Problem statement

Design & architect a system (backend) for leads prioritization on SquadVoice

Time:

75 mins = 60 mins for problem understanding, solutioning etc. + 15 mins for solution discussion

Description

SquadVoice is a B2B product. It enables sales teams to convert more leads.

Sales teams send their leads (live or stale) to SquadVoice. They use API endpoints to send leads or an admin interface to import leads via csv/xls files.

Lead attribute	Data type	Description	Sample value(s)
id	numeric	Lead identifier	1
customer_id	numeric	Customer identifier	55
campaign_id	numeric	Campaign identifier	4
lead_type	enum (string)	Type of lead	Stale or Live
city	string	City the lead belongs to	Delhi, Mumbai, Chennai,...
last_activity	timestamp	UTC timestamp of when the lead last did an activity	
stage	string	Stage in which the lead is	
pricing_plan	enum (string)	Pricing plan the customer belongs to	Starter, Pro or Enterprise
rescheduled_at	timestamp	UTC timestamp of when the lead was last rescheduled	
created_at	timestamp	UTC timestamp of when the lead was created	

There is an SLA guarantee of upto 30 mins for all live lead with customers - 90%.

Each customer can have multiple lead qualification campaigns. The leads can be called by any contractor in the workforce that is qualified for a campaign. Currently the live leads are assigned to the contractors on a FIFO basis. Any live lead that doesn't pick up a call is rescheduled to be called 22-24 hrs later (upto 3 reschedules). Any live lead that didn't pick up a call becomes a stale lead.

Our product team has recently run some conversion optimization experiments. They observed that leads prioritized in a certain fashion will yield higher conversion rates. They made the following observations

1. The faster the Turn-Around-Time (TAT) to serve a live lead the higher the conversion
2. For all live leads of a customer with similar TAT the conversion rates were strongly correlated with city, source, last_activity & stage of the lead. This though varies by each customer.
3. The more recent the last_activity of a stale lead the higher the conversion

Based on the above observations the product team will like to implement a prioritization system. The **prioritization engine** will interact with two existing systems as noted below;

1. **Lead Data store**- It stores the leads as the above noted schema
2. **Contract app API Engine** is what the Android app interacts with. It is responsible for assigning the next lead to the agent/contractor.
3. **Lead Data store** is the current single source of truth for the leads metadata.
4. **Contract app API Engine** will send the **prioritization engine** a tuple of campaign_ids that the contractor is qualified for & the **prioritization engine** will send the best lead_id based on the prioritization logic.
5. **Contract app API Engine** will update **Lead Data store** when a lead is rescheduled.
6. **Contract app API Engine** automatically propagates rescheduled_at & lead_type updates to **Lead Data store** when a lead is rescheduled.

Evaluation criteria

a. Tech Design skills - How do you keep your design simple yet effective? How extensible/modular is your design?
b. Problem solving - How effective are you at understanding the problem & designing a solution for it?
c. Communication - How well do you communicate, resolve gaps, highlight your assumptions etc.?
d. Domain expertise - How do you leverage your domain knowledge & expertise in thinking through a solution? What depth of finer details are you able to think of?