



# GrowLink Deliverable #3: Domain Class Model

## I. Domain Classes: Noun Technique Step-by-Step

### Step 1: Extract Nouns and Noun Phrases

The table below summarizes the key nouns and noun phrases extracted from the GrowLink System Vision Document and its functional requirements.

Source Noun/Phrase	Source Context
<b>Supplier</b>	Connects <b>suppliers</b>
<b>Agro-Retailers</b>	Connects <b>agro-retailers</b>
<b>Logistics Provider</b>	Connects <b>logistics providers</b>
<b>Financial Institutions</b>	Connects <b>financial institutions</b>
<b>System</b>	Single integrated <b>system</b>
<b>Agri-Input</b>	Simplify <b>agri-input</b> procurement
<b>BNPL (Buy Now Pay Later)</b>	Credit solutions such as <b>BNPL</b>

<b>Mobile App</b>	Responsive web application and a <b>mobile app</b>
<b>APIs</b>	Integration of secure <b>APIs</b>
<b>Users</b>	Centralized database for <b>users</b>
<b>Products</b>	Centralized database for <b>products</b>
<b>Transaction Data (Transaction)</b>	Centralized database for <b>transaction data</b>
<b>Order</b>	Retailers can place, modify, and track <b>orders</b>
<b>Order Status</b>	Automated <b>order status</b> updates
<b>Payment Gateways</b>	Secure digital payments via integrated <b>payment gateways</b>
<b>Analytics</b>	Implementation of <b>analytics</b> and reporting
<b>Sales Trends</b>	Dashboards showing <b>sales trends</b>

## Step 2: Filter and Refine Candidate Classes

We filter the list by eliminating implementation details, synonyms, and attributes to arrive at the final set of Domain Classes.

Candidate Class	Rationale (Keep / Reject / Attribute)
Supplier	<b>KEEP:</b> Core entity, distinct attributes (Manage Products).
Retailer	<b>KEEP:</b> Core entity, distinct attributes (Place Orders).
Logistics Provider	<b>KEEP:</b> External entity requiring configuration/tracking data ( <b>LogisticsPartner</b> ).
Financial Institutions/BNPL	<b>KEEP:</b> External entity requiring configuration/contract details ( <b>FinancialPartner</b> ).
System / App / APIs	<b>REJECT:</b> Represents the boundary or technology, not domain data.
Users	<b>KEEP:</b> Abstract superclass for <b>Supplier</b> and <b>Retailer</b> ( <b>User</b> ).

<b>Agri-Input / Products</b>	<b>KEEP:</b> The primary item being bought and sold ( <b>Product</b> ).
<b>Order</b>	<b>KEEP:</b> The core transactional document.
<b>Transaction Data (Transaction)</b>	<b>KEEP:</b> The financial record associated with an order.
<b>Payment Gateways</b>	<b>KEEP:</b> External entity requiring setup ( <b>PaymentGateway</b> ).
<b>Order Status</b>	<b>ATTRIBUTE:</b> An attribute of <b>Order</b> .
<b>Sales Trends / Analytics</b>	<b>KEEP:</b> Represents persistent compiled data ( <b>Report</b> or <b>AnalyticsData</b> ).

## Domain Classes for GrowLink

Class Name	Rationale	Key Attributes (Minimum)
<b>User</b>	Abstract superclass for all system accounts (Retailer/Supplier).	userID, email, passwordHash
<b>Retailer</b>	Core buying entity; subclass of <b>User</b> .	businessName, location, creditLimit
<b>Supplier</b>	Core selling entity; subclass of <b>User</b> .	companyName, contactInfo
<b>Product</b>	The central item being sold (agri-input).	productID, name, description, price, stockQuantity
<b>Order</b>	The main transactional document containing items and status.	orderID, orderDate, totalAmount, status
<b>Transaction</b>	The financial record linked to payment/credit processing.	transactionID, amount, type (Digital/BNPL), status

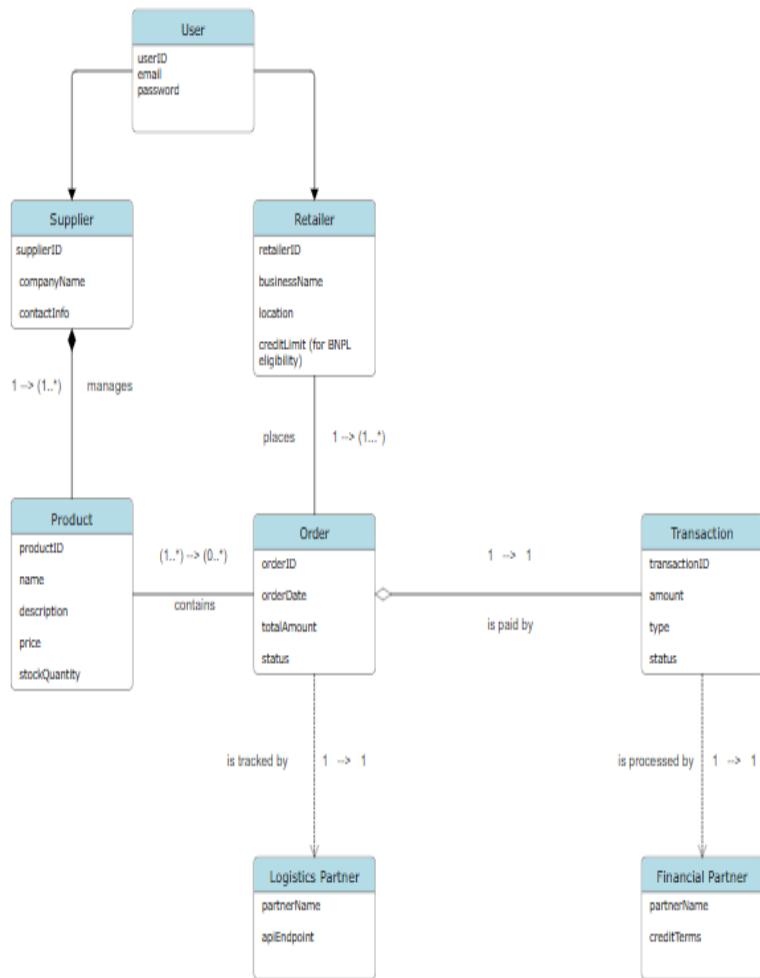
<b>LogisticsPartner</b>	Configuration and tracking details for the integrated external delivery provider.	apiEndpoint, trackingPrefix, partnerName
<b>FinancialPartner</b>	Configuration and terms for the integrated BNPL/Credit services.	bnplContractID, creditTerms, partnerName

## Final Domain Class Relationship Table for GrowLink

Source Class	Multiplicity	Relationship Type	Target Class	Multiplicity	Notes for Diagram
User		Generalization	Retailer		<i>Hollow Arrowhead:</i> User is the superclass.
User		Generalization	Supplier		<i>Hollow Arrowhead:</i> User is the superclass.
Retailer	1	Association	Order	\$0..*\$	<i>Simple Line:</i> One Retailer can place zero or many Orders.

<b>Supplier</b>	1	<b>Composition</b>	<b>Product</b>	\$1..*\$/	<i>Filled Diamond:</i> Supplier owns/manages its Products (Products cannot exist without a Supplier).
<b>Order</b>	1	<b>Aggregation</b>	<b>Transaction</b>	1	<i>Hollow Diamond:</i> Order aggregates one financial Transaction.
<b>Product</b>	\$1..*\$/	<b>Association</b>	<b>Order</b>	\$0..*\$/	<i>Simple Line:</i> Products are referenced by Orders (Line Items).
<b>Order</b>	1	<b>Dependency</b>	<b>LogisticsPartner</b>	1	<i>Dashed Arrow:</i> Order relies on the Partner for tracking data.
<b>Transaction</b>	1	<b>Dependency</b>	<b>FinancialPartner</b>	1	<i>Dashed Arrow:</i> Transaction relies on the Partner for payment/credit processing.

## Domain Class Diagram



## 1. CRUD Validation Matrix

The table below maps the 27 original Use Cases (UC-R for Retailer flow, UC-S for Supplier flow) against the core Domain Classes to identify any gaps in Create (C), Read (R), Update (U), or Delete (D) capabilities.

Domain Class	C (Create)	R (Read)	U (Update)	D (Delete)
<b>Retailer</b>	<b>GAP</b> (Requires Registration UC)	UC-R01, UC-R08 (Implicit Read for order/profile)	<b>GAP</b> (Requires Profile Management UC)	<b>GAP</b> (Requires Profile Management UC)
<b>Supplier</b>	<b>GAP</b> (Requires Registration UC)	UC-R09, UC-S05 (Implicit Read for order/products)	<b>GAP</b> (Requires Profile Management UC)	<b>GAP</b> (Requires Profile Management UC)
<b>Product</b>	<b>UC-S02</b> (Input Product Details)	<b>UC-R01</b> (Browse Products), <b>UC-S04</b>	<b>UC-S05/UC-S 06</b> (Modify Price/Stock)	<b>GAP</b> (Requires Deactivate Product UC)
<b>Order</b>	<b>UC-R02</b> (Initiate Checkout)	<b>UC-R08</b> (Review Order), UC-R12, UC-R13	<b>UC-R12</b> (Updates status), <b>UC-R14</b>	<b>UC-R15</b> (Decline/Cancel Order)

<b>Transaction</b>	<b>UC-R07</b> (Payment Confirmed)	<b>UC-R14</b> (Finalize Transaction)	<b>UC-R15</b> (Status update to Failed/Refund)	N/A (Transactions are archived)
<b>LogisticsPartner</b>	N/A (External entity)	UC-R13, UC-R11 (Read tracking data)	N/A	N/A
<b>FinancialPartner</b>	N/A (External entity)	UC-R05, UC-R06 (Read credit status)	N/A	N/A

## After Refining

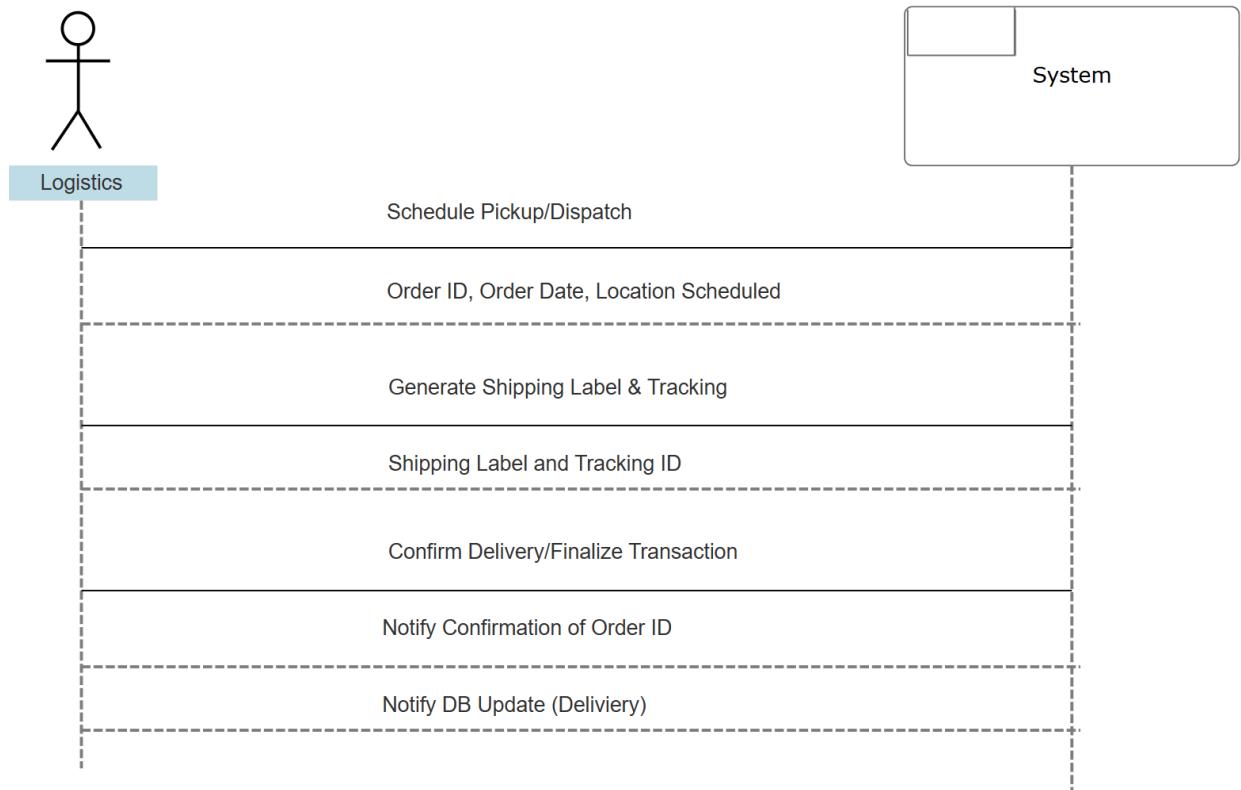
Domain Class	C (Create)	R (Read)	U (Update)	D (Delete)
<b>Retailer</b>	<b>UC-H01: Register User</b>	<b>UC-H02: Manage User Profile</b>	<b>UC-H02: Manage User Profile</b>	<b>UC-H02</b> (Deactivate/Close Account)
<b>Supplier</b>	<b>UC-H01: Register User</b>	<b>UC-H02: Manage User Profile</b>	<b>UC-H02: Manage User Profile</b>	<b>UC-H02</b> (Deactivate/Close Account)
<b>Product</b>	<b>UC-S02: Input Product Details</b>	<b>UC-R01: Browse Products, UC-S04</b>	<b>UC-S05/UC-S0 6 (Modify Price/Stock)</b>	<b>UC-H03: Deactivate Product</b>
<b>Order</b>	<b>UC-R02: Initiate Checkout</b>	<b>UC-R08: Review Order, UC-R12, UC-R13</b>	<b>UC-R12 (Status Update), UC-R14 (Finalization)</b>	<b>UC-R15</b> (Decline/Cancel Order)
<b>Transaction</b>	<b>UC-R07: Payment Confirmed</b>	<b>UC-R14: Finalize Transaction</b>	<b>UC-R15 (Status Update to Failed/Refund)</b>	N/A (Archived, not deleted)

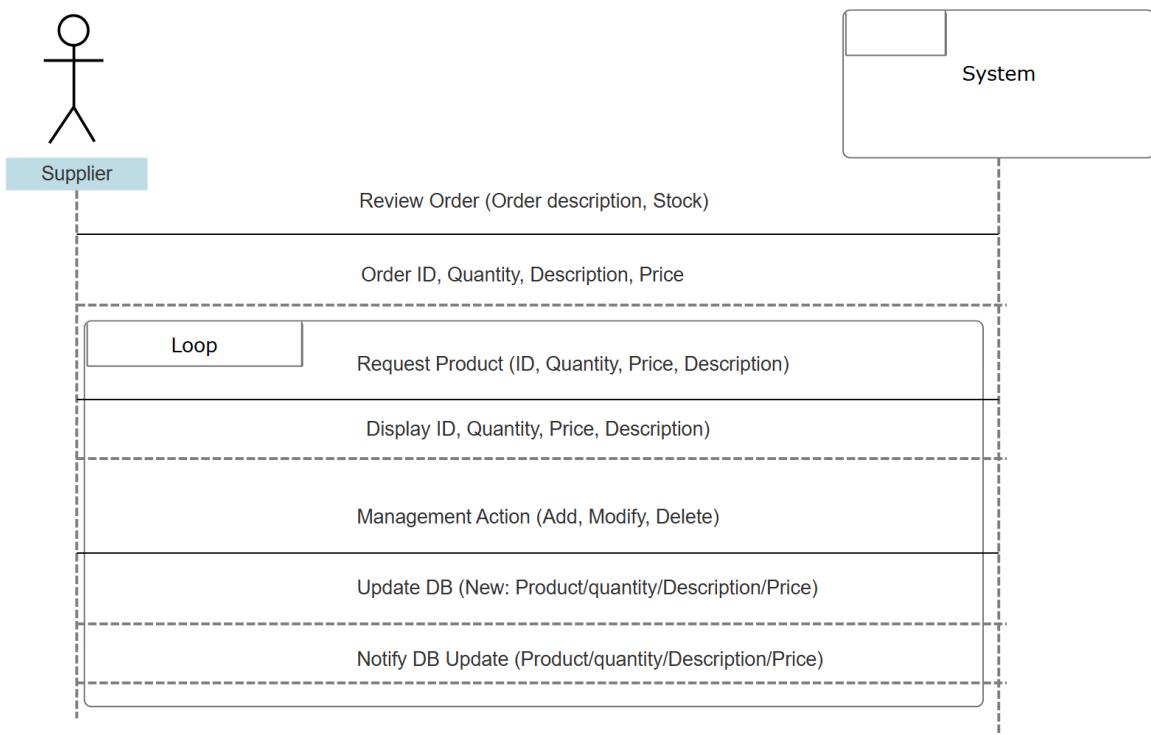
<b>LogisticsPartner</b>	N/A (External)	UC-R13, UC-R11 (Read tracking data)	N/A	N/A
<b>FinancialPartner</b>	N/A (External)	UC-R05, UC-R06 (Read credit status)	N/A	N/A

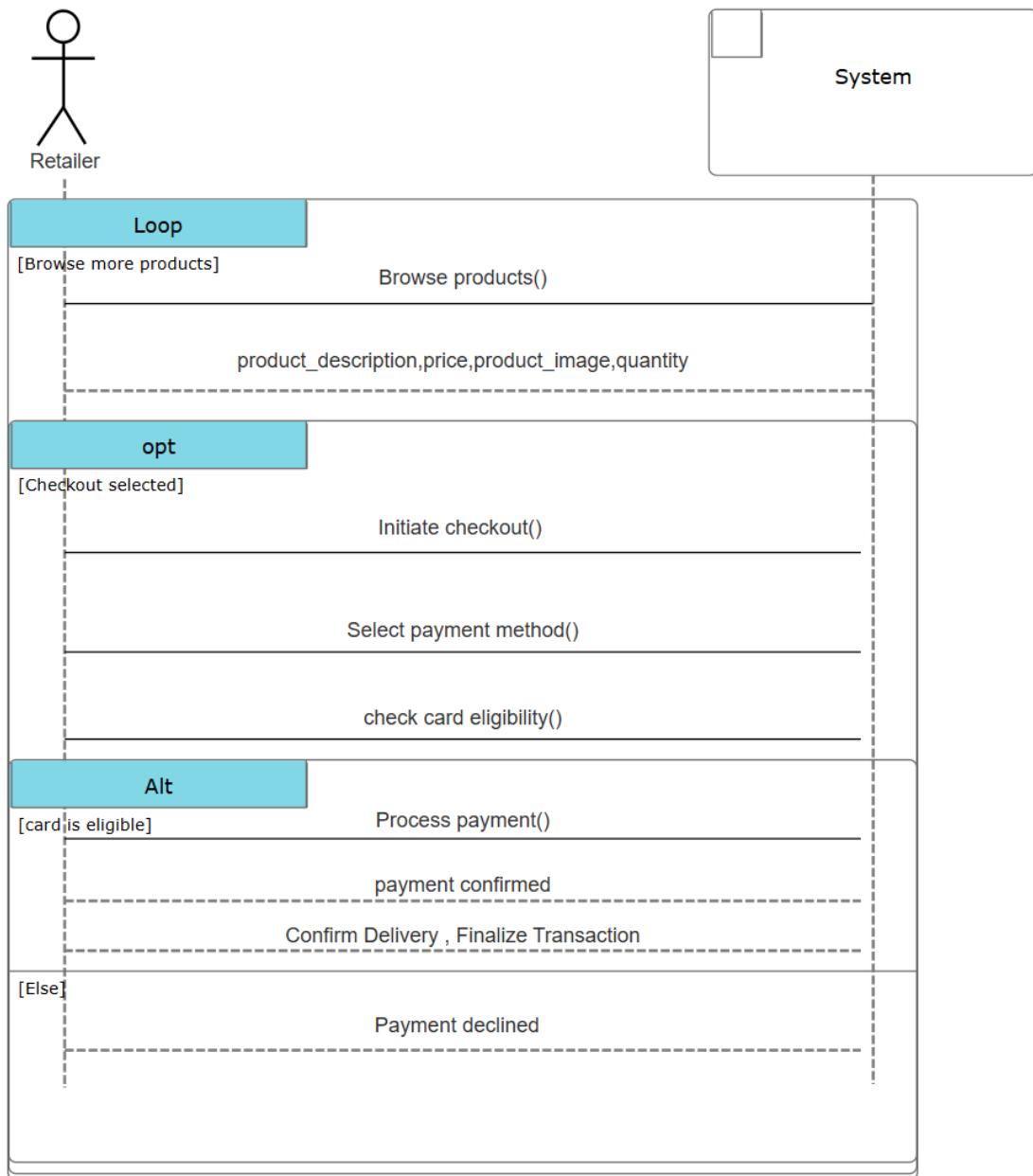
The additions are:

New Use Case ID	Use Case Name	CRUD Action Covered	Actor(s)	Rationale for Addition
UC-H 01	Register User	Create (C)	Retailer, Supplier	Provides the necessary function to create accounts for new Retailers and Suppliers, covering the 'C' for both User subclasses in the CRUD matrix.
UC-H 02	Manage User Profile	Read (R), Update (U), Delete (D)	Retailer, Supplier	Covers all profile modifications (U) like changing contact info, reading profile details (R), and account deactivation (D) for both user types.
UC-H 03	Deactivate Product	Delete (D)	Supplier	Provides the necessary function for a Supplier to remove a product from the marketplace (soft delete/deactivation), covering the 'D' for the <b>Product</b> Domain Class.

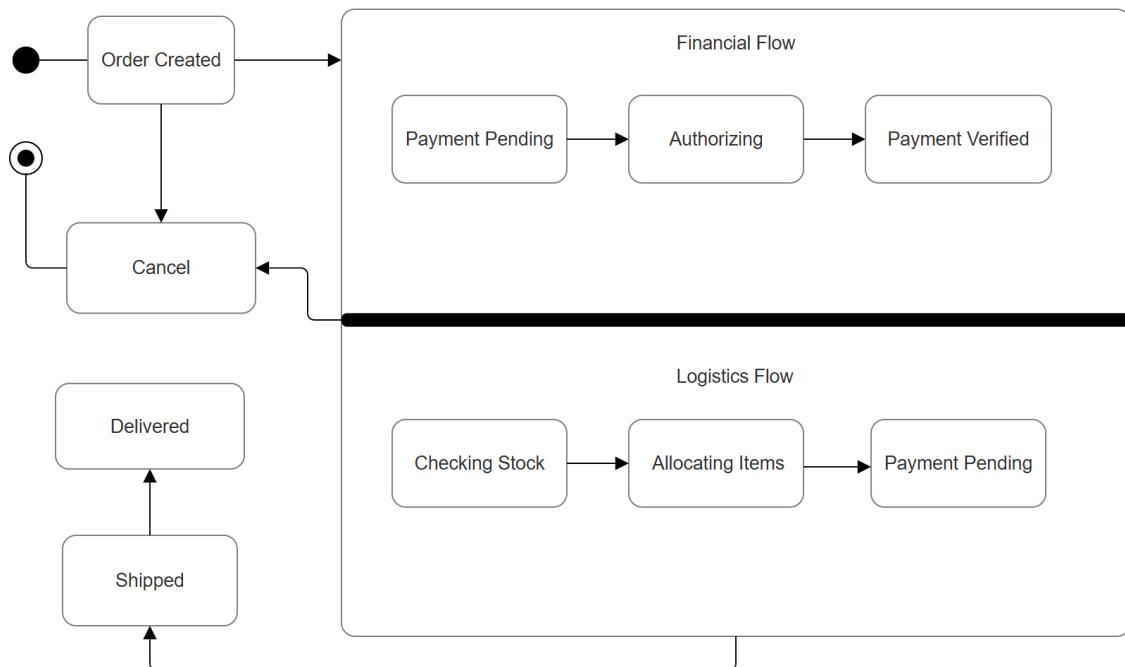
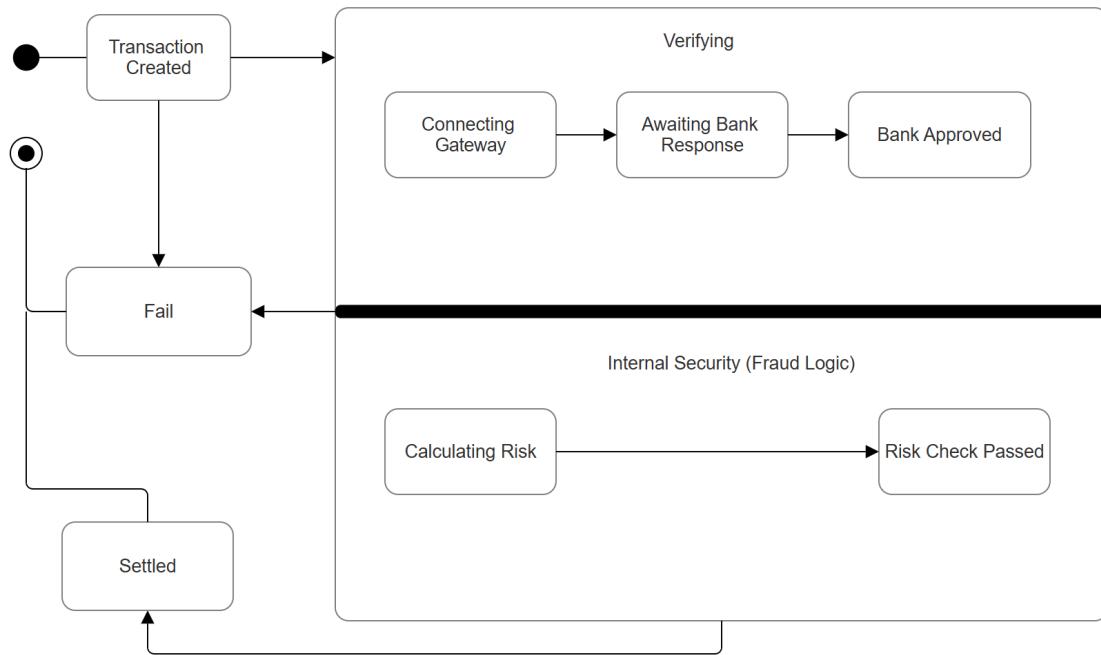
## Sequence Diagrams:







## State Machine Diagrams:



## WorkFlow Diagrams:

