

Business Use Case

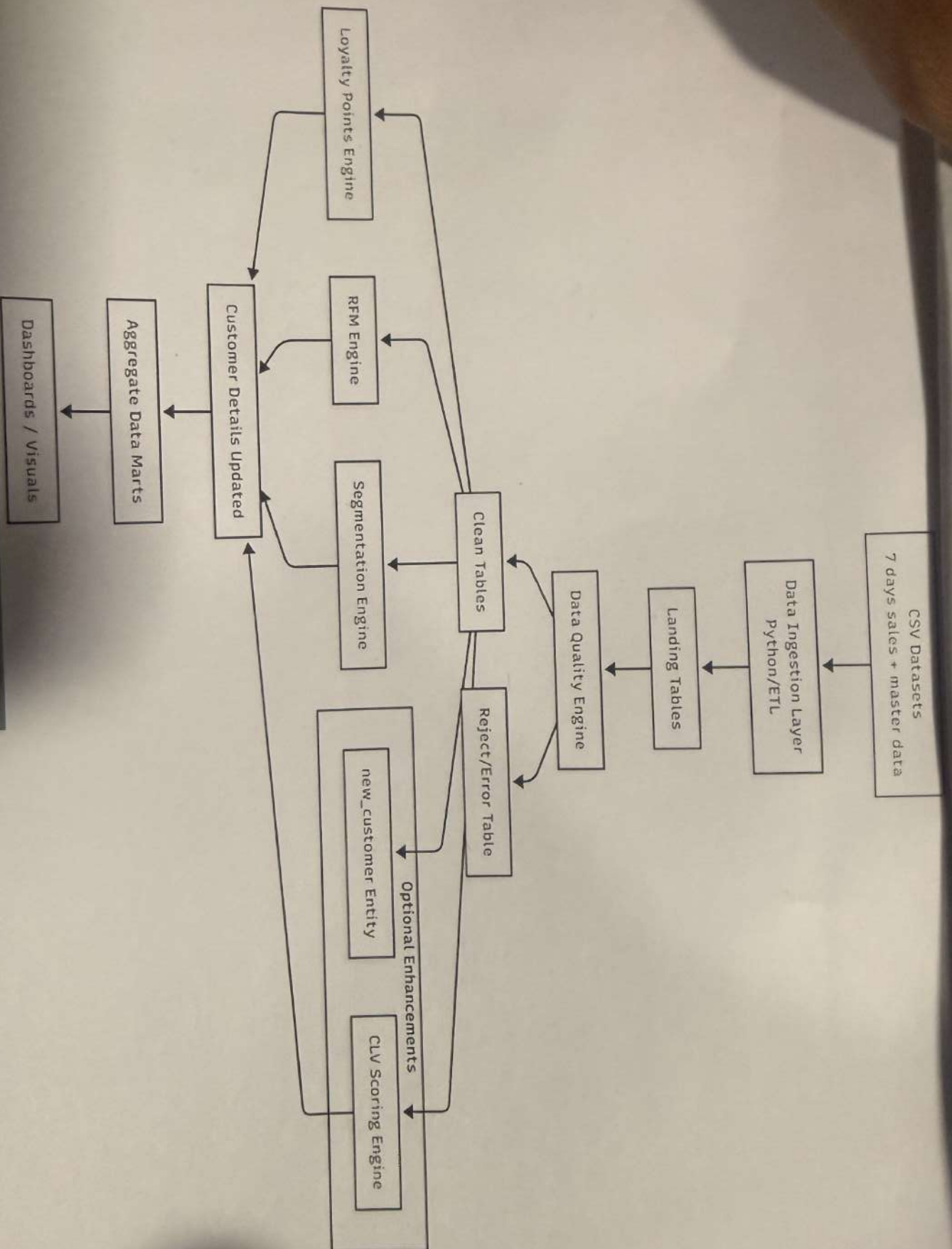
- This project enables retailers to automatically ingest and validate large volumes of daily sales data to ensure accuracy and reliability.
- It calculates loyalty points for every customer transaction and maintains an up-to-date loyalty balance.
- Using purchase history, RFM metrics, and loyalty behavior, the system segments customers into actionable groups like High-Spenders and At-Risk.
- Retailers can use these insights to run targeted promotions, improve customer retention, and boost revenue.
- Overall, the solution provides a unified view of customer value, enabling data-driven loyalty and marketing strategies.

CSV Datasets

7 days sales + master data



Data Ingestion Layer
Python/ETL



Scope for 5-Hour MVP

Data Ingestion

- Generate sample datasets using GenAI (stores, products, customers, 7-day sales)
- Load CSVs into landing tables (raw) using Python/ETL of your choice

Data Quality

- Apply basic DQ rules: Null check, Duplicate check, Invalid references (e.g., product_id not in products)
- Store: Clean data → clean tables, Bad data → reject/error table

Core Business Logic

- Loyalty Points Calculation using loyalty_rules
- Update customer total_loyalty_points
- Update last_purchase_date

Scope for 5-Hour MVP Continue...

Customer Segmentation

- Compute RFM (Recency, Frequency, Monetary)
- Identify two groups:
 - High-Spenders (Top 10%)
 - At-Risk (No purchase in 30+ days + have points)
- Update segment_id in customer_details

Output & Visualization

- Create tables with automated data ingestion with at least 1 working dashboard showing -
Customer Segments Loyalty Points Summary Basic RFM metrics

Documentation

- One ER diagram
- Push code + diagrams to GitHub

Technical Requirements

- Frontend: Any reporting tool (Power BI/Tableau/Snowflake Streamlit/Python etc.)
- Database: SQL Database of choice: Snowflake, MongoDB, Dynamo, SQL Server etc
- Data Generation: Use any GenAI/LLM tool to generate realistic, non-disjoint datasets.
- ETL / Data Processing: Python, Spark, Airflow, dbt, or any cloud ETL service (AWS Glue, Azure Data Factory, GCP Dataflow).
- Version Control: GitHub repository with all code

Key Features

Unified Data Ingestion

- Automated loading of sales, customer, product & promotion data
- Scalable for multi-store, high-volume transactions

Robust Data Quality Framework

- Rule-based validation (nulls, duplicates, FK checks)
- Outlier detection + reject/error repository

Loyalty Points Engine

- Dynamic rules for point earning & bonuses
- Real-time accrual and customer balance updates

Advanced Customer Insights

- RFM scoring (Recency, Frequency, Monetary)
- Segments: High-Spenders, At-Risk, Others

Promotion Impact Analysis

- Track promo effectiveness across products & stores
- Measure uplift in sales and loyalty point activity

Dashboard & Visualization Layer

- Insights on loyalty trends, RFM, segments, CLV & promotions
- Interactive filters and store/product drill-downs

Hackathon Structure & Sprint - 0 (Loyalty Lens Project)

Sprint 0 – Planning & Setup (First 1-1.5 hours)

👁️ Goal: Understand, design, and set up the foundation

Activities:

- Review the Loyalty Lens use case, entities, and requirements
- Define overall approach:
 - Data ingestion plan
 - Data quality rules strategy
 - Loyalty points calculation logic
 - Segmentation (RFM + loyalty) approach
- Create architecture diagram + ERD draft
- Set up GitHub repo, directory structure, and README
- Set up environment (Snowflake/Postgres, Python/ETL tool, BI tool)

Deliverable:

- ✓ Finalized solution design + architecture, ERD
- ✓ Clear task breakdown for team members
- ✓ README committed in GitHub

Hackathon Structure & Sprint – 1 (Loyalty Lens Project)

Sprint 1 – Core Development (Next 3–4 hours)

- Goal: Build ALL core functional components

Activities:

• Data Layer

- Generate 7-day datasets using GenAI
- Create database tables (stores, products, customers, promotions, sales header & line items, loyalty rules)
- Build data ingestion pipeline (CSV → DB)
- Build data quality validation

Valid/clean data → main tables

Reject/bad data → reject table

Processing Layer

- Implement loyalty points calculation for each transaction
- Update customer total loyalty balance
- Compute RFM metrics
- Build segmentation logic:

High-Spenders (Top 10%)

At-Risk (30+ days no purchase with positive points)

Deliverable:

- ✓ A working backend pipeline that loads, cleans, processes, and enriches loyalty data
- ✓ RFM + segmentation logic working end-to-end
- ✓ Mid-sprint review with mentors

Hackathon Structure & Sprint – 2 (Loyalty Lens Project)

Sprint 2 – Refinement, Testing, Demo (Final 1 hour)

🕒 Goal: Polish, test, and prepare final submission + demo

Activities:

- Final testing of ingestion, DQ, loyalty engine, segmentation
- Build a dashboard (Power BI / Tableau / Streamlit / Python) showing:
 - Loyalty points
 - Segments
 - RFM
 - Sales trends
- Package all SQL, Python, datasets, and diagrams into GitHub
- Prepare Demo script + Final deck
- Optional: Deploy Streamlit app or notebook in cloud

Deliverable:

- ✓ Fully functional MVP covering ingestion → DQ → loyalty → segmentation → dashboard
- ✓ GitHub repo with code, ERD, architecture diagram, datasets, *README*
- ✓ Final demo ready