CSCI 3901 Project Tests and Timeline

Shubham Mishra (B00917146)

Blackbox tests

Input Validation

- addPostalCode(String postalCode, int population, int area)
  - postalCode is NULL
  - postalCode is empty
  - population <= 0
  - area <= 0
- addDistributionHub(String hubIdentifier, Point location, Set<String> servicedAreas)
  - hubIdentifier is NULL
  - hubIdentifier is empty
  - location is NULL
  - servicedAreas is NULL
  - servicedAreas set contains empty strings
- hubDamage(String hubIdentifier, float repairEstimate)
  - hubIdentifier is NULL
  - hubIdentifier is empty
  - repairEstimate is negative
- hubRepair(String hubIdentifier, String employeeId, float repairTime, boolean inService)
  - hubIdentifier is NULL
  - hubIdentifier is empty
  - employeeId is NULL
  - employeeId is empty
  - repairTime is negative
- mostDamagedPostalCodes (int limit)
  - limit is negative
- fixOrder (int limit)
  - limit is negative
- rateOfServiceRestoration (float increment)
  - increment<=0
  - increment is greater than 1
- repairPlan (String startHub, int maxDistance, float maxTime)
  - startHub is NULL
  - startHub is empty
  - maxDistance is negative
  - maxTime is negative
- underservedPostalByPopulation (int limit)
  - limit is negative
- underservedPostalByArea (int limit)
  - limit is negative

Boundary Cases

- addPostalCode(String postalCode, int population, int area)
    - postalCode contains 1 letter
    - postalCode contains 6 letters
    - population is 1
    - area is 1
- addDistributionHub(String hubIdentifier, Point location, Set<String> servicedAreas)
    - hubIdentifier contains 1 letter
    - hubIdentifier is long string
    - servicedAreas contains 1 element
- hubDamage(String hubIdentifier, float repairEstimate)
    - repairEstimate is 0
- hubRepair(String hubIdentifier, String employeeId, float repairTime, boolean inService)
    - hubIdentifier contains 1 letter
    - employeeId contains 1 letter
    - hubIdentifier is long string
    - employeeId is long string
    - repairTime is 0
- mostDamagedPostalCodes (int limit)
    - limit is 0
    - limit is 1
- fixOrder (int limit)
    - limit is 0
    - limit is 1
- rateOfServiceRestoration (float increment)
    - increment is 0
    - increment is 1
- repairPlan (String startHub, int maxDistance, float maxTime)
    - maxDistance is 0
    - maxTime is 0
- underservedPostalByPopulation (int limit)
    - limit is 0
    - limit is 1
- underservedPostalByArea (int limit)
    - limit is 0
    - limit is 1

Control flow

- addPostalCode(String postalCode, int population, int area)
    - postalCode already exist in system
    - postalCode is new addition

- addDistributionHub(String hubIdentifier, Point location, Set<String> servicedAreas)
    - hubIdentifier already exist in system
    - hubIdentifier is new addition
    - servicedAreas set contains postalCodes that are not yet added in the system
    - servicedAreas set contains postalCodes that are present in the system
- hubDamage(String hubIdentifier, float repairEstimate)
    - hubIdentifier does not exist in the system
    - hubIdentifier exists in the system
- hubRepair(String hubIdentifier, String employeeId, float repairTime, boolean inService)
    - hubIdentifier does not exist in the system
    - hubIdentifier exists in the system
    - hub is fully repaired and ready to go
    - hub is not fully repaired and not ready to go
- peopleOutOfService()
    - Multiple hubs do not serve a postal code
    - Multiple hubs do serve a postal code
- mostDamagedPostalCodes (int limit)
    - number of damaged postal codes is less than limit
    - number of damaged postal codes is greater than limit
    - number of damaged postal codes is equal to limit
- fixOrder (int limit)
    - number of damaged postal codes is less than limit
    - number of damaged postal codes is greater than limit
    - number of damaged postal codes is equal to limit
- rateOfServiceRestoration (float increment)
    - all people already have power
    - some people do not have power
- repairPlan (String startHub, int maxDistance, float maxTime)
    - no hub is present within maxDistance from startHub
    - atleast 1hub is present within maxDistance from startHub
    - all hubs within maxDistance from startHub have repairTime greater than maxTime
    - atleast 1 hub within maxDistance from startHub has repairTime less than or equal to maxTime
    - all hubs within maxDistance from startHub are in service
    - atleast 1 hub within maxDistance from startHub needs repair
    - all hubs that can be repaired between first and last hub lie on the same side of the diagonal of rectangle
    - all hubs that can be repaired between first and last hub are spread out on both sides of the diagonal of rectangle
- underservedPostalByPopulation (int limit)
    - number of damaged postal codes is less than limit
    - number of damaged postal codes is greater than limit
    - number of damaged postal codes is equal to limit
- underservedPostalByArea (int limit)
    - number of damaged postal codes is less than limit
    - number of damaged postal codes is greater than limit

- number of damaged postal codes is equal to limit

Data flow

Data entry methods: addPostalCode, addDistributionHub, hubDamage, hubRepair
reporting methods: peopleOutOfService , mostDamagedPostalCodes , fixOrder, rateOfServiceRestoration, underservedPostalByPopulation, underservedPostalByArea
Planning methods: repairPan

- Try typical invocation
  - Create PowerService class object
  - Call addPostalCode to add some postal codes
  - Call addDistributionHub to add some hubs
  - Call hubDamage to report damages to some hubs
  - Call hubRepair
  - Call peopleOutOfService
  - Call mostDamagedPostalCodes
  - Call fixOrder
  - Call rateOfServiceRestoration
  - Call repairPlan
  - Call underservedPostalByPopulation
  - Call underservedPostalByArea
- Check for spots where order shouldn't matter
  - Call reporting/planning methods in any order after the data has been added using data entry methods
- Check for orders that leave information unknown
  - Call addDistributionHub that contains a postal code that has not yet been added using addPostalCode
  - Call hubDamage, hubRepair before calling addDistributionHub of the hub that was damaged or repaired
  - Call any reporting/planning method before any data has been entered using data entry methods.

Plan+Timeline of Feature Development

Data entry methods: addPostalCode, addDistributionHub, hubDamage, hubRepair
reporting methods: peopleOutOfService , mostDamagedPostalCodes , fixOrder, rateOfServiceRestoration, underservedPostalByPopulation, underservedPostalByArea
Planning methods: repairPan

Nov 24 - Nov 30

- Designing of data entry methods
- Designing of reporting methods
- Planning on where to use SQL queries
- Integrating SQL database with project

Dec 1 - Dec 7

- Designing of planning method i.e. repairPlan
- Implementation of data entry methods
- Implementation of some easy reporting methods
- Testing above implemented methods
- Preparing external documentation of data structures, code design, key algorithms

Dec 8 - Dec 15

- Implementation of remaining reporting methods
- Implementation of repairPlan
- Testing above implemented methods
- Optimizing methods
- Exception handling
- Code cleaning
- Testing whole project
- Adding comments for documentation