Java Map Collections and Date API

ABC Tech has to build the system to sell the various insurance policies online, catering to different roles and operations.

Git link available

https://github.com/fsdtrinings/batch2Noi_assignments/tree/main/map

System Overview

The system enables online sales of various insurance policies, catering to different roles and operations.

Roles and Responsibilities

Admin

- Add new policies under specific categories

- View reminder lists to notify users based on input dates

- Generate business report

- Manage user accounts

User

- View all available policies

- Select and purchase policies

- View owned policies

- View upcoming premium payments for the current month

Functional Requirements

- Policy Management

  - Create, read, update, and delete (CRUD) policies

- User Management

  - Register and login users

  - Manage user accounts

- Reminder System

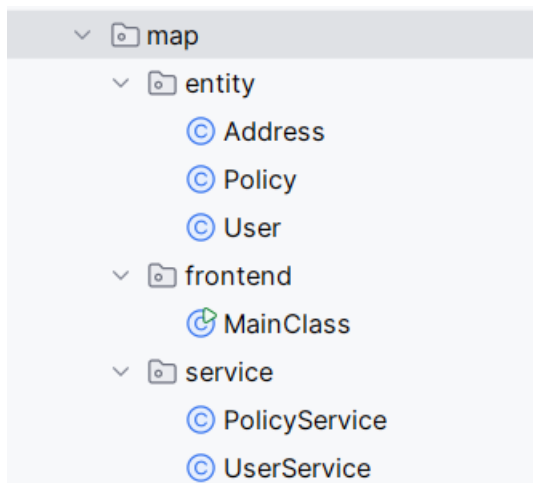 - Send reminders to users based on input dates
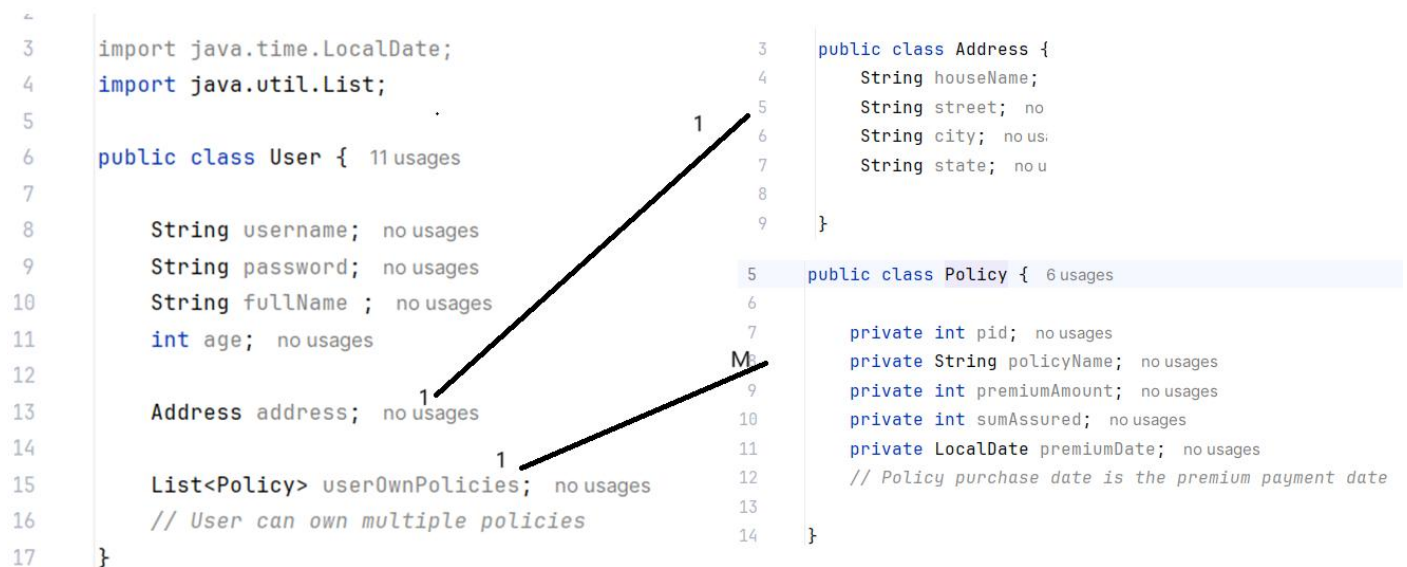
- Payment Management

  - Display upcoming payments

Non-Functional Requirements

- Usability: Create good user interface for easy navigation

Project Structure

- ⌄ ⬚ map
  - ⌄ ⬚ entity
    - © Address
    - © Policy
    - © User
  - ⌄ ⬚ frontend
    - © MainClass
  - ⌄ ⬚ service
    - © PolicyService
    - © UserService

User class

```java
3    import java.time.LocalDate;
4    import java.util.List;
5
6    public class User {  11 usages
7
8        String username;  no usages
9        String password;  no usages
10       String fullName ;  no usages
11       int age;  no usages
12
13       Address address;  no usages
14
15       List<Policy> userOwnPolicies;  no usages
16       // User can own multiple policies
17   }
```

```java
3    public class Address {
4        String houseName;
5        String street;  no
6        String city;  no us
7        String state;  no u
8
9    }
```

```java
5    public class Policy {  6 usages
6
7        private int pid;  no usages
8        private String policyName;  no usages
9        private int premiumAmount;  no usages
10       private int sumAssured;  no usages
11       private LocalDate premiumDate;  no usages
12       // Policy purchase date is the premium payment date
13
14   }
```

```java
7   public class UserService {  no usages
8
9       List<User> allUsers  = new ArrayList<~>();  no usages
10
11      UserService() {  no usages
12          /* Code to initilize allusers */
13          // Hint : Same as PolicyService
14      }
15
16      // getter & setter of allUsers
17
18      public List<Policy> getCurrentMonthPolicyPaymentReminders(LocalDate date) {  no usages
19
20          return null;
21      }
22  }
23
```

```java
3   import java.time.LocalDate;
4   import java.util.*;
5
6   public class PolicyService {  2 usages
7
8       Map<String,List<Policy>> allPolicies = new HashMap();  no usages
9       // contains all the policies available to sell in the system
10      /*Key : Policy Category like MedicalPolicy, Car Policy , Bike Policy , PensionPolicy , FamilyInsurance etc*/
11      /*Value List<Policy> : company has various policies under one category like multiple medical policies , multiple car policies*/
12
13      Map<LocalDate, List<User>> premiumReminder = new HashMap();  no usages
14      /*Once the User purchase the policy , user details & premium date should be added in the map
15      * Key : LocalDate : which is the policy sell date
16      * Value : List<User> : add user in the list
17      * */
18
19      PolicyService()  1 usage
20    > {}
23      // getter & setter of allPolicies
24
25    > public List<User> getTodayPremiumReminders(LocalDate today) { return null; }
29    > public List<User> getNext5DaysPremiumReminders(LocalDate today) { return null; }
33
34      // Developer can change the ? in return type
35      public Set<?> getSortedReportBasedOnCity()  no usages
36    > {...}
46
47  }
```

```java
 5       import java.util.List;

 6

 7 ▷    public class MainClass {

 8

 9            PolicyService policyService = new PolicyService();   1 usage

10

11 ▷        public static void main(String[] args) {

12

13            }
14            public void showAllAvailablePolicy(){   no usages

15

16            }
17            // Overloaded Method
18 >        public Policy getPolicy(int pId) { return null; }
21 >        public Policy getPolicy(String policyName) { return null; }

24

25 >        public User getUser(String username, String password) { return null; }
29            public void setPolicy(User user, Policy policy){}   no usages
30 >        public void printUserDetails(User user){}

35

36            public void getPaymentReminder(User user)   no usages
37 >        {}

42

43            public void printUsersForPaymentReminder()   no usages
44 >        {...}
52    }
```