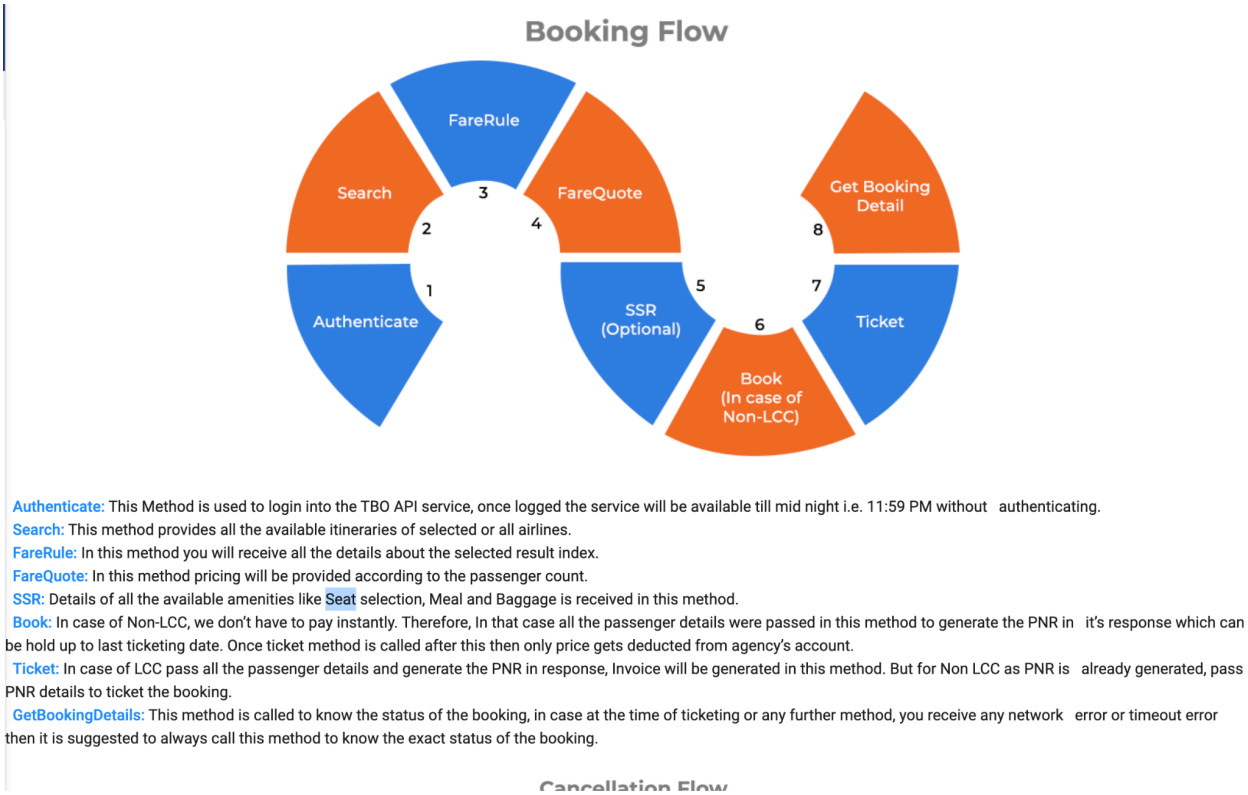
Figma Design Link:

https://www.figma.com/design/AsQzOlGbYMZcrie9jJ0wBG/Routechef?node-id=15-11217&t=HqUIAl8v3lvgXPiu-1



User clicks **Book Flight** on Routechef chat → itinerary/review booking page opens (has route details/stops/fareRules/fareQuote) → User clicks payment button → book api called (to be implemented from my side) → payment gateway opens →ticket api (to be implemented) is called and ticket confirmation page is shown

Fare Rules, Fare Quote, ticket-lcc

https://s4nvnzwoc8.execute-api.ap-south-1.amazonaws.com/default/tbo_api_orchesterator

Sample request body for GET call:


```
{
 "intent": "fareQuote/fareRules",
 "TraceId": "14f22093-edb5-4f19-b8b0-4e12de026295",
 "ResultIndex":
"OB21[TBO]IO0m32uBd9jAMCTwIhl2mJ54ELvXM+jndlOtET8dHgp3bKBAhQNVSSA81TlcVebFVi8TAlx+K/LD
dExdkzKL312KyHFRJ+pIm3O2luXMTcDpUpehj3PuuykZKX0KT95Cf+gozD3zqEzSXbf5ZXhSeDLmWD1futhb69
```

```
ynooCsKrpV3daoMd8V2MPb2z5UFOidwJp3F3HTNg+bio4fQ5fNjfL2w2bYuiqQ1Dr5EF9h5aGfhrlauM9bIw1s
x0/5v97k8ceFM7tP8xvNO2LCkM12JWQui5BOqaaRNpLT38GuAzUlmil0hT0TA1uqiCfOQ/zY6loBvM6MAck0mc
J1/LPoEs8PyKQfh67J0rWjv77nCmRK1FeYViwSUiJs5kns4fxlUTa8S4/2gyOHF0AYyql6o3l6xYt8oE4qn+5G
42b/oqgaGZIZY4P5l7lcCr2MZi9OPhkCdLx6TRuDzRQ5K71p871LzfaNrDD65CDDSSFGzKcmZy1GgN6CzmRVbm
D5aCyv45ONbXj+5VJ48Fq4O/Pfrdm2lw2Ud2G3kOSaiTzRGZ23OI8skV53h5dz1FlsIK7KijHr9qOw14t7bcT9
HvWcqZk76mLhKym5pYiv6M2d2kZ34DQpowgBDmJui5pdGWXboMkn2mWjC/khC3ZQoXCX99knqosUEw/ZpKRVFm
oZ1pGk6ZXbNE0xXZRJEcAJxQfp",
 "EndUserIp": "125.63.76.234",
 "From": "webbot_new:anon",
 "headers": {
   "Authorization": "Bearer dummy-token"
 }
}
```


Note: TraceId and ResultIndex to be picked from the flight segment in result. `EndUserIp` **is same across apis.**

`headers->Authorization->is the firebase authentication (google or phone login auth that we passs along all messages)`

SAMPLE RESPONSE 'BODY' FOR INTEGRATIONS

Fare Rules

[fareRules.json]

Fare Quote

[fareQuote.json]

```python
import json, zlib, base64
import binascii
```

```python
def defaultencode(o):
   if isinstance(o, Decimal):
       # Subclass float with custom repr?
       return fakefloat(o)
   raise TypeError(repr(o) + " is not JSON serializable")
```

```python
# 🔐 Compress & encode
```

```python
def compress_list_of_dicts(data_list):
    json_str = json.dumps(data_list, default=defaultencode)
    compressed = zlib.compress(json_str.encode('utf-8'), level=9)
    b64_encoded = base64.b64encode(compressed).decode('utf-8')
    return b64_encoded

# 🔓 Decode & decompress with base64 check
def decompress_list_of_dicts(b64_string):
    try:
        # Validate base64 string
        base64.b64decode(b64_string, validate=True)

        # Proceed if valid
        compressed = base64.b64decode(b64_string)
        json_str = zlib.decompress(compressed).decode('utf-8')
        return json.loads(json_str)
    except Exception as e:
        # print("⚠️ Invalid input or format:", e)
        return b64_string # to handle non b64 strings /lsits as well
```

Sample request for intent “ticket-lcc”


```
{
    "intent": "ticket-lcc",
    "EndUserIp": "125.63.76.234",
    "TraceId": "da252a82-1024-4a5b-906b-8a3bfaa499ed",
    "ResultIndex":
"OB1[TBO]DXQhmNRpF0h+3uY/r8TgFKAaIIo8u2M6y0O8g3ELplI18f0VyvgpqwkN8V0e0O9etKkz769T/fndp
sacPlUpYB/WqblWxfGYO93xENC+3JFcIRHiApaSvfKviMBqu8mFrKm4Th+izJBs27ivSmTG8xj5PAMY4htFTCi
jJeDeK2p16FnnLI230oD1cjXlRZ7Z3T7YVhv+BQQSq9hhjM/dw0hzl4md3hOi9xpL4el42PUv+QxrPzXYWoc+v
Bn/+MgSKR6NrjVVXtNB7zf0DcjW8cwIwgBTUenPJ15q61jqqXsSVhw2LrTH4/zzGV2dOen7v3DOO2slWk3YQTv
N3GZzpiJx8x1/vXFxU4Hw698L7/c9Wcdnc7giWmwM/5m3jy4oJfAUYvZzheTXC71z/yOdRaZM3totg2dmwfaUT
cFg6PFdXJSgk87QHKWVe6xjblTo/W9tX6rwEvFaLy0tMH/8ngaYqN7UV6Zj8IV2WkAEW59K5h6L3xsGyWG+byn
FNrhgRoJqGrbG0AGMUHYh40WRQG4hpeyHrTdmnUIlWaTDdIJek6qyQDx5FejHNkWjeCltV7NwskaoIi25gR1Vg
Jm9Ui6j5WyQKqTIjehBQwBWCY46nIwgU9LNCqtuptNxuoJUtXcdJ/qRqIygePhWYiFsBw==",
    "currency": null,
    "Passengers": [
        {
            "Title": "Mr",
            "FirstName": "Rakesh",
            "LastName": "Sharma",
```

```
            "PaxType": 1,
            "DateOfBirth": "1987-12-06T00:00:00",
            "Gender": 1,
            "AddressLine1": "123, Test",
            "AddressLine2": "",
            "Fare": {
                "BaseFare": 4972,
                "Tax": 1291,
                "YQTax": 0.0,
                "AdditionalTxnFeePub": 0.0,
                "AdditionalTxnFeeOfrd": 0.0,
                "OtherCharges": 0.0
            },
            "City": "Gurgaon",
            "CountryCode": "IN",
            "CountryName": "India",
            "Nationality": "IN",
            "ContactNo": "9879879877",
            "Email": "harsh@tbtq.in",
            "IsLeadPax": true,
            "FFAirlineCode": null,
            "FFNumber": null,
            "Baggage": [
            ],
            "MealDynamic": [
            ],
            "SeatDynamic": [
            ],
            "SpecialServices": [
            ],
            "GSTCompanyAddress": "",
            "GSTCompanyContactNumber": "",
            "GSTCompanyName": "",
            "GSTNumber": "",
            "GSTCompanyEmail": ""
        }
    ]
}
```

Sample request for intent “book-non-lcc”


```
{   "intent": "book-non-lcc",
   "EndUserIp": "125.63.76.234",
   "TraceId": "4d46c5fd-4bd1-4adf-8513-4eae7557683e",
 "ResultIndex":
"OB1[TBO]Q6avX5hY3ZgtzFXtkgjPVbeP+z+7z8HgKnZaYTp5P3X8LB0txsAms6YVxz4rYu1qrwbUv/rjYFwg7
PBJSyOGPjAbQnsZ5o1iB6am5w4hGqTC1NpNp/9ISzSDnDs0V83nO66zD+UPcYOlhVvts28PfN1+iDEs84H72Pi
2KD9xlMdws/U3rI5EYHOR6fN7Pya6UBqDmEJHZddUrcIX/PfGFy7s9i3S+4KVJQKrvFlsYG765fMVn/aTYy+x3
1xaEhpxEoiUHebPEr0pfJXnrtXPK+auT0aEkG7NnrNyIQ466Vk5B06mZ23zpmCzKbxO4FIjijb+xvk7ouwsTjC
b5ZvmEv9cOyVaOmPEuYA43iQ2SFWteu87rClK79Ru/CT1zwHxrxKK1KyCI3J4RoOHGx/W8K9N3eFIyAAyh7x6h
uSA+sEYAX+67oRxmFV2cxgJq58i3kL1ZTrlutxocAODNVv7PKUT5/Dic5b9Ifezq/NJfYWNUm+sUO4M9LTeMW/
nsCNdTDyoF3ZG7+plXVaDU0d7xc1ctO0/B8MTZeQD69tYo3ymo0GmeM0NlpBwMXfUOa9bYHNgM/wsrt231bqe5
OlS3bO2cLEUCs0BW62/gwUGDF3Ru2/Ani4wh+CTAwCzVRPqWZ0bM9V5joFn3EtP6+u/JavDNwtrZ8He1BIAoJ7
QhLwgqbb/YoNpgmK0/EyqYFLB7kZM20EF6HaHb9QzRYH3GF3zSi5X6zRzQvZfyVWrncZ5T4rDwYlyqlLOO2wDI
DHH07L2N8Vc7W8nDye+B1K6bA==",
   "Passengers": [{
       "Title": "Mr",
       "FirstName": "hgjsshsxsgjh",
       "LastName": "tbotest",
       "PaxType": 1,
       "DateOfBirth": "1987-12-06T00:00:00",
       "Gender": 1,
       "PassportNo": "KJHHJKHKJH",
       "PassportExpiry": "2025-12-11T00:00:00",
       "AddressLine1": "123, Test",
       "AddressLine2": "",
       "Fare": {
           "Currency": "INR",
           "BaseFare": 4972,
           "Tax": 1291,
           "YQTax": 0.0,
           "AdditionalTxnFeePub": 0.0,
           "AdditionalTxnFeeOfrd": 0.0,
           "OtherCharges": 0.0,
           "Discount": 0.0,
           "PublishedFare": 6263,
           "OfferedFare": 6112.71,
           "TdsOnCommission": 36.69,
           "TdsOnPLB": 31.49,
           "TdsOnIncentive": 32.02,
           "ServiceFee": 0.0
```

```
        },
        "City": "Gurgaon",
        "CountryCode": "IN",
        "CellCountryCode" : "+92581-",
                "ContactNo": "1234567890",
        "Nationality": "IN",
                "Email": "harsh@tbtq.in",
        "IsLeadPax": true,
        "FFAirlineCode": null,
        "FFNumber": "",
        "GSTCompanyAddress": "",
        "GSTCompanyContactNumber": "",
        "GSTCompanyName": "",
        "GSTNumber": "",
        "GSTCompanyEmail": ""
    }
    ]

}
```

Sample request for intent “ticket-non-lcc”

```
{   "intent": "ticket-non-lcc",
    "EndUserIp": "125.63.76.234",
    "TraceId": "4d46c5fd-4bd1-4adf-8513-4eae7557683e",
"Passport": [
 {
    "PaxId": 2040529,  # to be pulled from book-non-lcc response
    "PassportNo": "abcx12345",
    "PassportExpiry": "2030-03-03T00:00:00",
    "DateOfBirth": "1998-02-02T00:00:00"
 },
 {
    "PaxId": 2040530, # to be pulled from book-non-lcc response
    "PassportNo": "abcx54321",
    "PassportExpiry": "2030-03-05T00:00:00",
    "DateOfBirth": "1999-02-02T00:00:00"
 }
],
 "PNR": "GU2NFG", # to be pulled from book-non-lcc response
```

```
 "BookingId": 2027438, # to be pulled from book-non-lcc response
 "From": "web:anon",
 "headers": {
   "Authorization": "Bearer dummy-token"
 }
}
```