

Design and implement a real estate listing management application using an object-oriented approach.

The application should cater to a real estate company's needs for managing their residential listings currently stored on paper forms.

Requirements:

- Data Management:
 - Capture and store details like owner name (first & last), address, price, square footage, and number of bedrooms for each property.
 - Organize data efficiently to handle multiple listings per owner (using combined first and last name as a key).
 - Store buyer details (name, phone number, email).
 - Associate interested buyers with specific properties.
- Functionality
 - Add a new property listing.
 - Update existing listing information.
 - Delete a property listing.
 - Search and display details of a specific property based on the owner's name.
 - Generate an alphabetical list of all property owners.
 - Retrieve and display information for a specific buyer, including their list of interested properties.
 - Display the list of potential buyers interested in a specific property.
- Storage:
 - Store property data in a CSV file with appropriate delimiters for easy import/export.
 - Store Owner data in a CSV file with appropriate delimiters for easy import/export.
 - Store Buyer data in a CSV file with appropriate delimiters for easy import/export.
- Design Principles:
 - Employ object-oriented principles like encapsulation and abstraction to achieve code reusability, flexibility, and maintainability.
- Diagrams:
 - Create a UML class diagram using the draw.io tool to illustrate the classes and their relationships.
 - Develop a user case diagram using draw.io to depict the interactions between user(s) and the application functionalities.
- Architecture:
 - Implement a 3-layer architecture consisting of:
 - Presentation Layer: Develop a menu-driven user interface (console-based) for interacting with the application.
 - Business Layer: This layer handles the core functionalities like managing owner, buyer, and property objects, performing business logic (e.g., validating inputs), and enforcing business rules.
 - Data Layer: This layer manages reading and writing the data to and from the CSV file.