**Holiday Package**

Varsh Tours and Travels is a famous travel agency in the city that organizes trips and vacations. Every month they face problems in managing the holiday packages manually, and optimizing their available resources like resorts, transports and tour guides effectively. So, to provide the best in class customer experience, Varsh Tours and Travels plans to automate all their processes in trip booking and managing resources. Help them by automating the process of Holiday package manipulation.

Varsh Tours and Travels has the following business processes that must be automated.

1. Parse (read, split and process) data and calculate the package cost for all the packages
2. Extract the package details with minimum number of days

## Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. # | Req. Name | Req. Description | Actors / Users | Comments |
| 1 | Parse data and calculate the package cost for all the packages | The package details like package Id, source place, destination place, basic fare and number of days for each package are stored in a flat file. Retrieve the data from the file and calculate the package cost, based on the basic fare and number of days for each package. | Admin | The admin of Varsh Tours and Travels is responsible for parsing the data and calculating the cost for all the packages. |

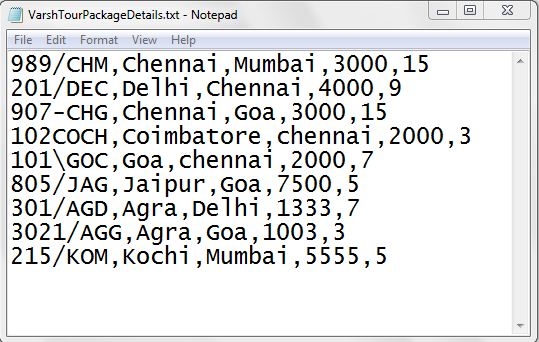
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | Extract the details of packages with minimum number of days | The package details including the number of days and cost of each package are stored in the database. From the database, Varsh Tours and Travels will find the package with the minimum number of days and display the package details. | Admin | The admin of Varsh Tours and Travels is responsible for identifying and retrieving the details of those packages with minimum number of days. |

### Parse data and calculate the package cost for all packages

The tour package details like package Id, source place, destination place, basic fare and number of days for each package are stored in a flat file. Retrieve the data from the file and calculate the package cost based on the basic fare and the number of days for each package. The package details are stored in a file named VarshTourPackageDetails.txt.

Sample file containing package details is shown below. The file is comma delimited. (values separated by commas)

[VarshTourPackageDetailsFormat: packageId,sourcePlace,destinationPlace,basicFare,noOfDays]



From the VarshTourPackageDetails.txt file, read the details, process the data, and construct a Package object for each record in the file. Now calculate the package cost for each package, based on the conditions given below, and then set the package cost.

|  |  |
| --- | --- |
| noOfDays | discount % |
| <=5 | 0% (No discount) |
| >5 and <=8 | 3% |
| >8 and <=10 | 5% |
| >10 | 7% |

Package Cost = ((Basic fare x number of days)-discount)+GST

The package cost should be calculated based on the basic fare and the number of days. The discount should be calculated depending on the number of days as given in the above table and deducted from the calculated package cost. Finally, a GST of 12% of the calculated package cost got after the discount, should be added to get the final package cost.

**For example:** If a package has a basic fare as Rs.3000 and the number of days as 15, then the package cost will be (3000\*15), which is Rs. 45000.00. Since the number of days is 15, the discount percentage will be 7%. So, the discount will be (45000.0\*(7/100)) which is Rs. 3150.00. Now, 12% of GST needs to be added. So the GST will be ((45000.0-3150.0)\*(12/100)) which is Rs. 5022.00.

Therefore, the total cost for this package will be (((3000\*15)-3150.0)+5022.0) which is Rs. 46872.00.

After calculating the cost for each package, set the package cost and then store the entire Package object into a list.

**Validation:**

The packageId should be validated before calculating the package cost; only if the packageId is valid, the Package object should be added to the list.

The packageId should be in the following format.

1. The packageId should contain exactly 7characters
2. The fourth character must be a forward slash symbol (/)

If the packageId is valid, then parse the data and calculate the package cost, else throw a user defined Exception “InvalidPackageIdException” with a message "Invalid Package Id".

**Note: This functionality is about only reading the records from the file, parsing each record data, validating the package Id, creation of Package object and then storing the Package Object into the list and save the list values to the database.**

**Component Specification: Package(model class)**

|  |  |  |
| --- | --- | --- |
| **Type(Class)** | **Attributes** | **Methods** |
| Package | String packageId  String sourcePlace  String destinationPlace  double basicFare  int noOfDays  double packageCost | Include getters and setters method for all the attributes. |

**Component Specification: Package(model class)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | **Type(Class)** | **Methods** | **Responsibilities** |
| Calculate the package cost for a Package | Package | void calculatePackageCost() | This method should calculate and set the package cost based on the basic fare and number of days. |

**TravelAgency(utility class)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Type(Class)** | **Method** | **Responsibilities** | **Exception** |
| Parse data and calculate the package cost for all the packages | TravelAgency | List<Package> generatePackageCost (String filePath) | This method should take the file path as the argument, parse the data stored in the file, and validate the packageId by invoking the validate(String packageId) method. If packageId is valid, construct a Package object for each record in the file, and then calculate the package cost by invoking the calculatePackageCost method of Package class. After calculating the package cost, each Package should be added in the list and this method should return the list of Packages. |  |
| Parse data and calculate the package cost for all the packages | TravelAgency | boolean validate(String packageId) | This method should validate the packageId. If it is valid it should return true, else this method should throw a user defined exception. | Throw a user defined exception “InvalidPackageIdException” if the packageId is invalid. |

**Note:** The data file will contain both valid and invalid details. Valid package details should be added to the list and user defined exception should be thrown for all the records with invalid packageId.

### Extract the package details with minimum number of days

The package details including the number of days and cost of each package are **already stored in the database**. From the database, Varsh Tours and Travels will find the package with the minimum number of days and display the package details.

**Component Specification: TravelAgency(utility class)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Type(Class)** | **Method** | **Responsibilities** | **Resources** |
| Extract the package details with minimum number of days | TravelAgency | List<Package> findPackagesWithMinimumNumberOfDays() | This method should extract all the packages with minimum number of days(based on no\_of\_days column)from the Package\_Details table and return the list.  Connect to the database by invoking the establishConnection() method of DBHandler class. | MYSQL database is used. Retrieve the details from Package\_Details table |

**DBHandler(DAO class)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Component Name** | **Type(Class)** | **Method** | **Responsibilities** | **Resourc****es** |
| Extract the package details with minimum number of days | DBHandler | Connection establishConnection() | This method should connect to the database by reading the database details from the db.properties file and it should return the connection object | MYSQL database is used. Store and retrieve the details into/from Package\_Details table.  db.properties file is used to store the database configuration details. |

