**WSI Code Exercise Technical Specification**

**Problem:**

Given a collection of 5-digit ZIP code ranges (each range includes both their upper and lower bounds), provide an algorithm that produces the minimum number of ranges required to represent the same restrictions as the input.

**Assumption**:  
a) Zip code will be 5 digit in length. If given zipcode range has zipcode less than or greater than 5 digits length, code will treat it as bad data.   
b) Input will be List of **ZipcodeRange**.

**Algorithm**:

Filter out bad data and sort list of ranges as per *zipcodeStart*. Get first data range from filtered list and push into stack. Compare top element in stack with other in filtered list and modify, push and remove accordingly.

**Class Diagram**:

|  |
| --- |
| **ZipcodeRange** |
| - zipcodeStart : int  - zipcodeEnd : int |
| + ZipcodeRange(start : int , end : int)  + setter & getter methods |

**Class and Methods**:

|  |  |  |
| --- | --- | --- |
| **Class** | **Method** | **Details** |
| ZipcodeController | public static void main(String[] args) | Create input data and invoke operation method. |
| private static void zipcodeOperations(List<ZipcodeRange> zipList) | Invoked by main method, this method is using getMinimalZipcodeRangeList in ZipcodeService to display minimal zipcode list. |
| ZipcodeService | public List<ZipcodeRange> getMinimalZipcodeRangeList(List<ZipcodeRange> zipcodeList) | This method is calculating the list of minimum ranges, from given list of ranges. |
| public List<ZipcodeRange> reorderZipcodeRangeList(List<ZipcodeRange> zipList) | This method is touching each ZipcodeRange object in list and cleanup null and out of range value. Swapping value if zipcodeEnd < zipcodeStart. at last sorting the filtered list. |

**Sequence Diagram**: