```
# from datetime import datetime, timedelta
# from data_manager import DataManager
# from flight_search import FlightSearch
# from notification_manager import
NotificationManager
#
#
# ORIGIN_CITY_IATA = "LON"
#
# data_manager = DataManager()
# flight_search = FlightSearch()
# notification_manager = NotificationManager()
#
# sheet_data = data_manager.get_destination_data()
#
# if sheet_data[0]["iataCode"] == "":
#     city_names = [row["city"] for row in sheet_data
]
#     data_manager.city_codes = flight_search.
get_destination_codes(city_names)
#     data_manager.update_destination_codes()
#     sheet_data = data_manager.get_destination_data
()
#
# destinations = {
#     data["iataCode"]: {
#         "id": data["id"],
#         "city": data["city"],
#         "price": data["lowestPrice"]
#     } for data in sheet_data}
#
# tomorrow = datetime.now() + timedelta(days=1)
# six_month_from_today = datetime.now() + timedelta(
days=6 * 30)
#
# for destination_code in destinations:
#     flight = flight_search.check_flights(
#         ORIGIN_CITY_IATA,
#         destination_code,
#         from_time=tomorrow,
#         to_time=six_month_from_today
```

```python
#     )
#     print(flight.price)
#     if flight is None:
#         continue
#
#     if flight.price < destinations[destination_code]["price"]:
#
#         users = data_manager.get_customer_emails()
#         emails = [row["email"] for row in users]
#         names = [row["firstName"] for row in users]
#
#         message = f"Low price alert! Only £{flight.price} to fly from {flight.origin_city}-{flight.origin_airport} to {flight.destination_city}-{flight.destination_airport}, from {flight.out_date} to {flight.return_date}."
#
#         if flight.stop_overs > 0:
#             message += f"\nFlight has {flight.stop_overs} stop over, via {flight.via_city}."
#
#         link = f"https://www.google.co.uk/flights?hl=en#flt={flight.origin_airport}.{flight.destination_airport}.{flight.out_date}*{flight.destination_airport}.{flight.origin_airport}.{flight.return_date}"
#
#         notification_manager.send_emails(emails, message, link)
#
#

print("Welcome to Lenar's Flight Club.\nWe find the best flight deals and email you.")
input("What is your first name?\n")
input("What is your last name?\n")
input("What is your email?\n")
input("Type your email again.\n")
print("You're in the club!")

```

```
#notification manager
import smtplib
from twilio.rest import Client

TWILIO_SID = YOUR TWILIO ACCOUNT SID
TWILIO_AUTH_TOKEN = YOUR TWILIO AUTH TOKEN
TWILIO_VIRTUAL_NUMBER = YOUR TWILIO VIRTUAL NUMBER
TWILIO_VERIFIED_NUMBER = YOUR TWILIO VERIFIED NUMBER
MAIL_PROVIDER_SMTP_ADDRESS = YOUR EMAIL PROVIDER
SMTP ADDRESS "smtp.gmail.com"
MY_EMAIL = YOUR EMAIL
MY_PASSWORD = YOUR PASSWORD

class NotificationManager:

    def __init__(self):
        self.client = Client(TWILIO_SID,
TWILIO_AUTH_TOKEN)

    def send_sms(self, message):
        message = self.client.messages.create(
            body=message,
            from_=TWILIO_VIRTUAL_NUMBER,
            to=TWILIO_VERIFIED_NUMBER,
        )
        print(message.sid)

    def send_emails(self, emails, message,
google_flight_link):
        with smtplib.SMTP(MAIL_PROVIDER_SMTP_ADDRESS
) as connection:
            connection.starttls()
            connection.login(MY_EMAIL, MY_PASSWORD)
            for email in emails:
                connection.sendmail(
                    from_addr=MY_EMAIL,
                    to_addrs=email,
                    msg=f"Subject:New Low Price
Flight!\n\n{message}\n{google_flight_link}".encode('
utf-8')
                )
```

```

#data manager
from pprint import pprint
import requests

SHEETY_PRICES_ENDPOINT = YOUR SHEETY PRICES ENDPOINT
SHEETY_USERS_ENDPOINT = YOUR SHEETY USERS ENDPOINT

class DataManager:

    def __init__(self):
        self.destination_data = {}

    def get_destination_data(self):
        response = requests.get(url=SHEETY_PRICES_ENDPOINT)
        data = response.json()
        self.destination_data = data["prices"]
        return self.destination_data

    def update_destination_codes(self):
        for city in self.destination_data:
            new_data = {
                "price": {
                    "iataCode": city["iataCode"]
                }
            }
            response = requests.put(
                url=f"{SHEETY_PRICES_ENDPOINT}/{city['id']}",
                json=new_data
            )
            print(response.text)

    def get_customer_emails(self):
        customers_endpoint = SHEETY_USERS_ENDPOINT
        response = requests.get(url=customers_endpoint)
        data = response.json()
        self.customer_data = data["users"]
        return self.customer_data
```

```python
    #flight data
    class FlightData:

        def __init__(
                self, price, origin_city, origin_airport, destination_city, destination_airport, out_date, return_date,
                stop_overs=0, via_city=""):
            self.price = price
            self.origin_city = origin_city
            self.origin_airport = origin_airport
            self.destination_city = destination_city
            self.destination_airport = destination_airport
            self.out_date = out_date
            self.return_date = return_date
            self.stop_overs = stop_overs
            self.via_city = via_city

#flight search
import requests
from flight_data import FlightData

TEQUILA_ENDPOINT = "https://tequila-api.kiwi.com"
TEQUILA_API_KEY = YOUR FLIGHT SEARCH API KEY


class FlightSearch:

    def get_destination_code(self, city_name):
        location_endpoint = f"{TEQUILA_ENDPOINT}/locations/query"
        headers = {"apikey": TEQUILA_API_KEY}
        query = {"term": city_name, "location_types": "city"}
        response = requests.get(url=location_endpoint, headers=headers, params=query)
        results = response.json()["locations"]
        code = results[0]["code"]
        return code

    def check_flights(self, origin_city_code, destination_city_code, from_time, to_time):
        headers = {"apikey": TEQUILA_API_KEY}
        query = {
            "fly_from": origin_city_code,
            "fly_to": destination_city_code,
            "date_from": from_time.strftime("%d/%m/%Y"),
            "date_to": to_time.strftime("%d/%m/%Y"),
            "nights_in_dst_from": 7,
            "nights_in_dst_to": 28,
            "flight_type": "round",
            "one_for_city": 1,
            "max_stopovers": 0,
            "curr": "GBP"
        }

        response = requests.get(
            url=f"{TEQUILA_ENDPOINT}/v2/search",
            headers=headers,
            params=query,
        )
        try:
            data = response.json()["data"][0]
        except IndexError:
            query["max_stopovers"] = 1
            response = requests.get(
                url=f"{TEQUILA_ENDPOINT}/v2/search",
                headers=headers,
                params=query,
            )
            data = response.json()["data"][0]
            pprint(data)
            flight_data = FlightData(
                price=data["price"],
                origin_city=data["route"][0]["cityFrom"],
                origin_airport=data["route"][0]["flyFrom"],
                destination_city=data["route"][1]["cityTo"],
                destination_airport=data["route"][1]["flyTo"],
                out_date=data["route"][0]["local_departure"].split("T")[0],
                return_date=data["route"][2]["local_departure"].split("T")[0],
                stop_overs=1,
                via_city=data["route"][0]["cityTo"]
            )
            return flight_data
        else:
            flight_data = FlightData(
                price=data["price"],
                origin_city=data["route"][0]["cityFrom"],
                origin_airport=data["route"][0]["flyFrom"],
                destination_city=data["route"][0]["cityTo"],
                destination_airport=data["route"][0]["flyTo"],
                out_date=data["route"][0]["local_departure"].split("T")[0],
                return_date=data["route"][1]["local_departure"].split("T")[0]
            )

            return flight_data


```