High Level Design

Bus Ticketing System

Viral Shah

5/18/2020

Table of Contents

1 Document Control

This is a controlled document. All modifications require Version Number to be incremented (update Document Property 'Status') and detail of all modifications made in the Version History section below. All modifications must be approved and reviewed by nominated Approvers and Reviewers.

Version Number	Date Modified	Author	Modification Details
0.1	05-18-2020	Viral Shah	Initial Draft

2 Purpose of Document

This document provides a high-level solution design for the Bus Ticketing functionality

This document is not intended to provide a solution design for all requirements for the project and is limited to:

- As an end-user, I should be able to book tickets between 2 cities from a mobile app

- As an end-user, I should be able to select the seat and pay for the ticket

- As an end-user, I should see the auto-populated payment preference options

- As an admin, I should be able to create bus trips and see the list of customers

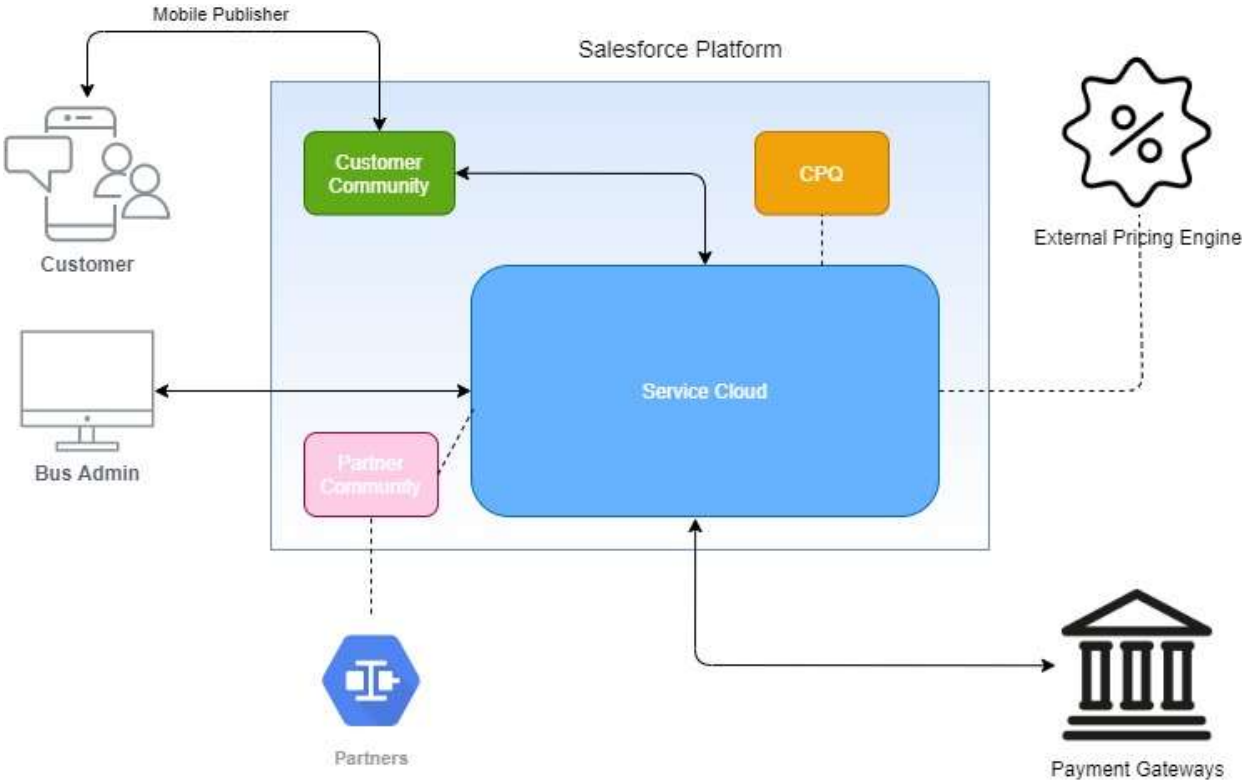
- As an admin, I should be able to cancel any bus trip

3 Technical Specifications

3.1 Business Context

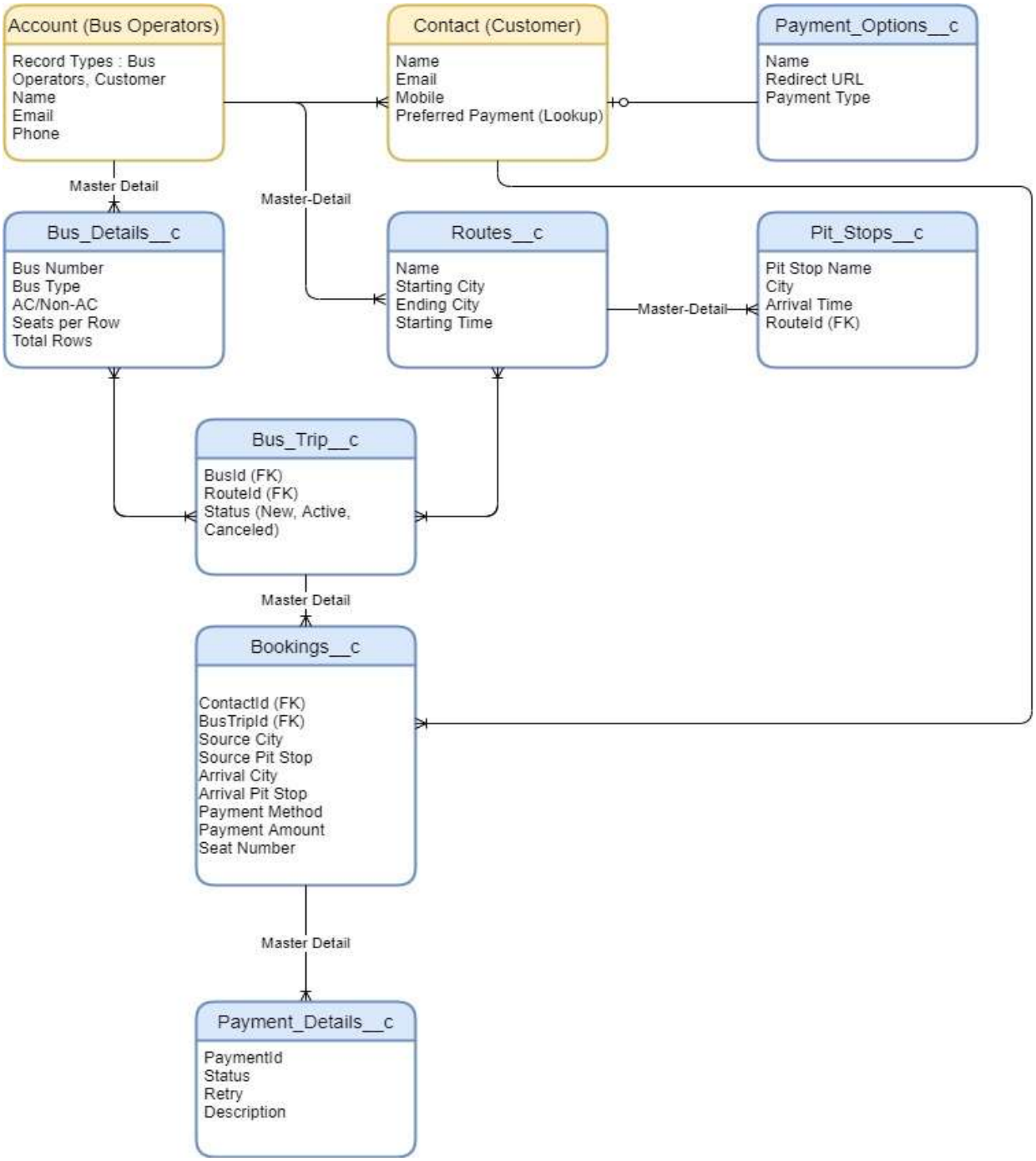
We are creating a Bus ticketing system which will cater to different audiences – Bus Admin & Customer. Bus admins responsibility includes setting up backend data needed to create a Bus trip. The customer should be able to book tickets via the mobile application and should be able to make payment for reserving a seat

3.2 High Level System diagram



Sr. No	Entity	Usage
1	Customer	The end user. Customer will install the Mobile Publisher Community app and login via that to book the bus tickets
2	Bus Administrators	They are Salesforce internal users with CRUD access to all Salesforce objects needed. They are responsible for creating Bus Trips and have options to cancel the trip
3	Service Cloud	This is the flavour of Salesforce environment to be used
4	Customer Community	Custom Salesforce community to be created via which Users can book the tickets
5	Payment Gateways	External third-party payment gateways outside of salesforce
6	CPQ	Optional. If the pricing of the Trips/Seats is complex, we can leverage Salesforce CPQ for our pricing requirements
7	Pricing Engine	Optional. Instead of Salesforce CPQ, we can also connect to an external Pricing engine
8	Partner	Optional. These are external Vendors to whom the buses may belong to
9	Partner Community	Optional. We can create a Partner community for the vendors to track the bus bookings, bus details and more

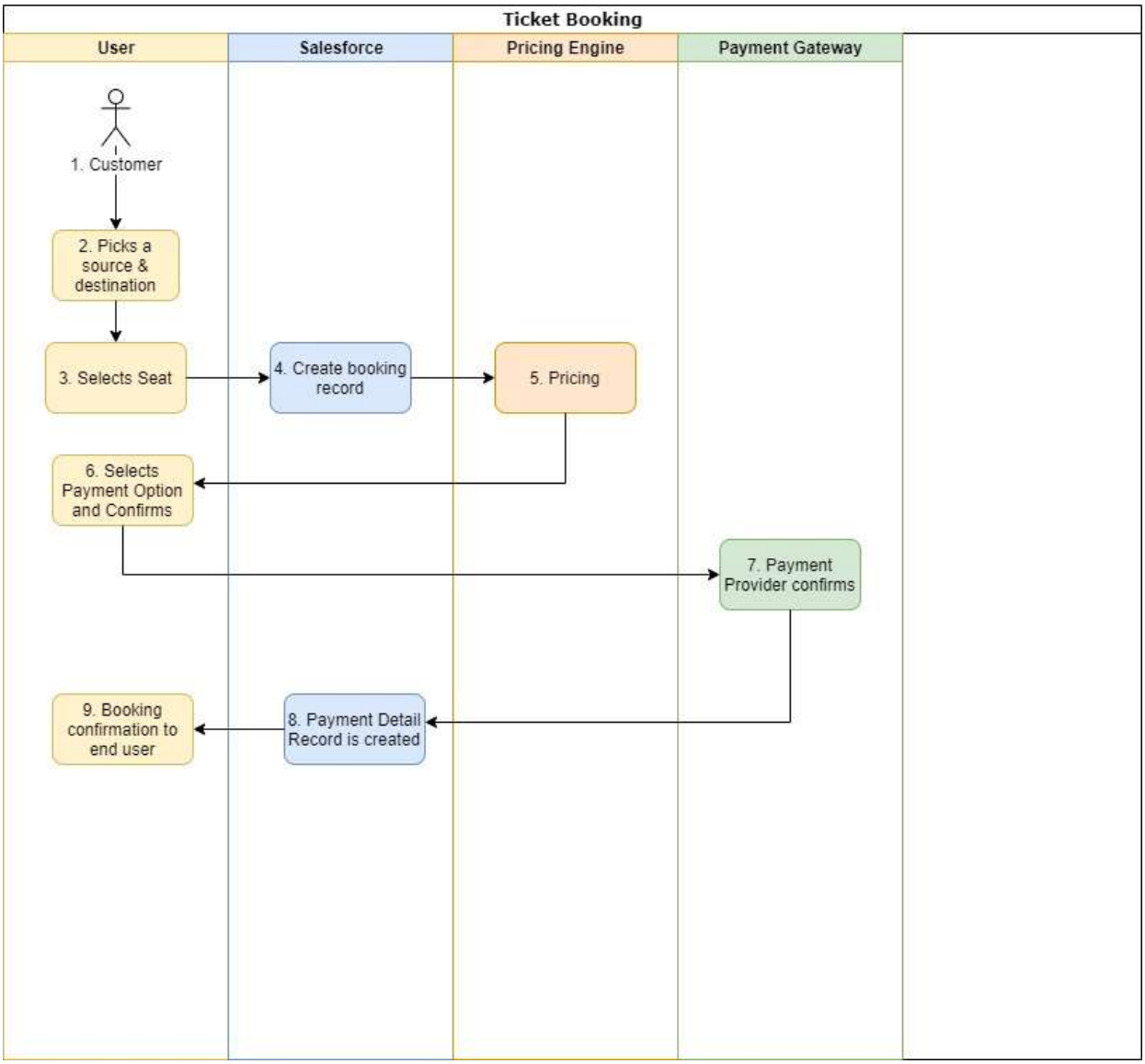
3.3 Object Model



Sr. No	Object	Description
1	Account	Account will be primarily used to store Vendor information. It will also contain the Accounts of each contacts
2	Contact	Contact will be used to store Customer Information. Contacts will be converted into Community Users
3	User	User object captures details of Internal and Community Users in Salesforce
4	Bus Detail	Bus detail will store important details about the bus – bus number, total number of seats, AC, type, etc. The same bus detail record shall be used for planning multiple trips on multiple routes
5	Route	Route will contain the source and destination cities. Each route shall also contain a start time. Multiple routes to be created for different starting times
6	Bus Trip	This is a junction object between Bus Detail and Route. A Bus Admin will create a trip choosing the bus and route. Every trip will contain a Status field to make it available for the user or cancel the trip
7	Booking	A booking record is created when a customer intends to book a seat for a specific bus trip. Booking details stores the source and destination stop of each customer
8	Payment Details	This is a child object to Booking. Payment details record is created when a customer submits for payment based on the payment method selected. It holds every individual transaction to the third-party Payment Gateway. In case, a payment errors out, a new payment detail record will be created for the new payment
9	Pit Stops	Child object of Route. This object contains the sequence of stops between each source and destination. A customer can choose to select the boarding and departure stop inside the cities
10	Payment Options	Stand-alone object to store multiple payment methods available and the corresponding URL. Can set a default payment option in case a user hasn't preferred one
11		

4 Use Cases

4.1 Bookings



A. Book tickets between 2 cities from a mobile app

- Customer will open the app to the Salesforce Community or he can open the Community URL in a browser
- Once logged in, customer can click on 'New booking' to book a seat
- They will need to input the Source and Destination City and click on search
- Salesforce will run the logic to fetch all the Trips with the chosen cities and display the list of Trips
- Optionally, the customer can also select the time of boarding to further limit the search

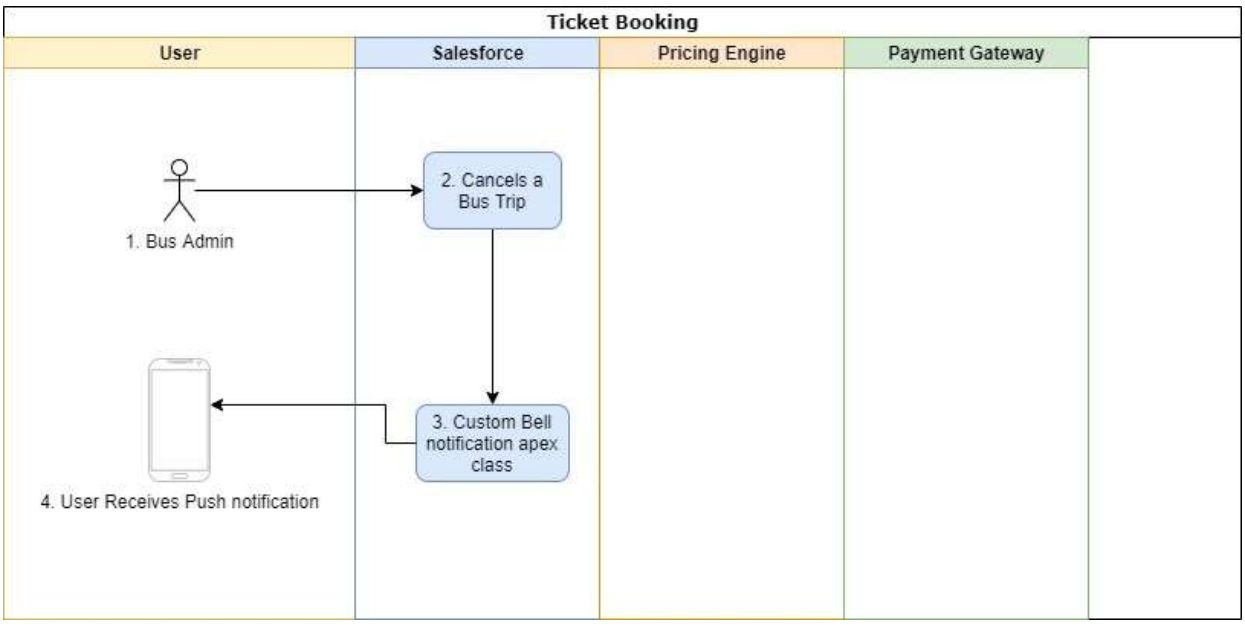
B. Customer should be able to select the seat and pay for the ticket

- Customer selects the Trip
- Salesforce logic returns the layout of the bus based on the Bus Details object and marks the seats which are already booked based on the Bookings table
- Customer is free to choose from the available seats and once the customer decides the seat, he clicks on the submit to proceed

C. Customer should see the auto-populated payment preference options

- Once submit is clicked, a booking record for the user is created in Salesforce
- Additionally, the fare for the ticket is calculated by sending a request to a Pricing engine
- The list of Payment methods (including the default or the preferred payment), final amount and the booking Id is sent back to the community page
- The customer now sees a list of payment methods – pre-selected, if applicable and should be able to proceed to the payment gateway
- Once payment confirmation is received to Salesforce, payment detail record is created for the booking in Salesforce
- User receives confirmation on the booking

4.2 Bus Admin



A. Bus Admins should be able to create bus trips and see the list of customers

- Bus Admins have the CRUD access with 'Modify All' on the set-up objects – Bus Details, Routes, Bus Trips. They would be able to Create, Update, Delete any Bus trips
- Bus Admins can check the bookings object to get the list of customers for each trip

B. Bus Admin should be able to cancel any bus trip

- Bus Trip object has a 'Status' field with values – New, Active, Cancelled
- Upon cancellation of a bus trip, you pass the list of 'Users' whose are had booked the seats for the trip to a custom Apex class
- The apex class does a REST call to insert a new Custom Notification record which sends a Push notification to the Customers on the mobile