**IMPORTANT INSTRUCTIONS**

* 1. **Please read the document thoroughly before you code.**
  2. **Import the given skeleton code into your Eclipse.**
  3. **Use Java 8 for solving the code challenge.**
  4. **Run the database script provided to set up your database.**
  5. **You have to test the code and ensure there are no compilation errors before submission**

**I. Business Scenario:**

A leading construction company wants to expedite their process of cost estimations for their various ongoing projects. The company wants to automate the process of this cost & time estimation aspect, for all those projects starting this financial year.

The construction project details of the company are stored in a .txt file (as a comma separated fields). This file contains the details/records of the projects under various categories like Commercial, Residential and Infrastructural. The cost and Time duration in months are to be calculated for all those projects starting this financial year.

The proposed system is supposed to pick up the project details which are commencing in the current financial year from the source data and calculate the cost & duration of the project in months, based on the data provided in the requirement specification. After calculating, the system needs to persist those records in the database.

1. **Functional Requirement Specification:**

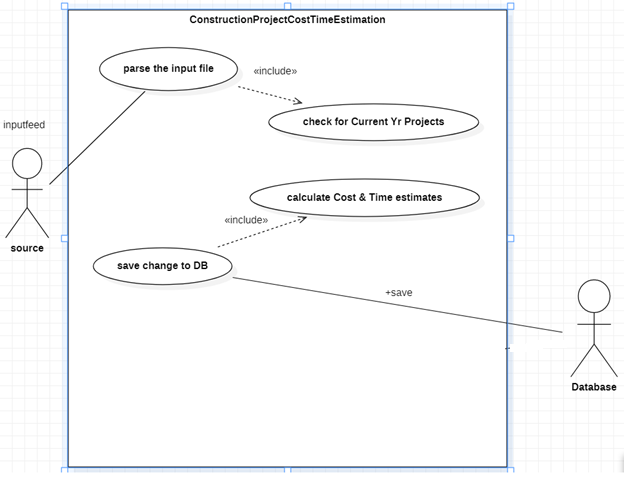
|  |  |  |
| --- | --- | --- |
| **Req. #** | **Req. Name** | **Req. Description** |
| **1** | Parse Input | The input file has to be parsed and Project records need to filter based on the start date of the projects. |
| **2** | Update the Projects estimated Cost and duration and persist the data in the database | The Project’s Cost & Duration have to be calculated and then the modified records need to be saved in the database. |

1. **Skeleton File for Development**

Import the below attached skeleton code into your eclipse project and implement the required functionalities. The skeleton also has .SQL file which can be used to set up your database.



1. **Use case Diagram**



1. **Technical Requirements**

For both the functional requirements 1 and 2, component specification and method specification are given below. Please follow the same order to implement them using the code skeleton.

1. **A. Component Specification:**

|  |  |
| --- | --- |
| ***Requirement Name*** | **Parse Input** |
| ***Component Definition*** | Reads the input text file, and convert the data into objects |
| ***Files Included***  ***(refer Skeleton)*** | ConstructionProjectEstimationService.java, ApplicationUtil.java, ConstructionProject.java, inputfeed.txt,ConstructionEstimationException.java |
| ***Responsibilities*** | * Reads the input file. * Perform validation to check if the Construction Project is planned for the current year. * Build and return the ConstructionProject value object. |
| ***Resources*** | Input File Name: inputfeed.txt  **File Structure:**  <projectId>,<plannedDOStart>, <typeOfProject>, < structure>, <areaInSqFt>, <estimatedCostInlac>, <estimatedTimeInMonths> |
| ***Design Constraints*** | 1. Input file format is .txt and is comma separated (Sample rows are added. You can add any number of rows to test your service class, from main method. 2. Do not hard code the input file name inside any method .It has to be referred from the input argument only as per code skeleton. 3. From the given input file, filter the Construction Project details, which are planned for the current financial year. 4. Assume that the Currency related fields are in INR. 5. Assume that the Date of Start in the file will be in the format yyyy-MM-dd. 6. Do not change the data types of the value object given in POJO. 7. Always convert the date of start value to java.util.date with format, yyyy-MM-dd before setting in ConstructionProject value object. 8. Use ApplicationUtil.java for reading file, performing date operations, etc.   **Note**: Sample rows are given in the input file. You can add any number of rows to test your service class, from main method. |
| ***Process Flow*** | 1. The app will be invoked by calling the ConstructionProjectEstimationService. addConstructionProjectDetails with the inputfeed (.txt file) 2. Read the file using File I/O or Java Streams in ApplicationUtil.readFile method 3. Return a list of Construction Project rows from input file, from the readFile method considering records which are passing the current financial year check constraint. 4. Code the method ConstructionProjectEstimationService. buildConstructionProjectList. Call the readFile method from this method. Read every line from the list returned by readFile method, split the records based on comma separator and use 5. For each record calculate the time and cost involved by calling the appropriate service method 6. The method should return a list of ConstructionProject objects. 7. Use the ApplicationUtil. convertStringToDate method to convert the date from String Format to java.util.Date format (yyyy-MM-dd). 8. Build the ConstructionProject Value Object from the values obtained in every line (Check the Input file format in Design Constraints row) |
| ***Exceptional Conditions*** | While doing File I/O in the ApplicationUtil.readFile method, catch all exceptions and throw application specific exception, ConstructionEstimationException. |

1. **B. Method Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Class******Name*** | ***Method Name*** | ***Input Parameters*** | ***Output Parameters*** |
| ConstructionProjectEstimationService | addConstructionProjectDetails | String inputFeed | boolean |
| ApplicationUtil | readFile | String fileName | static ArrayList<String> |
| ConstructionProjectEstimationService | buildConstructionProjectList | List <String> consProjectRecords | ArrayList <ConstructionProject> |
| ApplicationUtil | convertStringToDate | String input Date | Date |

1. **A. Component Specification:**

|  |  |
| --- | --- |
| ***Requirement Name*** | Persist Data into Database |
| ***Component Definition*** | Helps to calculate the costEstimation & timeEstimation values and add the changes made to the database. |
| ***Files Included***  ***(refer Skeleton)*** | ConstructionProjectEstimationService.java, ApplicationUtil.java, ConstructionProject.java, DBConnectionManager.java, ConstructionEstimationException |
| ***Responsibilities*** | Updates ConstructionProject object, based on area, structure and typeOfProject. Persists all ConstructionProject details to the database. |
| ***Design Constraints*** | 1. The database.properties has connection details required to connect to the backend 2. Do not change the keys of the property files, you can update the values based on the local database settings. For example, do not change the key, db.username. Rather you can have any value as user name based on local settings. 3. Use only JDBC to establish Database connection 4. Assume the location of the property file will be always as given in the skeleton. 5. Don’t Hardcode the connection string to establish database connection. Read it from property files. 6. Use Prepared Statement to insert records 7. Close all the resources after use 8. Catch all database related exception and throw Application specific exception only from DAO or from DBConnectionManager class. There has to be a private constructor in DBConnectionManager class, to load the database property file and to establish a database connection using JDBC 9. Rollback the Insert if any SQL exception has occurred. Throw application specific exception, ConstructionEstimationException 10. Calculate CostEstimation & TimeEstimation, based on the constraints in the table given below  |  |  |  |  | | --- | --- | --- | --- | | **Project Type** | **Structure** | **Cost PerSqFt(Rs)** | **TimeInMonths/1000sqft** | | Commercial | Shopping Complex | 2600 | 0.23 | | Commercial | ResApartments | 2750 | 0.24 | | Commercial | Community Hall | 2600 | 0.2 | | Infrastructural | Bridge | 10000 | 0.25 | | Infrastructural | FlyOver | 14000 | 0.22 | | Infrastructural | UnderPass | 8000 | 0.25 | | Residential | House | 2250 | 0.26 | | Residential | Apartments | 2500 | 0.24 | | Residential | Villa | 2750 | 0.23 |   Based on the type of the Project & the Structure , according to the required area of Construction, the cost & time have to be calculated based on the base data available in the above table:  For eg. If the Project Type is “Commercial” and the structure is “Shopping Complex” the cost incurred for the construction of per sq. ft is Rs.2600 and the time taken for the construction of the 1000 sq ft of the same project is 0.23 Months, calculation has to be performed on the similar basis i.e Pro rata basis depending upon the type and the area of construction. |
| ***Resources*** | database.properties – has connection details, used to establish database connection. |
| ***Process Flow*** | 1. Modify the ConstructionProjectEstimationService.buildConstructionProjectList method (refer the above section) then set updated costEstimation & timeEstimation values to ConstructionProject objects.   b. Use ConstructionProjectEstimationService. estimateTimeAndCostForConstruction () method to calculate the costEstimation & timeEstimation based on the Structure,TypeOfConstruction & areaInSqFt passed as parameters,   1. The method ConstructionProjectEstimationService. buildConstructionProjectList must return the list of ConstructionProject objects with updated costEstimation & timeEstimation fields, let’s persist the records as it is.   d. After reading file, building records as List<ConstructionObject>, call the CostAndTimeEstDAO. insertConstructionProject method to insert values to database. You may have to convert the java.util.date to java.sql.date before storing to database.  e.If Insert has happened successfully, return true; false otherwise.  f. Use getConstructionProjectsData to fetch the updated records and display. |
| ***Exceptional Conditions*** | While working with DAO methods, catch all exceptions and throw application specific exception, ConstructionEstimationException. |

1. **B. Method Specification:**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Class******Name*** | ***Method Name*** | ***Input Parameters*** | ***Output Parameters*** |
| ConstructionProjectEstimationService | estimateTimeAndCostForConstruction() | String projectType,String structure,double areaInSqFt | double[] |
| ConstructionProjectEstimationService | buildConstructionProjectList | List <String> consProjectRecords | ArrayList <ConstructionProject> |
| DBConnectionManager | DBConnectionManager() | NA | NA |
| DBConnectionManager | getInstance() | NA | DBConnectionManager |
| CostAndTimeEstDAO | insertConstructionProject | List< ConstructionProject > | boolean |

**Note:** You are allowed to modify input file text to incorporate more test data for various test scenarios / boundary conditions. Test your application by invoking the service methods from the main class, main () method. Follow Java Naming Conventions, test the code quality by running PMD rules in Eclipse or any other IDE that you use.

**Sample Output Data:**

CP001 2020-11-17 Residential Apartments 100000 250000000 24

CP002 2020-09-15 Commercial ShoppingComplex 200000 520000000 46

CP003 2020-08-10 Residential Villa 1000000 2750000128 230

CP004 2020-07-13 Commercial CommunityHall 100000 260000000 20

CP005 2020-08-12 Infrastructural FlyOver 1000000 14000000000 220