Project Design Phase

Automation Design

Date	25 October 2025
Team ID	NM2025TMID02695
Project Name	Garage Management system
Maximum Mark	2 Marks

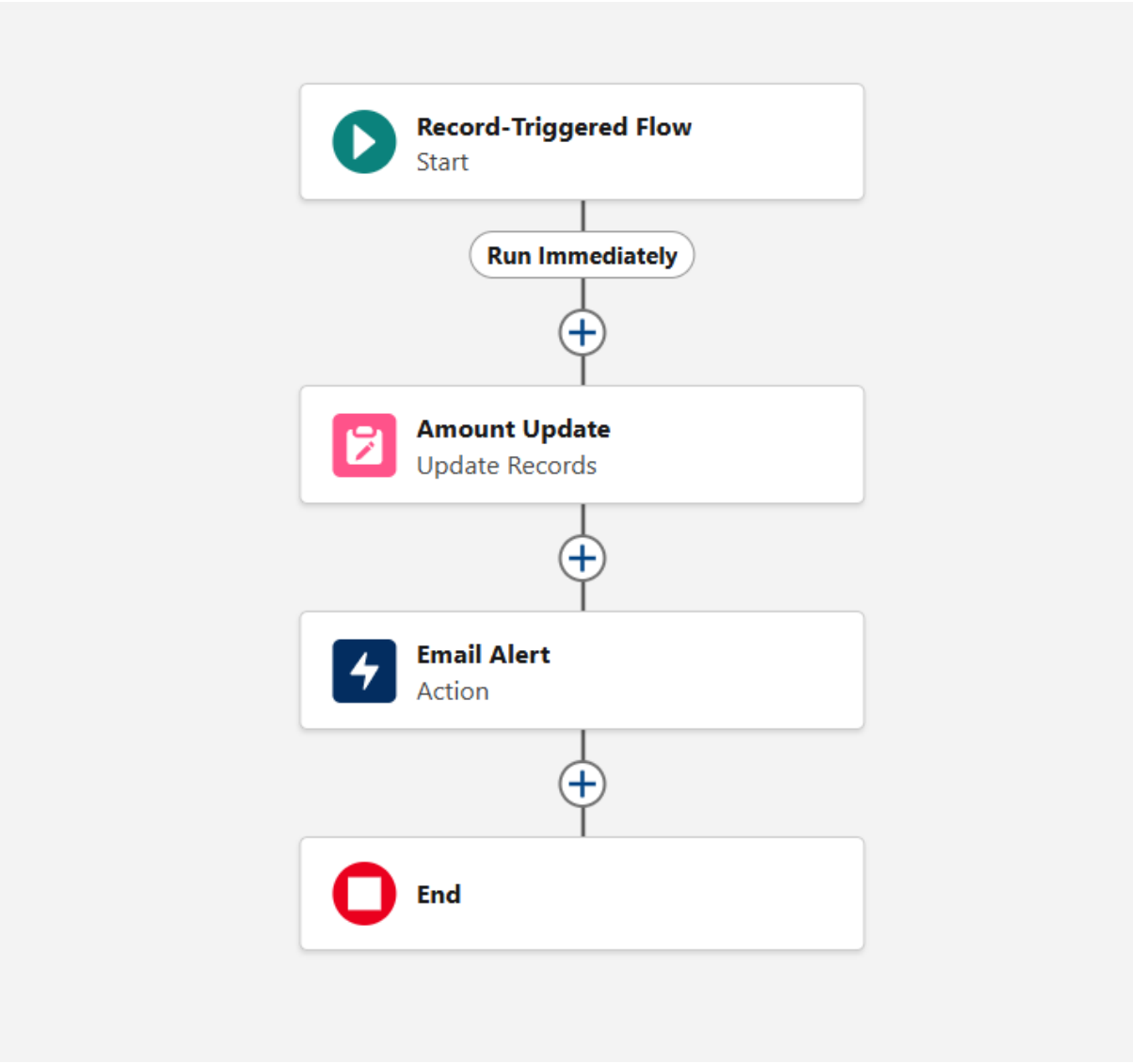
4.2 Automation Design

The GMS system utilizes **Salesforce Flows** and **Apex triggers** to automate repetitive tasks, maintain data synchronization, and enhance user productivity.

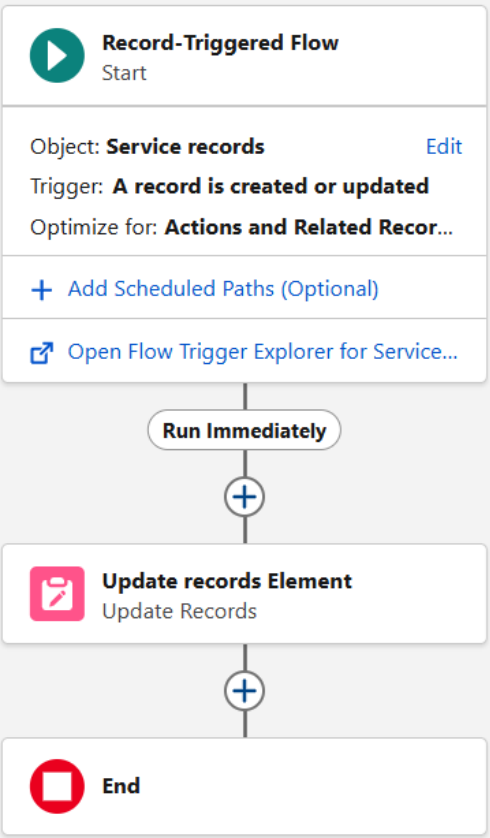
4.2.1 Record-Triggered Flow Design

Flow Name	Trigger Object	Trigger Event	Core Logic & Actions	Outcome
Update Service Status Flow	Service Records	On Create or Update	If `Quality_Check_Status__c = TRUE`, update `Service_Status__c = "Completed"`.	Automates closure of services upon QA success.
Amount Update & Email Alert Flow	Billing Details	On Create or Update	If `Payment_Status__c = "Completed"`, then: 1 Update parent Appointment's `Service_Amount__c` using Roll-Up/Apex. 2 Send a custom email alert to the Customer's `Gmail__c`.	Ensures financial updates and customer notifications are instantaneous.
Appointment Notification Flow *(optional)*	Appointment	On Create	Sends an email/SMS confirming appointment booking.	Enhances user experience with real-time communication.

Flow 1- Amount Update & Email Alert



Flow 2- update service record



Flow Best Practices Used:

- Used **Decision Elements** to branch logic based on status.
- Enabled **Fault Paths** to handle automation errors gracefully.
- Deployed flows as **"Before-Save" updates** where possible for performance optimization.

4.2.2 Programmatic Design (Apex Components)

While most automation is declarative, a lightweight Apex implementation handles specialized cost aggregation logic.

Component	Trigger Object	Purpose	Implementation Details
Apex Class: `AmountDistributionHandler`	N/A	Contains reusable methods to calculate the total cost across all Service	Uses SOQL aggregation queries to fetch and update the parent Appointment's `Service_Amount__c` field.

		Records under an Appointment.	
Apex Trigger: `AmountDistribution`	Appointment__c	Invokes the handler whenever a related Billing or Service record is created or updated.	Ensures real-time synchronization between Service Records and Appointment totals.

Apex Class: `AmountDistributionHandler`

```
public class AmountDistributionHandler {
    public static void amountDist(list<Appointment__c> listApp){
     list<Service_records__c> serList = new list <Service_records__c>();
      for(Appointment__c app : listApp){
        if(app.Maintenance_service__c == true && app.Repairs__c == true &&
app.Replacement_Parts__c == true){
            app.Service_Amount__c = 10000;
        }
        else if(app.Maintenance_service__c == true && app.Repairs__c == true){
            app.Service_Amount__c = 5000;
        }
        else if(app.Maintenance_service__c == true && app.Replacement_Parts__c == true){
            app.Service_Amount__c = 8000;
        }
        else if(app.Repairs__c == true && app.Replacement_Parts__c == true){
            app.Service_Amount__c = 7000;
        }
        else if(app.Maintenance_service__c == true){
            app.Service_Amount__c = 2000;
        }
        else if(app.Repairs__c == true){
            app.Service_Amount__c = 3000;
        }
```

```
        else if(app.Replacement_Parts__c == true){
            app.Service_Amount__c = 5000;
        }
    }
    }
}
```

Apex Trigger: `AmountDistribution`

Apex Trigger(AmountDistribution) code :

```
trigger AmountDistribution on Appointment__c (before insert, before update) {
    if(trigger.isbefore && trigger.isinsert || trigger.isupdate){
        AmountDistributionHandler.amountDist(trigger.new);
    }
}
```