Lab 02	Design and Implement Business Logic
IT618 Enterprise Computing, Autumn'2023; pm_jat @ daiict	

The objective of this lab is to design and implement Business Logic for a given small application.

Consider creating **visitor tracking application for a residential society**.

Suppose there are a finite set of houses in the society. Let us say we record House No (unique), Contact Name, and Contact Number for each house.

When a visitor visits to the society, security guard collects following details from the visitor: Visitor Name, Contact Number, and House No to which the visitor wants to visit. All visits require approval from the visiting house contact. But for the sake of flexibility, we do allow visitor in case of non-response from the house. Visitor id is given to every visitor. Let application record all stated details of a visitor. Also, let timestamps of entry time and exit time also be recorded.

For simplicity, let us exclude family guests, who can stay for longer duration. Let us also exclude visitors that come for society common services like housekeeping, and so on regular basis. Let the kind of visitors that are in scope right now are those who visit either for some delivery or for household services to individual houses.

When visitor leaves the society, gives visitor id to the security guard and visitor exit entry is made into the application.

Considering this scenario, let following use cases to be in scope of the application.

1. **Visitor enters the society**.
    a. Visitor details are collected through application (don't worry about what is mode of data collection or so, let it be simple UI that allows inputting the required data)
    b. Application sends a text message to the household, waits for response for some time. Also do not worry about required communication for this purpose. Assume there is some way of doing it.
    c. If approval arrives, visitor ID is assigned to visitor, all data are recorded, and visitor is allowed to enter the society d. In case of non-response from house also visitor is allowed to proceed, however visits approval remains pending.

2. **Visitor Leaves Society**
    a. Visitor gives visitor id to the security, security guard make entry in the application, and visitor is allowed to leave.
    b. By this time if disapproval comes from house, visitor is not allowed to leave the society – an offline activity.

3. **Admin Reports**
    a. List Name, Contact Number, House No for all visitors on a given day
    b. List Name, Contact Number, House No for all visitors whose approval is still pending for a given date range.

c. List Name, Contact Number, and House No for all visitors whose approval is declined, and the visitor has left the society

Considering this functional description, your goal here is to -

1. Design and implement business classes for said requirements.

2. Let there be a Service object that is primarily responsible for
   a. Holding collection of all houses in the society
   b. Stores all visitor entries.
   c. Provides necessary services implementing required functionalities as described in the use cases. Following is an indicative list of operations that service may provide. Note this list may not complete and

   ```
   void addVisitor (visitor)
   void approve(visitor_id)
   Visitor getVisitor ( visitor_id )
   House getHouse ( house_no )
   Visitor_List getVisitorsDay(date);
   Visitor_List getVisitorsPendingApproval(date_from, date_to);
   Visitor_List getVisitorsUnApprovedLeft(date_from, date_to);
   ```

   Note following:

   - Interface given is not finalized; you can do necessary changes, whatever you think is most appropriate.
   - We do not have any database access here. Let all databases to be in the memory. You can use some map data structure for creating in memory database, for instance, HashMap in Java.
   - Also, note that the service class should not perform any input and output.

   For your reference here is a `BankService` class for Saving Bank Account use cases discussed in lectures.
   https://github.com/pmjat/j2ee/blob/master/j2ee/bank/src/main/java/jee/oop/bank/service/BankService.java

3. Create a console-based tester client, that
   a. Adds few houses, say 5
   b. Add few visitor entries, say 10 – entering to the society and exiting the society; let few of them be not approved, and few are declined.
   c. Print all reports (use-case-3) on console

   Note that client should not store anything locally; all storage responsibilities is delegated to Service object. Client only is responsible for collecting input, invoking appropriate method of Service object. Collect required data from service and print on console.

   May also refer client program for above BankService at
   https://github.com/pmjat/j2ee/blob/master/j2ee/bank/src/main/java/jee/oop/bank/client/BankClientConsole.java