REQUIREMENT ANALYSIS

Date	17th June 2025
Team ID	LTVIP2025TMID29987
Project Name	CRM Application for Jewelry Management – (Developer)
Maximum Marks	

CUSTOMER JOURNEY MAP

This map meticulously illustrates the step-by-step process a jewelry store customer undertakes, from their initial spark of product interest to the final notification post-payment. It critically highlights the points of interaction with the proposed CRM system, detailing how each system interaction not only supports the customer's action but also streamlines the store's operations and enhances data capture for future engagement.

Step	Customer Action (Detailed Scenario & Customer Intent)	System Interaction (Detailed CRM Role & Data Capture)
1	**Enters store or contacts seller** A potential customer walks into the physical jewelry store, or perhaps initiates contact online (via website chat, phone call, or social media inquiry). Their primary intent is to explore options, ask questions, or seek specific items or services.	**New customer record created (Jewel Customer)** Upon first interaction, the sales associate or system automatically captures essential customer details (Name, Contact Information, Preferred Communication Method) into a new **Jewel Customer** record. For returning customers, their existing record is quickly retrieved, providing the associate with their history and preferences. This ensures a personalized experience from the outset and begins building a comprehensive customer profile.
2	**Views items and selects jewelry** The customer browses available jewelry pieces, tries on items, discusses options with the sales associate, and eventually makes a decision on what they wish to purchase. They might consider design, material, price, and occasion.	**Item record selected from Item__c object** As the customer shows interest or selects items, the sales associate logs these interactions. The specific **Item__c** records (e.g., Ring_001, Necklace_B2C) are identified and associated with the customer's interaction. This selection process might involve checking real-time inventory levels, viewing detailed product specifications (like gold purity,

		diamond cut, stone weight, and current market price for precious metals/stones) directly from the Item__c object.
3	**Places order** Having chosen their desired piece(s), the customer confirms their intention to purchase. This might involve discussing customization options, delivery timelines, or setting up a layaway plan. Their intent is to formalize the commitment to buy.	**Customer Order record is created** A new **Customer Order** record is generated within the CRM. This record links directly to the Jewel Customer and the selected Item__c records. It captures all order-specific details such as quantities, agreed-upon price (including any discounts or custom charges like KDM), delivery instructions, and payment terms (e.g., full payment, partial payment, layaway schedule). This centralizes all order-related information, making it accessible and trackable.
4	**Billing is generated** The customer is now ready to finalize the transaction. They expect a clear, accurate, and detailed bill that reflects their purchase, including all applicable taxes and any specific charges. Their goal is to understand the total cost before making payment.	**Billing__c record created with price, tax** The CRM system automatically generates a new **Billing__c** record, pre-populating it with data from the Customer Order and Item__c records. This record precisely calculates the Total_Amount__c due, applying complex tax slabs (e.g., GST), any KDM charges, stone charges, and discounts. This Billing__c record is intelligently linked via **Lookup relationships** to both the Jewel Customer and the Customer Order, ensuring a complete audit trail and accurate financial reporting.
5	**Makes payment** The customer tenders payment using their preferred method (cash, card, UPI, bank transfer, etc.). This might be a full payment or a partial payment, as per a layaway or installment plan. Their intent is to complete the financial transaction.	**Paid_Amount__c auto-updated via trigger** Upon receiving and recording a payment, whether full or partial, an intelligent **Apex Trigger** or a **Record-Triggered Flow** on the Billing__c object is activated. This automation precisely updates the **Paid_Amount__c** field, ensuring the Outstanding_Amount__c is immediately and accurately reflected. For partial payments, the system maintains a clear record of the payment history, which is crucial for managing customer

		accounts and preventing billing disputes.
6	**Receives confirmation email** After successful payment, the customer expects a confirmation or receipt. This provides reassurance, a record of their purchase, and often details for follow-up (e.g., warranty, pickup details). This fulfills their need for transparency and validation.	**Flow sends email using customer email ID** A robust **Record-Triggered Flow** is executed automatically upon the finalization of the billing record or confirmation of full payment. This Flow dynamically retrieves the customer's primary email address from their **Jewel Customer** record (via a **Lookup**). Using a pre-designed **Email Template**, the Flow dispatches a personalized confirmation email, including details like the purchased items, total amount paid, any remaining balance, and delivery/pickup instructions. This proactive communication significantly enhances customer satisfaction and reduces inbound inquiries.

DATA FLOW DIAGRAM (DFD)

Level 0 Description :

The Level 0 DFD is the **highest-level overview** of a system. It's also known as the **Context Diagram**.

Single Process: It shows the entire system as a single process (often represented as a single circle or rounded rectangle). In your case, this is the entire "CRM for Jewel Management" system.

- **External Entities:** It identifies the external entities that interact with the system. In your diagram, this is primarily the **[Customer]**.
- **Data Flows:** It shows the major data flows *between* the external entities and the single system process. It doesn't show internal processes or data stores within the system itself at this level, only the primary inputs and outputs from and to the external world.

Level 0 diagram:

[Customer] --> (Jewel Customer Object) --> (Customer Order Object) --> (Item__c) --> (Billing__c) --> [Email Flow]

This is a somewhat simplified Level 0 representation, as traditionally a Level 0 DFD would show:

Level 0 – Context Diagram

Entities:

- **Customer**
- **Admin (Goldsmith, Staff)**
- **System (CRM)**

Data Flows:

- **Customer sends order details**
- **Admin inputs item and billing info**
- **System processes and stores billing**
- **Email notification sent to customer**

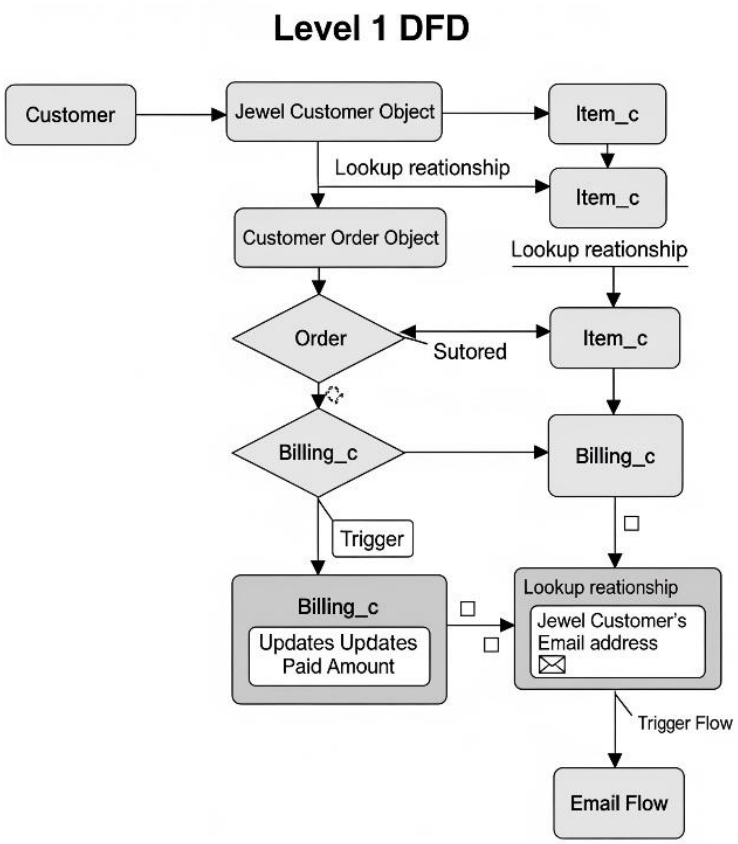
[Customer] --> (CRM for Jewel Management System) --> [Email Notification]

- And then the internal objects like "Jewel Customer Object," "Customer Order Object," etc., would be elaborated in a **Level 1 DFD** which breaks down that single system process into its major sub-processes and data stores.

Level 1 Description:

- **Customer selects an item:** This action is captured by linking the customer's choice to an existing Item__c object record via a Lookup relationship.

- **Order is placed:** All relevant order data, including selected items, quantities, and customer details, is securely stored in a Customer Order record.
- **Billing created:** A new Billing__c record is generated. This record stores pricing information, applies relevant taxes, and includes Lookup references to both the associated Customer and Item records to ensure data integrity.
- **Trigger updates Paid Amount:** Upon a payment transaction being recorded, a system trigger automatically updates the Paid_Amount__c field within the Billing__c record, reflecting the amount received.
- **Flow sends email:** Following a successful billing or payment, a pre-configured Flow is initiated. This Flow utilizes a Lookup relationship to retrieve the customer's email ID from their Jewel Customer record and sends an automated confirmation email.



Level 1 – DFD

Step	Process	Input	Output	Data Store
1	Create Jewel Customer	Customer info	Jewel_Customer__c record	Jewel_Customer__c
2	Select Item	Ornament details	Item__c record	Item__c

3	**Place Order**	**Selected Item**	**Customer_Order__c**	**Customer_Order__c**
4	**Generate Billing**	**Order + Prices**	**Billing__c record**	**Billing__c**
5	**Auto-update Payment**	**Paying amount**	**Paid_Amount__c update via Apex**	**Billing__c**
6	**Send Email**	**Billing record**	**Email to customer**	**Sent Log / Notification**

SOLUTION REQUIREMENTS

Functional Requirements:

- **Ability to create customer records (Jewel Customer):** The system must allow for the creation, storage, and management of comprehensive customer information.
- **Ability to create item records (Item__c):** The system must support the creation, detailed categorization, and management of all jewelry items (gold, silver, diamonds, etc.).
- **Create and track Customer Orders:** The system must provide functionality to generate, track the status of, and manage all customer orders from initiation to fulfillment.
- **Auto-update Paid Amount based on Paying Amount (Trigger):** A system trigger must automatically calculate and update the Paid_Amount__c field in the Billing__c object whenever a payment is recorded, ensuring financial accuracy.
- **Send confirmation emails after billing (Flow):** An automated Flow must be in place to send transaction confirmation emails to customers upon successful billing or payment completion.
- **Generate reports and dashboards:** The system must enable users to create, customize, and view various reports and dashboards for insightful analysis of sales, inventory, customer data, and financial performance.
- **Validate Paid Amount <= Total Amount (Validation Rule):** A validation rule must be implemented to prevent data entry errors by ensuring that the Paid_Amount__c never exceeds the Total_Amount__c on a Billing__c record.
- Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	- Registration through Form- Registration through Gmail- Registration through LinkedIn
FR-2	User Confirmation	- Confirmation via Email- Confirmation via OTP
FR-3	Jewelry Inventory Management	- Create Item records- Categorize by type (Gold, Silver)- Track KDM & stone charges
FR-4	Customer Order Management	- Create customer order records- View past orders by customer- Link items with orders
FR-5	Billing Automation	- Generate Billing__c from Orders- Auto-calculate total amount- Apply taxes, discounts
FR-6	Payment Update	- Auto-update Paid Amount via Trigger- Validate Paid Amount ≤ Total Amount
FR-7	Communication via Email	- Send confirmation email on billing- Use Flow with customer email lookup
FR-8	Role-Based Access	- Define profiles for Goldsmith, Worker- Set permissions using Permission Sets
FR-9	Reports & Dashboards	- Generate reports for Billing, Orders, Items- Create dashboards for performance tracking
FR-10	Data Validation & Integrity	- Use validation rules to ensure correct data- Use Lookup

		relationships to connect objects

Non-Functional Requirements:

- **System responsiveness and low latency:** The system should operate quickly and efficiently, with minimal delays in response times for all user interactions.
- **Ensure user access control (Profile & Permission Sets):** Robust security measures, including Profiles and Permission Sets, must be implemented to control user access to data and functionalities based on their roles.
- **Secure email communication:** All automated email communications sent from the system must adhere to security best practices to protect customer data and prevent unauthorized access.
- **Maintain data integrity across related objects:** The system must ensure consistency and accuracy of data across all linked objects (e.g., Customer, Item, Order, Billing) through appropriate relationships and validation.
- **Maintain audit trail with field history tracking:** The system should track changes to key fields on important objects (e.g., Billing, Customer Order) to provide an audit trail for accountability and historical analysis.
- Following are the non-functional requirements of the proposed solution.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The CRM must be intuitive and user-friendly for store staff, goldsmiths, and administrators.
NFR-2	Security	Enforce access control using profiles and permission sets to protect sensitive billing and order data.
NFR-3	Reliability	Ensure the system works consistently without errors during all standard operations.
NFR-4	Performance	System should load records, trigger Flows, and generate reports with minimal latency.
NFR-5	Availability	The system should be accessible during all store operating hours with minimal downtime.
NFR-6	Scalability	Capable of supporting increased customers, items, and transactions as the business expands.
NFR-7	Maintainability	Easy to update and enhance automation components like Flows and Triggers as business logic evolves.
NFR-8	Auditability	Field history tracking must be enabled on key objects to provide change logs for transparency.

TECHNOLOGY STACK

The CRM for Jewel Management project was developed on the Salesforce platform, leveraging its low-code and pro-code capabilities. The system incorporates custom data modeling, automation, validation, reporting, and role-based access control, making it a complete CRM tailored for jewelry store operations.

Category	Tools/Technologies Used	Explanation
Platform	**Salesforce Lightning**	The entire application is built on Salesforce Lightning Experience, which provides a modern UI and component-based architecture, improving user interaction speed and clarity.
Automation	**Record-Triggered Flows**, **Workflow Rules**	- **Flows** are used for automating actions like sending emails after billing updates.- **Workflow Rules** can trigger simple actions like field updates or alerts.
Scripting	**Apex Triggers**	Apex Triggers handle logic such as calculating Paid Amount on billing updates or inserts. It helps automate backend processes based on user actions.
Data Modeling	**Custom Objects**: Billing__c, Item__c, Jewel_Customer__c, Customer_Order__c	Custom objects are used to represent real-world entities like customers, items, orders, and billing. Relationships (lookup fields) are established among them.
Validation & Rules	**Validation Rules**, **Formula Fields**	- **Validation Rules** prevent incorrect or overpaid entries (e.g., Paid_Amount > Total_Amount).- **Formula Fields** calculate KDM charges, total weight, amount, etc.
Communication	**Email Alerts**, **Email Templates**, **Flows**	Record-Triggered Flows are configured to send email notifications using Email Templates. This automates customer communication upon actions like billing creation.
Reporting & Insights	**Reports**, **Dashboards**	Custom reports are created (e.g., Billing Report, Item Report), and Dashboards are built to visualize key metrics such as total sales, order history, customer count.
Access Control	**Profiles**, **Permission Sets**	Different user roles like Store Admin, Goldsmith, and Workers are granted specific access via Profiles. **Permission Sets** are used for granular access control.

Why This Stack Was Chosen:

- **Salesforce Lightning** allows rapid development without compromising on scalability.
- **Apex** gives flexibility to implement business-specific logic beyond what declarative tools can achieve.
- **Flows and Email Alerts** reduce manual work by automating communication and background tasks.
- **Reports and Dashboards** ensure that store owners can track performance, inventory, and financial data with ease.
- **Validation and Formula Fields** ensure data integrity and accurate calculations without needing manual entries.

Sample Tools Used in Development:

- **Object Manager** – to create custom fields and objects.
- **Flow Builder** – to automate processes like sending emails.
- **Developer Console** – for writing and testing Apex triggers.
- **Email Template Builder** – for creating formatted confirmation messages.
- **Report Builder** – to customize and visualize tabular and summary reports.
- **Setup Menu (Profiles/Permission Sets)** – to control field/object-level access based on user roles.

TECHNOLOGY STACK

The **CRM Application for Public Transport Management System** was developed on the **Salesforce platform**, leveraging its robust low-code and pro-code capabilities to automate public transport operations such as trip scheduling, fare management, bus tracking, and employee role assignment. The system integrates custom data modeling, backend automation, real-time validations, reports, dashboards, and secure access control, ensuring reliability, scalability, and efficiency.

Category	Tools/Technologies Used	Explanation
Platform	Salesforce Lightning Experience	The entire system is developed using Salesforce Lightning UI, which enhances user experience through a fast, responsive, and component-based interface.
Automation	Record-Triggered Flows, Scheduled Flows	- Flows automate critical tasks such as fare fetching based on route and bus type, and email notifications for trip assignments.
Scripting	Apex Triggers and Apex Classes	Apex is used to validate that only eligible employees (Driver/Conductor) are assigned to trips and enforce role-specific logic.
Data Modeling	Custom Objects: Bus__c, Trip__c, Ticket_Fare__c, Employee__c	Custom objects represent operational units such as buses, trips, fares, and staff. Lookup relationships link these entities for accurate data mapping.
Validation & Rules	Validation Rules, Formula Fields	- Validation rules ensure correct assignments (e.g., only drivers can be assigned as drivers).- Formula fields calculate fares and trip info.
Communication	Email Alerts, Record-Triggered Flows, Email Templates	Automated flows send trip assignment confirmations and fare update emails using pre-designed templates, ensuring timely communication.
Reporting & Insights	Custom Reports, Dashboards	Reports analyze trip frequency, bus utilization, and fare data. Dashboards visualize performance metrics such as trips per route and employee allocation.
Access Control	Profiles, Permission Sets	Profiles and Permission Sets define what data and functionality is accessible to roles like Admin, Dispatcher, Driver, and Conductor.

Why This Stack Was Chosen:

- **Salesforce Lightning** provides a fast, modular UI suitable for managing complex transport data.

- **Apex scripting** enables enforcement of specific transport policies (e.g., employee-role mapping).
- **Flows and Email Automation** streamline backend operations such as fare auto-fetching and trip notifications.
- **Reports and Dashboards** allow stakeholders to monitor key KPIs like daily trips, fare collection, and bus usage.
- **Validation Rules and Formula Fields** ensure real-time data integrity and calculated accuracy in fare and trip assignments.

Sample Tools Used in Development:

- **Object Manager** – for defining custom fields and relationships across Bus, Trip, Fare, and Employee objects.
- **Flow Builder** – to create automation for fare calculation and email notifications.
- **Apex Developer Console** – for writing and testing Apex Triggers to enforce employee role logic.
- **Email Template Builder** – for designing professional trip assignment and fare summary messages.
- **Report Builder** – to generate analytical and operational reports (e.g., Trips per Bus, Fare by Route).
- **Setup Menu (Profiles/Permission Sets)** – to restrict access and enable role-based usage of features.