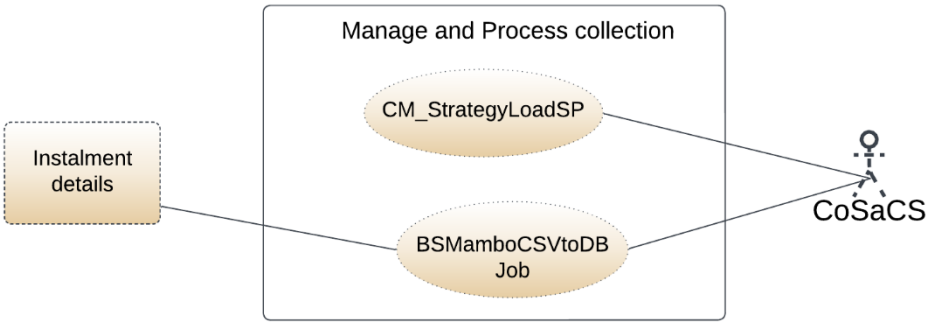
Data Flow Diagrams.

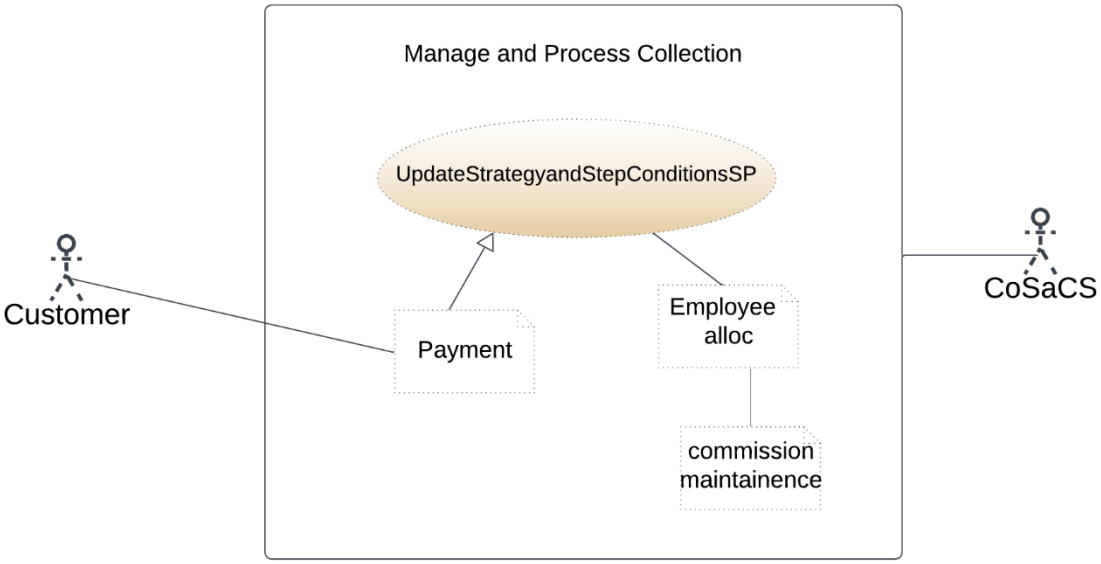
A data flow “Sale order for refrigerator” using ‘Sell Refrigerator’ and ‘Perform service’ as a process, data store ‘customers’, ‘services’ to source/sink customer, ‘refrigerator manufacturer’, ’Sales employee’, ‘Workshop Foreman’, ‘Service Employee’, ‘Workshop employee’

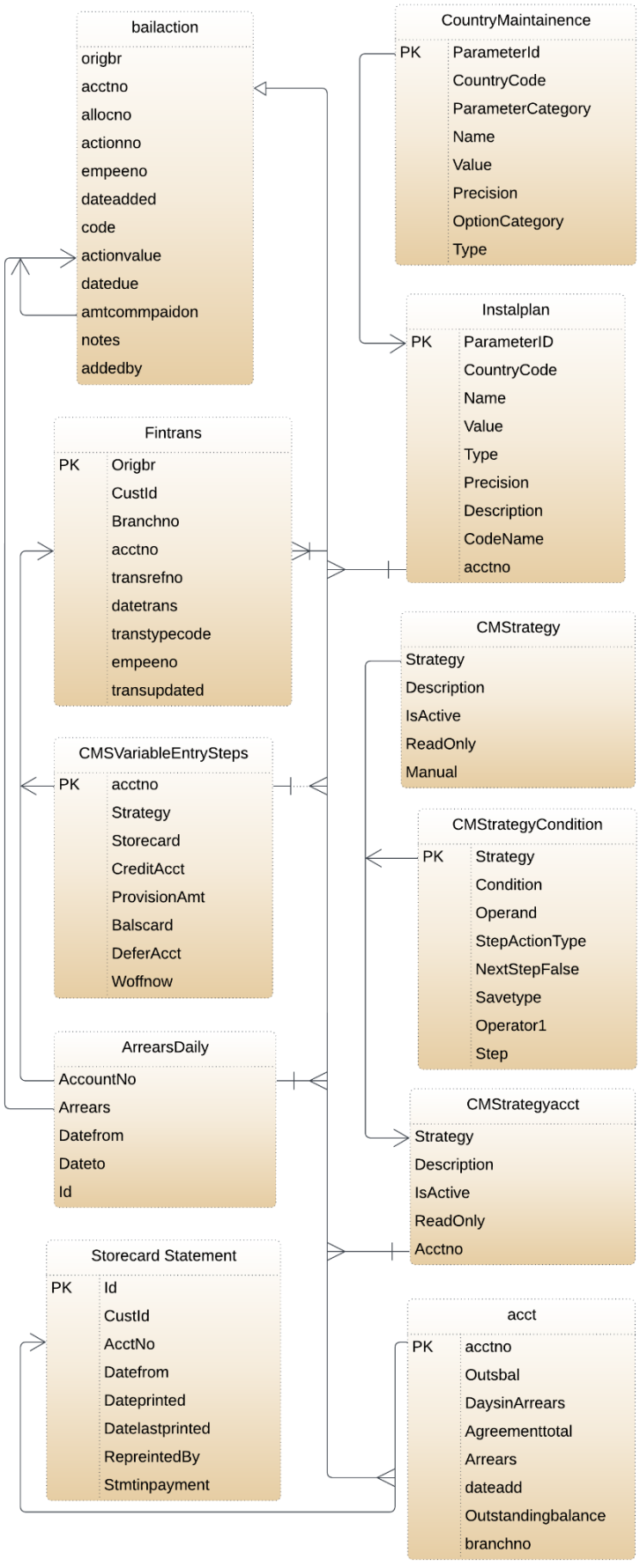


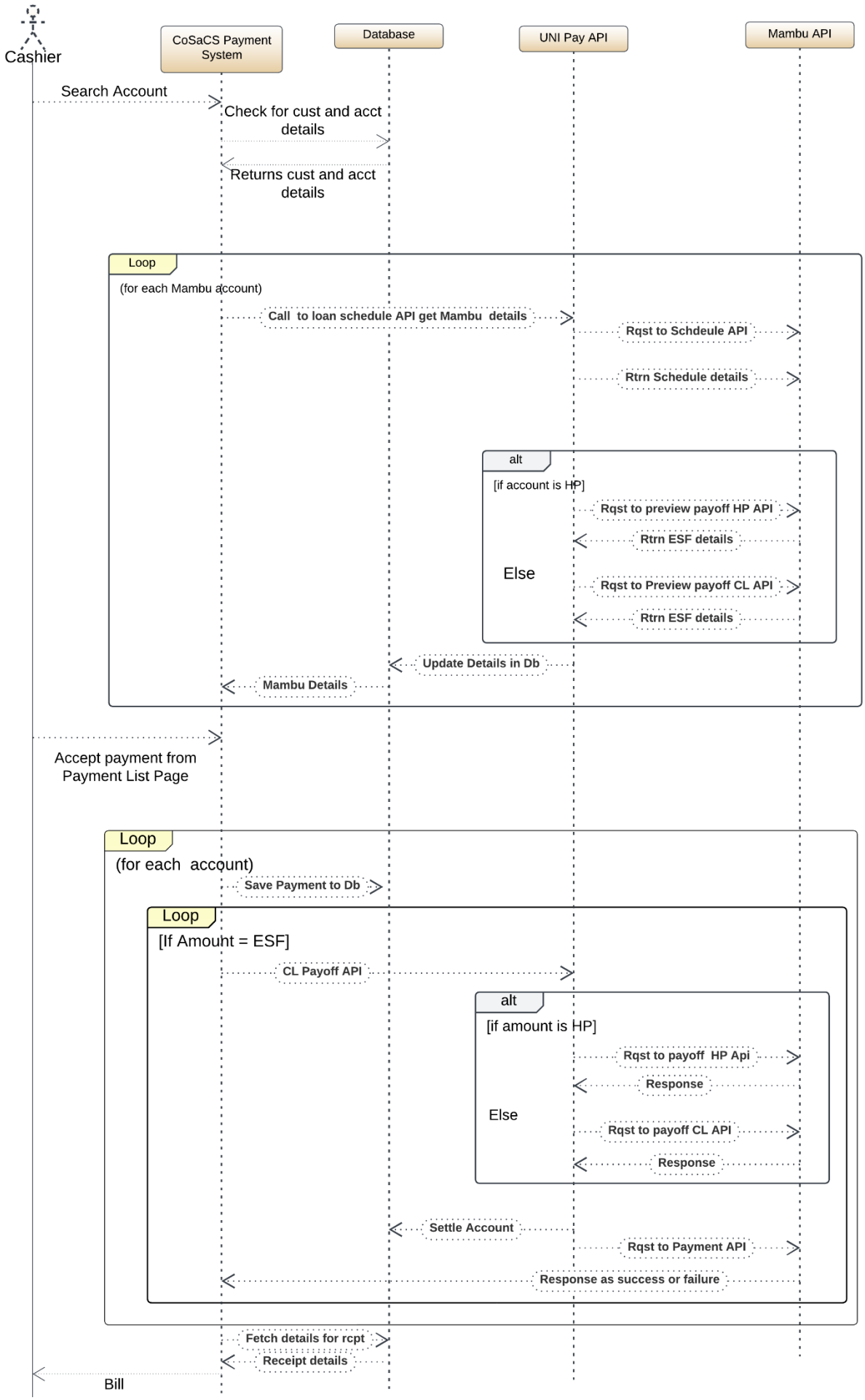

Financing
Contract No.
Credit Amount
Instalments
Month
Outstanding Bal
Cash Purchase
Amount
Daily Statistics
Branch no.
Sales value
Sales Statistics
Number of Sales
Branch no.
Employee no.
Del Address
Category ID
Disjoint
Weekly Statistics
Categoy ID
Date
Refrigerator Purchase
Contract No.
Date
Purchase Price
Margin
Refrigerator Sale
Sale
Advices
Employee
Empeeno
RoleId
Salesperson
Customer
Customer Id
Name
Address1
Telephone1
Buyer
Refrigerator
SKU
CategoryID
BrandID
SKUStatus
Refrigerator Purchased
[Incomplete, Overlapping]
[complete, disjoint]
Customer
Customer Id
Name
Payment Type
Corporate Customer
Name
Emp ID
Payment Target
Passenger Vehicle
MiniTrucl
Truck details
Picklist
Delivery Scheudle
Driver details
Address Zone
Picklist


Buys refrigerator with HP account
Missed 1st instalment
BsMambuCSVtoDb Job
Strategy Batch Process
START
Worklist allocation
CM_StrategyLoadSP
Yes
Payment made
No
Strategy Batch Process
save data in database
UpdateStrategyandStepConditionsSP
service charge + severe action
END
Process continues


Manage and Process collection
CM_StrategyLoadSP
Instalment details
BSMamboCSVtoDB Job
CoSaCS


Manage and Process Collection
UpdateStrategyandStepConditionsSP
Customer
Payment
Employee alloc
commission maintainence
CoSaCS


bailaction
origbr
acctno
allocno
actionno
empeeno
dateadded
code
actionvalue
datedue
amtcommpaidon
notes
addedby
CountryMaintainence
PK
ParameterId
CountryCode
ParameterCategory
Name
Value
Precision
OptionCategory
Type
Instalplan
PK
ParameterID
CountryCode
Name
Value
Type
Precision
Description
CodeName
acctno
Fintrans
PK
Origbr
CustId
Branchno
acctno
transrefno
datetrans
transtypecode
empeeno
transupdated
CMStrategy
Strategy
Description
IsActive
ReadOnly
Manual
CMSVariableEntrySteps
PK
acctno
Strategy
Storecard
CreditAcct
ProvisionAmt
Balscard
DeferAcct
Woffnow
CMStrategyCondition
PK
Strategy
Condition
Operand
StepActionType
NextStepFalse
Savetype
Operator1
Step
ArrearsDaily
AccountNo
Arrears
Datefrom
Dateto
Id
CMStrategyacct
Strategy
Description
IsActive
ReadOnly
Acctno
Storecard Statement
PK
Id
CustId
AcctNo
Datefrom
Dateprinted
Datelastprinted
RepreintedBy
Stmtinpayment
acct
PK
acctno
Outsbal
DaysinArrears
Agreementtotal
Arrears
dateadd
Outstandingbalance
branchno


Cashier
CoSaCS Payment System
Database
UNI Pay API
Mambu API
Search Account
Check for cust and acct details
Returns cust and acct details
Loop
(for each Mambu account)
Call to loan schedule API get Mambu details
Rqst to Schdeule API
Rtrn Schedule details
alt
[if account is HP]
Rqst to preview payoff HP API
Rtrn ESF details
Else
Rqst to Preview payoff CL API
Rtrn ESF details
Update Details in Db
Mambu Details
Accept payment from Payment List Page
Loop
(for each account)
Save Payment to Db
Loop
[If Amount = ESF]
CL Payoff API
alt
[if amount is HP]
Rqst to payoff HP Api
Response
Else
Rqst to payoff CL API
Response
Settle Account
Rqst to Payment API
Response as success or failure
Fetch details for rcpt
Receipt details
Bill