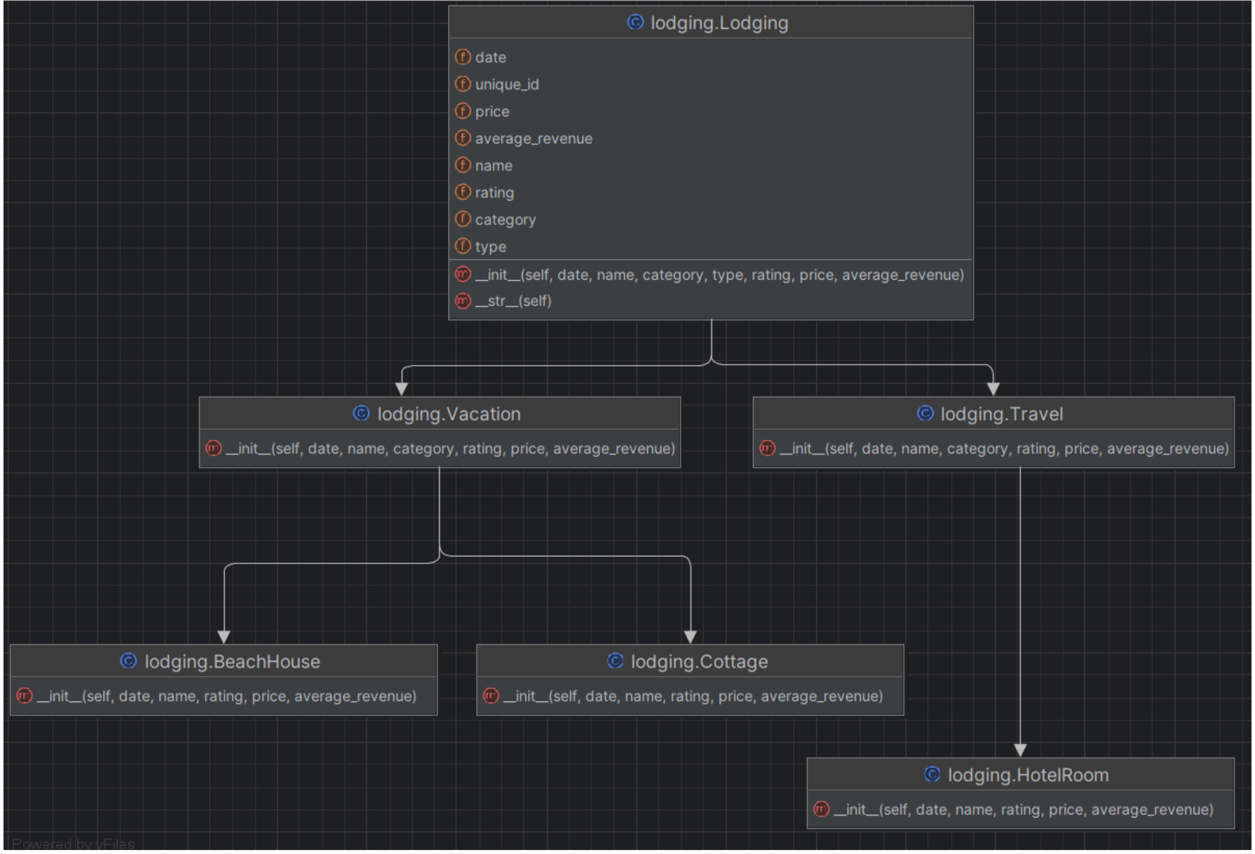
**Appendix:**
**Follow the instructions carefully to make sure the dataset can be loaded properly:**

Scenario #9: Data about Lodging rentals offered by a travel agency website.

1. Create a class called Lodging with the following attributes in its constructor (__init__ method) **in this order**:
    a. date: Date on which lodging place is added to the website.
    b. name: Name of the lodging's rental place.
    c. category: Category of lodging.
    d. type: Type of lodging.
    e. rating: Customers's rating for the accommodation.
    f. price: Price of stay per night (in USD).
    g. average_revenue: Average revenue of the rental (in USD).
2. In the constructor, add another attribute **unique_id** (Unique Identifier) and set the attribute's value to **id(self)** – the memory id of the instance of the class.
3. Implement the __str__ method within the Lodging class to return a formatted string containing the **values** of all the attributes in this format: "unique_id,date,name,category,type,rating,price,average_revenue"(no spaces)
4. Create two subclasses: Travel and Vacation, both inheriting from the Lodging class
5. The constructors in the Travel and Vacation classes should:
    a. Take input **in this order**: date,name,category,rating,price,average_revenue
    b. Use super() to call the constructor of the parent Lodging class, passing the type attribute as "Travel" for Travel and "Vacation " for Vacation with rest of the input.
6. Create three additional subclasses: HotelRoom, Cottage, BeachHouse inheriting from Travel and Vacation, Vacation respectively.
7. The constructors in the HotelRoom, Cottage and BeachHouse classes should:
    a. Take input **in this order**: date,name,rating,price,average_revenue
    b. Set the category as class name (i.e. category = "Vacation" if class is Vacation).
    c. Use super() to call the constructor of the parent class with the rest of the input.
8. Once the classes are ready, test your code with the examples shown below:
    a. VacationLodging = Vacation("2022-10-08","Urban Retreat", 1.0, 85.32, 69020.0)
    b. print(str(VacationLodging))
    c. should return "**[unique_id],2022-10-08, Urban Retreat,Cottage,Vacation,1.0,85.32,69020.0**"
9. Once that is verified, load the file provided in this link.
10. Create a csv file with the steps below:
    a. Write a first line that denotes the column headers as "unique_id,date,name,category,type,rating,price,average_revenue"(no spaces)
    b. Loop through all the objects retrieved from the above pickle file and use str method to print formatted string as mentioned in step 3. Refer to snippet of code provided below.
11. Use the above generated csv file to create visualizations in Python.

**Class diagram/ hierarchy should look like below:**


lodging.Lodging
date
unique_id
price
average_revenue
name
rating
category
type
__init__(self, date, name, category, type, rating, price, average_revenue)
__str__(self)
lodging.Vacation
__init__(self, date, name, category, rating, price, average_revenue)
lodging.Travel
__init__(self, date, name, category, rating, price, average_revenue)
lodging.BeachHouse
__init__(self, date, name, rating, price, average_revenue)
lodging.Cottage
__init__(self, date, name, rating, price, average_revenue)
lodging.HotelRoom
__init__(self, date, name, rating, price, average_revenue)
Powered by yFiles

**Snippet for step 10:**

```
with open('data.csv', 'w') as f:
    f.write("unique_id,date,name,category,type,rating,price,average_revenue\n")
    for obj in objects:
        f.write(str(obj)+'\n')
```