Ride Request MVP: Full Service Provider Flow

FE Dash | FE Assign | API | DB | DispatchBot | Twilio

Load all ready and active ride requests into the Front End Dashboard

FE Dash → API: ReadRideRequests()

API → DB: GetRideRequests()

DB: SELECT FROM RIDE_REQUESTS WHERE RIDE_REQUEST_STATUS IN (123, 456) AND dropoff_time BETWEEN THEN() AND NOW()

API ⇠ DB

API ⇠ FE Dash: Display rides that are ready or active

FE Dash → FE Assign: Pick a ready ride request and assign driver and vehicle

Autocomplete search dialog to find available drivers

FE Assign → API: AutoSearch(SearchText, "DRIVERS")

API → DB: SearchTable(SearchText, "DRIVERS")

DB: SELECT FROM DRIVERS WHERE LIKE SearchText

API ⇠ DB

FE Assign ⇠ API: Search Results

Autocomplete search dialog to find available vehicles

FE Assign → API: AutoSearch(SearchText, "VEHICLES")

API → DB: SearchTable(SearchText, "VEHICLES")

DB: SELECT FROM VEHICLES WHERE LIKE

SearchText

Search Results

Ride Request is
assigned and can
begin on schedule

Process of starting ride with automation and notifications

Dispatcher selects a ride to start
InitiateRide(Ride Request UUID)

StartRide(Ride Request UUID)

ReadRideRequest(Ride Request UUID)

**DispatchBot to confirm ride request status,
ride status**: Return info for SMS

SendSMS(Phone
nums, message)

SMS messages ack

Confirm SMS: ChangeRideStatus(Ride Status,
Ride Request UUID)

UpdateRideStatus(Ride
Status, Ride Request
UUID)

Status updated

SMS confirmed

Ride Started

Ride initiated

Dashboard checks for status updates

Status updated

DispatchBot listens for incoming SMS replies when messages require confirmation

Listening

SMS confirm received

Confirm SMS: ChangeRideStatus(Ride Status,
Ride Request UUID)

UpdateRideStatus(Ride Status, Ride Request UUID)

Status updated

SMS confirmed

SendSMS(Phone nums, message)

SMS messages ack

Dashboard checks for status updates

Status updated

FE Dash

FE Assign

API

DB

DispatchBot

Twilio