

SOLID Principles - SafetyNet Alerts Application

Patrick Callaghan

PROG220

Final Project
– Alerts Notification System

November 18th 2024

SOLID Principle	Description	Applied Example in Your Code
Single Responsibility	<i>A class should have only one reason to change, meaning it should have only one job or responsibility.</i>	AlertsController.java handles HTTP requests and delegates business logic to AlertsService.java , ensuring separation of concerns. AlertsService focuses solely on business logic related to alerts. GlobalExceptionHandler.java exclusively manages exception handling across the application. AlertsService.java implements an interface (AlertsServiceInterface.java), allowing you to extend functionality by creating new implementations without altering the existing service class. Additionally, FireStationRepository and PersonRepository extend JpaRepository , enabling easy addition of new query methods without modifying existing code.
Open/Closed	<i>Software entities should be open for extension but closed for modification.</i>	FireStation and Person classes are well-defined entities. If you were to introduce subclasses (e.g., SeniorPerson extending Person), ensuring that they adhere to the Person class's behavior would maintain LSP. Currently, your code does not have subclass implementations that violate LSP.
Liskov Substitution	<i>Objects of a superclass should be replaceable with objects of subclasses without affecting behavior.</i>	PersonRepository.java and FireStationRepository.java define specific query methods relevant to their respective entities, ensuring that repositories only expose methods that clients need. This avoids bloated interfaces and adheres to ISP by providing focused, role-specific interfaces.
Interface Segregation	<i>Clients should not be forced to depend on interfaces they do not use.</i>	AlertsController.java currently depends on the concrete class AlertsService.java via @Autowired . To fully adhere to DIP, AlertsController should depend on an abstraction (interface) AlertsServiceInterface.java instead of the concrete implementation. This allows for greater flexibility and easier testing (e.g., mocking services during unit tests).
Dependency Inversion	<i>High-level modules should depend on abstractions, not on concrete implementations.</i>	

```
localhost:8080/firestation?stationNumber=1
{
  "persons": [
    {
      "id": 1,
      "firstName": "Homer",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1234",
      "email": "homer.simpson@example.com",
      "age": 42,
      "medications": [
        "aspirin:100mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 2,
      "firstName": "Marge",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1235",
      "email": "marge.simpson@example.com",
      "age": 40,
      "medications": [
        "vitaminC:500mg"
      ],
      "allergies": [
        "shellfish"
      ]
    },
    {
      "id": 3,
      "firstName": "Bart",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1236",
      "email": "bart.simpson@example.com",
      "age": 12,
      "medications": [],
      "allergies": []
    },
    {
      "id": 4,
      "firstName": "Lisa",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1237",
      "email": "lisa.simpson@example.com",
      "age": 10,
      "medications": [
        "ibuprofen:200mg"
      ],
      "allergies": [
        "peanut"
      ]
    },
    {
      "id": 5,
      "firstName": "Maggie",
      "lastName": "Simpson"
    }
  ]
}

localhost:8080/firestation?stationNumber=2
{
  "persons": [
    {
      "id": 6,
      "firstName": "Ned",
      "lastName": "Flanders",
      "address": "744 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-5678",
      "email": "ned.flanders@example.com",
      "age": 60,
      "medications": [
        "lisinopril:10mg"
      ],
      "allergies": [
        "latex"
      ]
    },
    {
      "id": 7,
      "firstName": "Rod",
      "lastName": "Flanders",
      "address": "744 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-5679",
      "email": "rod.flanders@example.com",
      "age": 35,
      "medications": [],
      "allergies": []
    },
    {
      "id": 8,
      "firstName": "Todd",
      "lastName": "Flanders",
      "address": "744 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-5680",
      "email": "todd.flanders@example.com",
      "age": 14,
      "medications": [],
      "allergies": [
        "pollen"
      ]
    },
    {
      "id": 13,
      "firstName": "Milhouse",
      "lastName": "Van Houten",
      "address": "300 School Lane",
      "city": "Springfield",
      "phone": "555-2345",
      "email": "milhouse.vh@example.com",
      "age": 10,
      "medications": [],
      "allergies": []
    },
    {
      "id": 14,
      "firstName": "Martin",
      "lastName": "Prince",
      "address": "300 School Lane",
      "city": "Springfield",
      "phone": "555-2346",
      "email": "martin.prince@example.com",
      "age": 11,
      "medications": []
    }
  ]
}

localhost:8080/firestation?stationNumber=2
{
  "persons": [
    {
      "id": 14,
      "firstName": "Martin",
      "lastName": "Prince",
      "address": "300 School Lane",
      "city": "Springfield",
      "phone": "555-2346",
      "email": "martin.prince@example.com",
      "age": 11,
      "medications": [
        "amoxicillin:500mg"
      ],
      "allergies": [
        "penicillin"
      ]
    },
    {
      "id": 26,
      "firstName": "Luann",
      "lastName": "Van Houten",
      "address": "300 School Lane",
      "city": "Springfield",
      "phone": "555-6780",
      "email": "luann.vh@example.com",
      "age": 28,
      "medications": [],
      "allergies": []
    },
    {
      "id": 28,
      "firstName": "Martin",
      "lastName": "Prince",
      "address": "300 School Lane",
      "city": "Springfield",
      "phone": "555-2347",
      "email": "martin.prince2@example.com",
      "age": 12,
      "medications": [
        "ibuprofen:200mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 37,
      "firstName": "Rainier",
      "lastName": "Wolfcastle",
      "address": "1300 Fitness Blvd",
      "city": "Springfield",
      "phone": "555-7890",
      "email": "rainier.wolfcastle@example.com",
      "age": 50,
      "medications": [
        "prednisone:10mg"
      ],
      "allergies": [
        "none"
      ]
    }
  ]
}
```

- http://localhost:8080/firestation?stationNumber=<station_number>

This URL should return a list of people serviced by the corresponding fire station. So if station number = 1, it should return the people serviced by station number 1. The list of people should include these specific pieces of information: first name, last name, address, and phone number. As well, it should provide a summary of the number of adults in the service area and the number of children (anyone aged 18 or younger).

```
localhost:8080/childAlert?address=742%20Evergreen%20Terrace
pretty print ☒

{
  "children": [
    {
      "id": 3,
      "firstName": "Bart",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1236",
      "email": "bart.simpson@example.com",
      "age": 12,
      "medications": [],
      "allergies": []
    },
    {
      "id": 4,
      "firstName": "Lisa",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1237",
      "email": "lisa.simpson@example.com",
      "age": 10,
      "medications": [
        "ibuprofen:200mg"
      ],
      "allergies": [
        "peanut"
      ]
    },
    {
      "id": 5,
      "firstName": "Maggie",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1238",
      "email": "maggie.simpson@example.com",
      "age": 2,
      "medications": [],
      "allergies": [
        "none"
      ]
    }
  ],
  "otherResidents": [
    {
      "id": 1,
      "firstName": "Homer",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1234",
      "email": "homer.simpson@example.com",
      "age": 42,
      "medications": [
        "aspirin:100mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 2,
      "firstName": "Marge",
      "lastName": "Simson"
    }
  ]
}
```

- **http://localhost:8080/childAlert?address=<address>**

This URL should return a list of children (anyone under the age of 18) at that address. The list should include the first and last name of each child, the child's age, and a list of other persons living at that address. If there are no children at the address, then this URL can return an empty string.

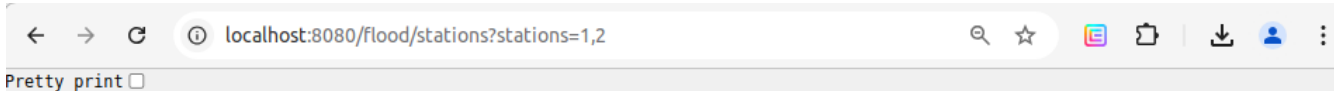
```
localhost:8080/fire?address=742%20Evergreen%20Terrace
pretty print ☒

{
  "stationNumber": 1,
  "residents": [
    {
      "id": 1,
      "firstName": "Homer",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1234",
      "email": "homer.simpson@example.com",
      "age": 42,
      "medications": [
        "aspirin:100mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 2,
      "firstName": "Marge",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1235",
      "email": "marge.simpson@example.com",
      "age": 40,
      "medications": [
        "vitaminC:500mg"
      ],
      "allergies": [
        "shellfish"
      ]
    },
    {
      "id": 3,
      "firstName": "Bart",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1236",
      "email": "bart.simpson@example.com",
      "age": 12,
      "medications": [],
      "allergies": []
    },
    {
      "id": 4,
      "firstName": "Lisa",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1237",
      "email": "lisa.simpson@example.com",
      "age": 10,
      "medications": [
        "ibuprofen:200mg"
      ],
      "allergies": [
        "peanut"
      ]
    },
    {
      "id": 5,
      "firstName": "Maggie",

```

- <http://localhost:8080/fire?address=<address>>

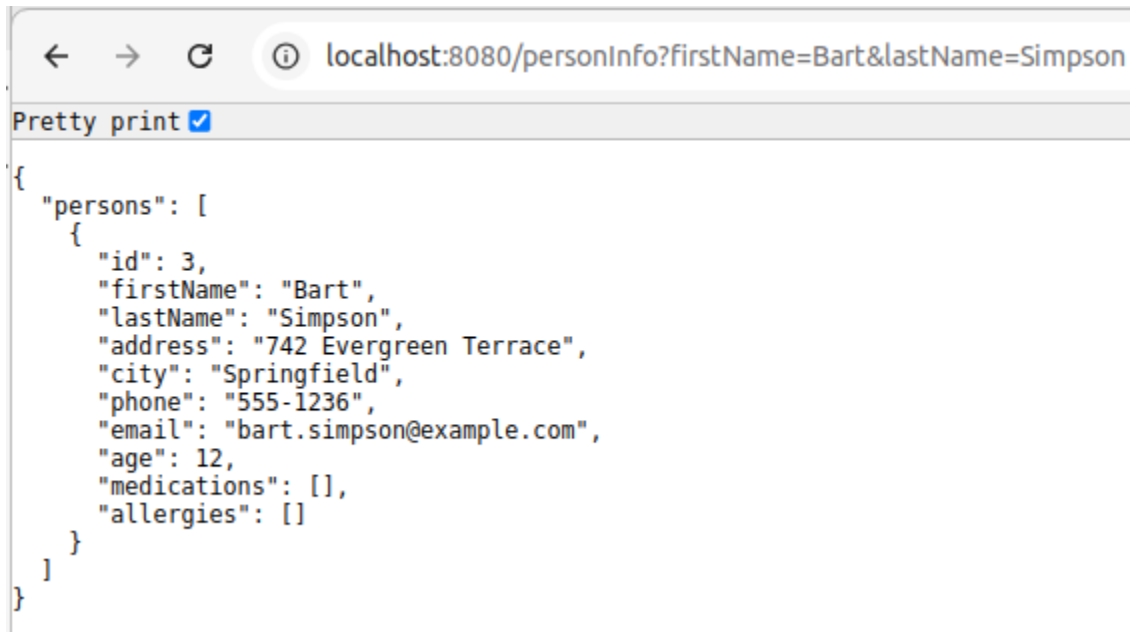
This URL should return the fire station number that services the provided address as well as a list of all of the people living at the address. This list should include each person's name, phone number, age, medications with dosage, and allergies.



```
{
  "households": [
    {
      "id": 35,
      "firstName": "Laura",
      "lastName": "Keene",
      "address": "1200 Community Drive",
      "city": "Springfield",
      "phone": "555-6789",
      "email": "laura.keene@example.com",
      "age": 32,
      "medications": [
        "atorvastatin:20mg"
      ],
      "allergies": [
        "latex"
      ]
    },
    {
      "id": 36,
      "firstName": "Frank",
      "lastName": "Grimes",
      "address": "1200 Community Drive",
      "city": "Springfield",
      "phone": "555-6790",
      "email": "frank.grimes@example.com",
      "age": 45,
      "medications": [
        "furosemide:40mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 37,
      "firstName": "Rainier",
      "lastName": "Wolfcastle",
      "address": "1300 Fitness Blvd",
      "city": "Springfield",
      "phone": "555-7890",
      "email": "rainier.wolfcastle@example.com",
      "age": 50,
      "medications": [
        "prednisone:10mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 38,
      "firstName": "Seymour",
      "lastName": "Skinner",
      "address": "1300 Fitness Blvd",
      "city": "Springfield",
      "phone": "555-7891",
      "email": "seymour.skinner@example.com",
      "age": 49,
      "medications": [
        "lisinopril:10mg"
      ],
      "allergies": [
        "dust"
      ]
    },
    {
      "id": 1,
      "firstName": "Homer",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1234",
      "email": "homer.simpson@example.com",
      "age": 42,
      "medications": [
        "aspirin:100mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 2,
      "firstName": "Marge",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1235",
      "email": "marge.simpson@example.com",
      "age": 40,
      "medications": [
        "vitaminC:500mg"
      ],
      "allergies": [
        "shellfish"
      ]
    },
    {
      "id": 3,
      "firstName": "Bart",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1236",
      "email": "bart.simpson@example.com",
      "age": 12,
      "medications": [
        "none"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 4,
      "firstName": "Lisa",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1237",
      "email": "lisa.simpson@example.com",
      "age": 10,
      "medications": [
        "ibuprofen:200mg"
      ],
      "allergies": [
        "peanut"
      ]
    },
    {
      "id": 5,
      "firstName": "Maggie",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1238",
      "email": "maggie.simpson@example.com",
      "age": 2,
      "medications": [
        "none"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 48,
      "firstName": "Dolph",
      "lastName": "Starbeam",
      "address": "1900 Health Hub",
      "city": "Springfield",
      "phone": "555-8916",
      "email": "dolph.starbeam@example.com",
      "age": 33,
      "medications": [
        "prednisone:10mg"
      ],
      "allergies": [
        "pollen"
      ]
    },
    {
      "id": 52,
      "firstName": "Sideshow",
      "lastName": "Bob",
      "address": "2200 Magnolia Ave",
      "city": "Springfield",
      "phone": "555-8917",
      "email": "sideshow.bob@example.com",
      "age": 48,
      "medications": [
        "omeprazole:40mg"
      ],
      "allergies": [
        "dust"
      ]
    },
    {
      "id": 53,
      "firstName": "Frank",
      "lastName": "Grimes",
      "address": "2200 Magnolia Ave",
      "city": "Springfield",
      "phone": "555-6793",
      "email": "frank.grimes4@example.com",
      "age": 48,
      "medications": [
        "furosemide:40mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 49,
      "firstName": "Ralph",
      "lastName": "Wiggum",
      "address": "2000 Spruce Street",
      "city": "Springfield",
      "phone": "555-2356",
      "email": "ralph.wiggum11@example.com",
      "age": 17,
      "medications": [
        "none"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 50,
      "firstName": "Ling",
      "lastName": "Bouvier",
      "address": "2000 Spruce Street",
      "city": "Springfield",
      "phone": "555-9034",
      "email": "ling.bouvier@example.com",
      "age": 9,
      "medications": [
        "none"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 15,
      "firstName": "Chief",
      "lastName": "Wiggum",
      "address": "200 Police Plaza",
      "city": "Springfield",
      "phone": "555-6789",
      "email": "chief.wiggum@example.com",
      "age": 50,
      "medications": [
        "atorvastatin:10mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 16,
      "firstName": "Lou",
      "lastName": "Griffin",
      "address": "200 Police Plaza",
      "city": "Springfield",
      "phone": "555-6790",
      "email": "lou.griffin@example.com",
      "age": 48,
      "medications": [
        "none"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 17,
      "firstName": "Eddie",
      "lastName": "Park",
      "address": "200 Police Plaza",
      "city": "Springfield",
      "phone": "555-6791",
      "email": "eddie.park@example.com",
      "age": 35,
      "medications": [
        "lisinopril:10mg"
      ],
      "allergies": [
        "latex"
      ]
    },
    {
      "id": 54,
      "firstName": "Jimbo",
      "lastName": "Jones",
      "address": "2300 Poplar Street",
      "city": "Springfield",
      "phone": "555-9035",
      "email": "jimbo.jones13@example.com",
      "age": 26,
      "medications": [
        "none"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 55,
      "firstName": "Cletus",
      "lastName": "Spuckler",
      "address": "2300 Poplar Street",
      "city": "Springfield",
      "phone": "555-9036",
      "email": "cletus.spuckler7@example.com",
      "age": 41,
      "medications": [
        "albuterol:90mcg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 6,
      "firstName": "Ned",
      "lastName": "Flanders",
      "address": "744 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-5678",
      "email": "ned.flanders@example.com",
      "age": 60,
      "medications": [
        "lisinopril:10mg"
      ],
      "allergies": [
        "latex"
      ]
    },
    {
      "id": 7,
      "firstName": "Rod",
      "lastName": "Flanders",
      "address": "744 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-5679",
      "email": "rod.flanders@example.com",
      "age": 35,
      "medications": [
        "none"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 8,
      "firstName": "Todd",
      "lastName": "Flanders",
      "address": "744 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-5680",
      "email": "todd.flanders@example.com",
      "age": 14,
      "medications": [
        "none"
      ],
      "allergies": [
        "pollen"
      ]
    },
    {
      "id": 24,
      "firstName": "Professor",
      "lastName": "Frink",
      "address": "700 Science Drive",
      "city": "Springfield",
      "phone": "555-6543",
      "email": "professor.frink@example.com",
      "age": 55,
      "medications": [
        "gabapentin:300mg"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 25,
      "firstName": "Sideshow",
      "lastName": "Gandalf",
      "address": "700 Science Drive",
      "city": "Springfield",
      "phone": "555-6544",
      "email": "sideshow.gandalf@example.com",
      "age": 42,
      "medications": [
        "prednisone:10mg"
      ],
      "allergies": [
        "pollen"
      ]
    },
    {
      "id": 13,
      "firstName": "Milhouse",
      "lastName": "Van Houten",
      "address": "300 School Lane",
      "city": "Springfield",
      "phone": "555-2345",
      "email": "milhouse.vh@example.com",
      "age": 10,
      "medications": [
        "none"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 14,
      "firstName": "Martin",
      "lastName": "Prince",
      "address": "300 School Lane",
      "city": "Springfield",
      "phone": "555-2346",
      "email": "martin.prince@example.com",
      "age": 11,
      "medications": [
        "amoxicillin:500mg"
      ],
      "allergies": [
        "penicillin"
      ]
    },
    {
      "id": 26,
      "firstName": "Luann",
      "lastName": "Van Houten",
      "address": "300 School Lane",
      "city": "Springfield",
      "phone": "555-6780",
      "email": "luann.vh@example.com",
      "age": 28,
      "medications": [
        "none"
      ],
      "allergies": [
        "none"
      ]
    },
    {
      "id": 28,
      "firstName": "Martin",
      "lastName": "Prince",
      "address": "300 School Lane",
      "city": "Springfield",
      "phone": "555-2347",
      "email": "martin.prince2@example.com",
      "age": 12,
      "medications": [
        "ibuprofen:200mg"
      ],
      "allergies": [
        "none"
      ]
    }
  ]
}
```

- http://localhost:8080/flood/stations?stations=<a list of station_numbers>

This should return a list of all the households in each fire station's jurisdiction. This list needs to group



```
localhost:8080/personInfo?firstName=Bart&lastName=Simpson
Pretty print ☒
{
  "persons": [
    {
      "id": 3,
      "firstName": "Bart",
      "lastName": "Simpson",
      "address": "742 Evergreen Terrace",
      "city": "Springfield",
      "phone": "555-1236",
      "email": "bart.simpson@example.com",
      "age": 12,
      "medications": [],
      "allergies": []
    }
  ]
}
```

- <http://localhost:8080/personInfo?firstName=<firstName>&lastName=<lastName>>

This should return the person's name, address, age, email, list of medications with dosages and allergies.

If there is more than one person with the same name, this URL should return all of them.


```
localhost:8080/communityEmail?city=Springfield
pretty print
{
  "emails": [
    "homer.simpson@example.com",
    "marge.simpson@example.com",
    "bart.simpson@example.com",
    "lisa.simpson@example.com",
    "maggie.simpson@example.com",
    "ned.flinders@example.com",
    "rod.flinders@example.com",
    "todd.flinders@example.com",
    "apu.n@example.com",
    "manjula.n@example.com",
    "krusty.clown@example.com",
    "sideshow.bob@example.com",
    "milhouse.vh@example.com",
    "martin.prince@example.com",
    "chief.wiggum@example.com",
    "lou.griffin@example.com",
    "eddie.park@example.com",
    "moe.szyslak@example.com",
    "barney.gumble@example.com",
    "carl.carlson@example.com",
    "lenny.leonard@example.com",
    "kent.brockman@example.com",
    "waylon.smithers@example.com",
    "professor.frink@example.com",
    "sideshow.gandalf@example.com",
    "luann.vh@example.com",
    "nelson.muntz@example.com",
    "martin.prince2@example.com",
    "kearney.z@example.com",
    "jimbo.jones@example.com",
    "ralph.wiggum@example.com",
    "dolph.starbeam@example.com",
    "sally.bouvier@example.com",
    "ling.bouvier@example.com",
    "laura.keene@example.com",
    "frank.grimes@example.com",
    "rainier.wolfcastle@example.com",
    "seymour.skinner@example.com",
    "edna.krabappel@example.com",
    "nelson.muntz@example.com",
    "sideshow.gandalf@example.com",
    "jimbo.jones@example.com",
    "ralph.wiggum@example.com",
    "cletus.spuckler@example.com",
    "seymour.skinner8@example.com",
    "edna.krabappel7@example.com",
    "nelson.muntz7@example.com",
    "dolph.starbeam6@example.com",
    "ralph.wiggum11@example.com",
    "ling.bouvier5@example.com",
    "kearney.zyzwicz6@example.com",
    "sideshow.bob4@example.com",
    "frank.grimes4@example.com",
    "jimbo.jones13@example.com",
    "cletus.spuckler7@example.com"
  ]
}
```

- <http://localhost:8080/communityEmail?city=<city>>

This will return the email addresses of all of the people in the city

For all of these endpoints, the results should be JSON, and if there is an address or fire station number not found within our file, it should return an empty JSON object. We'll also need a set of unit tests that test each of the requirements and validate that the application working correctly. Also, don't forget about logging. The application should log every request and every response.

Also, SafetyNet Alerts need to be architected following the Model-View-Controller design pattern. If you follow the SOLID principles and separate your model from your controllers, you'll be right in line with our coding best practices. When coding up the application, please commit often, so that we can validate little by little; this will help us piece together the overall flow of development, and give us an idea of lines of code per day and other metrics that will be helpful in gauging the timeline for the next phase of SafetyNet Alerts.

When you've finished SafetyNet Alerts, you prepare a presentation with a description of your solution. This document will serve as additional documentation to other team members who will be working on SafetyNet Alerts during our next phase of development.

Alert Service and Alert Controller Mockito Tests cases

Run FinalProject1Application x AlertsControllerTest (1) x

Tests passed: 9 of 9 tests – 189 ms

AlertsControllerTest (com.example.finalproject1. 189 ms)

- ✓ testGetPhoneAlert_NotFound() 99 ms
- ✓ testGetFloodDetailsByStations_Found() 43 ms
- ✓ testGetPeopleByFireStation_NotFound() 7 ms
- ✓ testGetPhoneAlert_Found() 5 ms
- ✓ testGetFloodDetailsByStations_NotFound() 6 ms
- ✓ testGetPeopleByFireStation_MissingParam() 9 ms
- ✓ testGetPeopleByFireStation_Found() 6 ms
- ✓ testGetChildAlert_NotFound() 9 ms
- ✓ testGetChildAlert_Found() 5 ms

13:13:46.238 [main] INFO org.springframework.boot.test...
13:13:46.288 [main] INFO org.springframework.boot.test...
:: Spring Boot :: (v3.3.4)
2024-11-17T13:13:46.445-04:00 INFO 117297 --- [Fina...
2024-11-17T13:13:46.446-04:00 INFO 117297 --- [Fina...

Run FinalProject1Application x AlertsServiceTest x

Tests passed: 14 of 14 tests – 79 ms

AlertsServiceTest (com.example.finalproject1) 79 ms

- ✓ testGetCommunityEmails_NotFound() 29 ms
- ✓ testGetPhoneNumbersByFireStation_NotFound() 6 ms
- ✓ testGetFireDetailsByAddress_Found() 5 ms
- ✓ testGetPersonInfo_Found() 4 ms
- ✓ testGetFloodDetailsByStations_Found() 5 ms
- ✓ testGetPhoneNumbersByFireStation_Found() 4 ms
- ✓ testGetPersonInfo_NotFound() 4 ms
- ✓ testGetPeopleByFireStation_NotFound() 4 ms
- ✓ testGetFloodDetailsByStations_NotFound() 3 ms
- ✓ testGetChildrenByAddress_Found() 3 ms
- ✓ testGetChildrenByAddress_NotFound() 3 ms
- ✓ testGetCommunityEmails_Found() 4 ms
- ✓ testGetFireDetailsByAddress_NotFound() 3 ms
- ✓ testGetPeopleByFireStation_Found() 2 ms

/home/patrick/.jdk/openjdk-23/bin/java ...
13:06:27.027 [main] INFO org.springframework.test.co...
13:06:27.135 [main] INFO org.springframework.boot.te...
13:06:27.239 [main] INFO org.springframework.boot.de...
:: Spring Boot :: (v3.3.4)
2024-11-17T13:06:27.385-04:00 INFO 116214 --- [Fina...
2024-11-17T13:06:27.386-04:00 INFO 116214 --- [Fina...
2024-11-17T13:06:27.820-04:00 INFO 116214 --- [Fina...
2024-11-17T13:06:27.860-04:00 INFO 116214 --- [Fina...
2024-11-17T13:06:28.141-04:00 INFO 116214 --- [Fina...
2024-11-17T13:06:28.278-04:00 INFO 116214 --- [Fina...
2024-11-17T13:06:28.288-04:00 INFO 116214 --- [Fina...

The image displays four browser windows, each showing a JSON response from a web application. The address bar for all windows is `localhost:8080/phoneAlert?stationNumber=` followed by a number. The 'Pretty print' checkbox is checked in each window.

- Window 1 (stationNumber=1):** Returns a list of 14 phone numbers for station 1.
- Window 2 (stationNumber=3):** Returns a list of 11 phone numbers for station 3.
- Window 3 (stationNumber=2):** Returns a list of 16 phone numbers for station 2.
- Window 4 (stationNumber=4):** Returns a list of 10 phone numbers for station 4.

```
{
  "phoneNumbers": [
    "555-1234",
    "555-1235",
    "555-1236",
    "555-1237",
    "555-1238",
    "555-6789",
    "555-6790",
    "555-6791",
    "555-6543",
    "555-6544",
    "555-6789",
    "555-6790",
    "555-6791",
    "555-6543"
  ]
}
```

```
{
  "phoneNumbers": [
    "555-9012",
    "555-9013",
    "555-7890",
    "555-7891",
    "555-3456",
    "555-3457",
    "555-8901",
    "555-8902",
    "555-7899",
    "555-8905"
  ]
}
```

```
{
  "phoneNumbers": [
    "555-5678",
    "555-5679",
    "555-5680",
    "555-2345",
    "555-2346",
    "555-6780",
    "555-2347",
    "555-7890",
    "555-7891",
    "555-2356",
    "555-9034",
    "555-9035",
    "555-9036"
  ]
}
```

```
{
  "phoneNumbers": [
    "555-3456",
    "555-3457",
    "555-4567",
    "555-4568",
    "555-4567",
    "555-4568",
    "555-9012",
    "555-9013"
  ]
}
```

- http://localhost:8080/phoneAlert?firestation=<firestation_number>
This URL should return a list of phone numbers of each person within the fire station's jurisdiction. We'll use this to send emergency text messages to specific households.