# Project Architecture

**Introduction:**

This document outlines the architecture of a charterRewardsprogram architecture, with integrated security using a security adapter. It describes the components, their interactions, and the flow of data within the application.

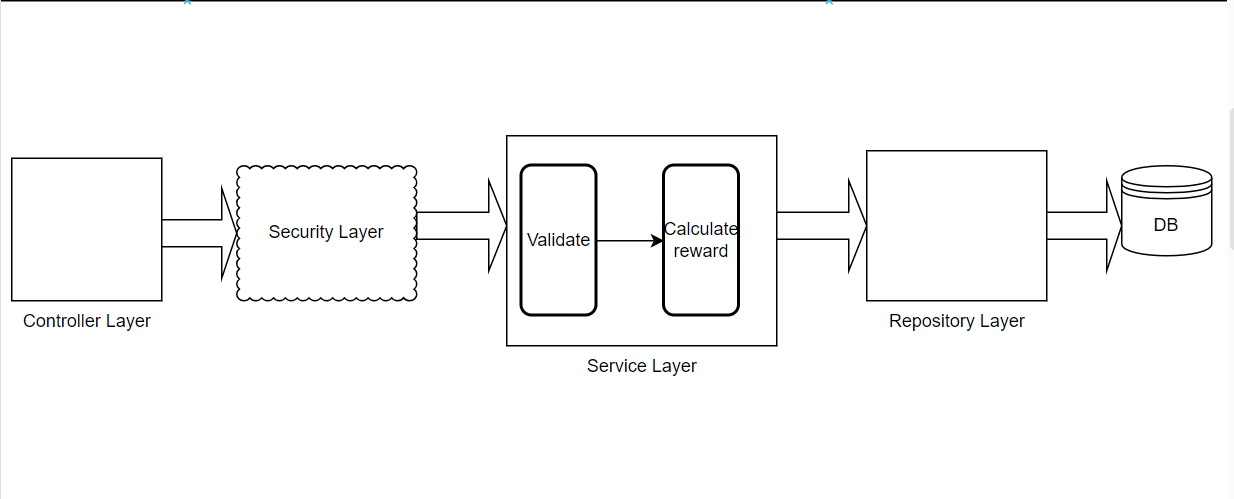
**Architecture Overview:** It includes the following key components

**Controller Layer**: Restful API endpoints through which we can interact and get the information.

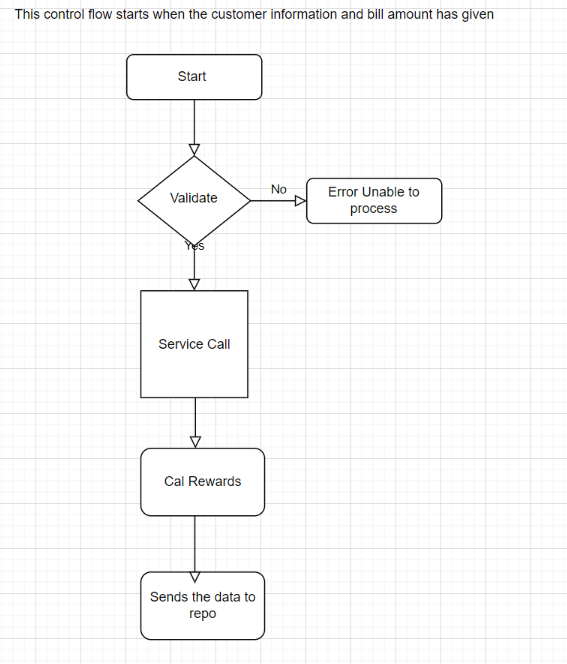
**ServiceLayer:** Business logic encapsulated in service classes.

**Data Access Layer:** Persistence managed through spring data repositories

**Security Layer:** Authorization provided by spring security with custom security adapter



**Flow Chart:**

****

# Database Design

Explanation:

1. Customer Table:
   * id: Primary key of the Customer table.
   * name, phone\_number, email: Attributes describing the customer.
   * Includes a list (collection) of Transaction entities associated with each Customer.
2. Transaction Table:
   * id: Primary key of the Transaction table.
   * amount, reward\_points: Attributes related to the transaction.
   * customer\_id: Foreign key referencing id in the Customer table (indicating a many-to-one relationship: many transactions belong to one customer).

Relationships:

* Customer to Transaction: One Customer can have multiple Transaction records. This is represented by the one-to-many relationship arrow from Customer to Transaction.
* Transaction to Customer: Each Transaction record is associated with exactly one Customer. This is represented by the relationship arrow from Transaction to Customer.

Key Considerations:

* Primary Keys: Ensure each table has a unique primary key (id for both Customer and Transaction).
* Foreign Key: Use foreign keys (customer\_id in Transaction) to establish the relationship between tables.
* Attributes: Include essential attributes that describe each entity (name, amount, etc.).
* One-to-Many Relationship: Use a collection (like a List in Java) in the Customer entity to maintain multiple Transaction records associated with each Customer.