Holiday Package Automation

Java Knock Out Challenge

Version 1.0

	Prepared By / Last Updated By	Reviewed By	Approved By
Name			
Role			
Signature			
Date			

Table of Contents

1.0 Introduction

1.1 Purpose of this document

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

1.2 Definitions & Acronyms

Definition / Acronym	Description
Nill	

1.3 Project Overview

This project captures the various concepts, techniques and skills learned and helps to put them into practice using Java with JDBC. Admittedly, this would be at a scaled-down level since the purpose is to let the associate experience the various concepts learned in Java as an individual. The individual associate is expected to carry out the knockout challenge and complete it within 4 hours.

1.4 Scope

The scope of the system is explained through its following modules

1. Parse data and calculate the package cost for all the packages
2. Extract the details of packages with minimum number of days.

1.5 Target Audience

Learner Level

1.6 Hardware and Software Requirements

1.6.1 Hardware Requirements

#	Item	Specification/Version

1.6.2 Software Requirements

#	Item	Specification/Version
1.	Java	8
2.	MYSQL	5.1

Note: All the required hardware and software is provided in the Tekstac platform

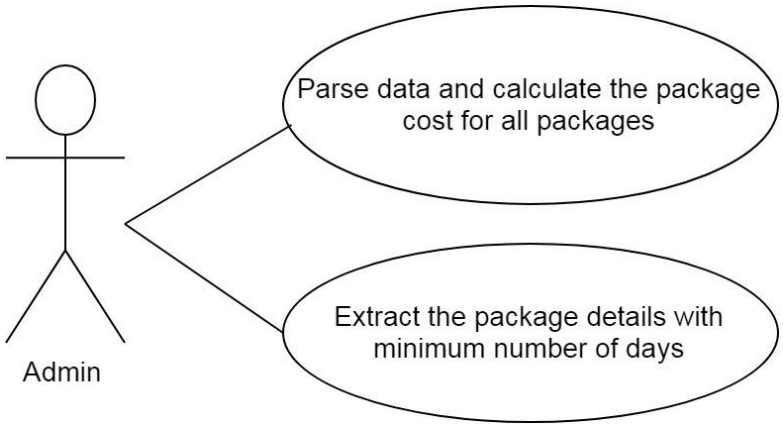
2.0 Functional Requirements

2.1 Functional Requirements

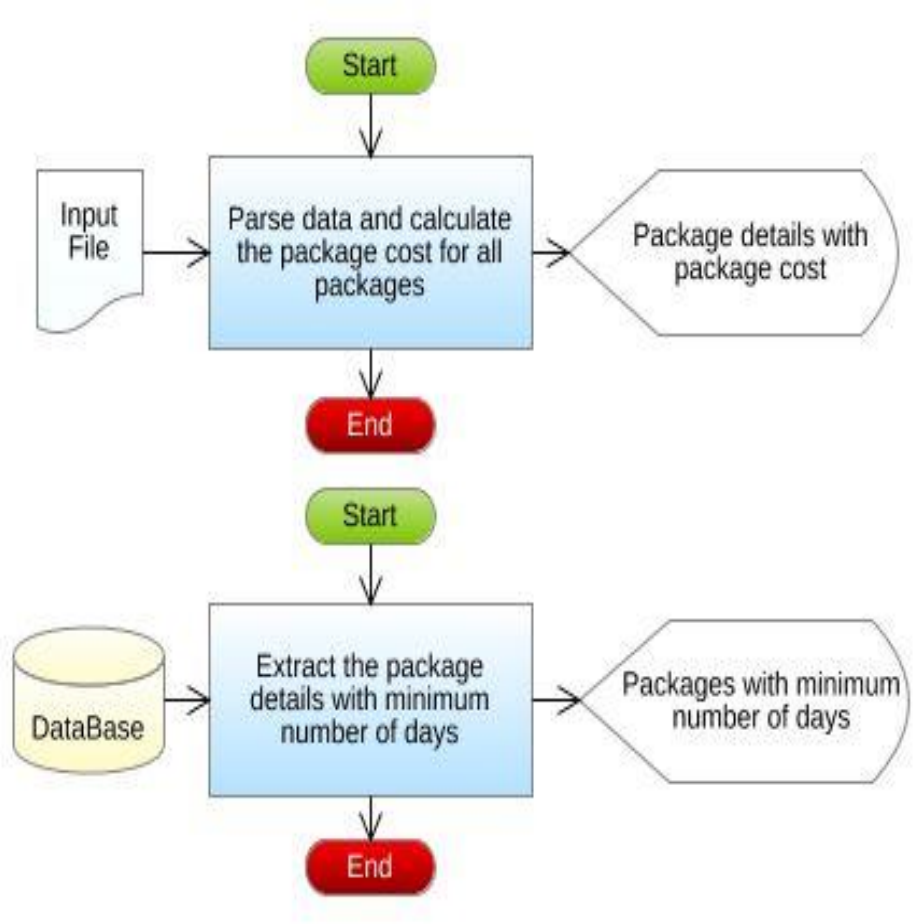
Req. #	Req. Name	Req. Description	Actors / Users	Comments
1	Parse data and calculate the package cost for all	The package details like package Id, source place, destination place, basic fare and number of days for each package are	Admin	The admin of Varsh Tours and Travels is responsible for parsing the data

	the packages	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.		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.

2.2 Use case Diagram



2.3 System Architecture Diagram



3.0 Design Specification

3.1 Data Design

Table Structure:

Table name: Package_Details	
Column Name	**Data type**
package_id	varchar(25)
source_place	varchar(25)
destination_place	varchar(25)
no_of_days	int
package_cost	int

Design Constraints:

- Use MYSQL database to store the data. The database name should be “HolidayPackage”.
- The above table has been already created. To create the table in your local machine, you can avail the script from “script.sql, which will be provided as part of the code skeleton.
- The table names and the column names should be the same as specified in the table structure.
- Database connections should be configurable; it should not be hard coded. The database information is specified in the “db.properties” file, which is also provided as part of the code skeleton.

Note: The code skeleton is made available in the Tekstac platform. Skeleton incudes the script file. If working with Eclipse IDE,Copy and paste the script inside the script file into MYSQL editor so that the database, table with the required records are created.

3.2 Component Design for identified Use cases

3.2.1 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 return the list. This functionality does not deal with DB Connectivity.

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	TravelAgency	List<Package> generatePackageCost (String filePath)	This method should take the file path as the argument, parse the data	

package cost for all the packages			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.

3.2.2 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.

Note: Script file containing the records are for implementing this requirement only. So copy and paste the records when working with Eclipse and then implement the requirement and test your code.

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)

Compone	**Type(Clas**	**Method**	**Responsibiliti**	**Resources**

nt Name	s)		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.

Note: When working with Eclipse, please change the values of db.classname ,db.url,db.username,db.password according to your MYSQL Configuration.

3.3 General Design Constraints

1. The attribute/method/class name should be correctly specified as given in the document.

2. Do not hardcode the database configuration details in the DBHandler class. Read it from the db.properties file.

4.0 Submission

4.1 Code submission instructions

1. Do not change the code skeleton given, as your code will be auto evaluated.

2. You can validate your solution against sample test cases during the assessment duration.

3. Your last submitted solution will be considered for detailed evaluation.

4. Ensure to submit the solution before the specified time limit. You will not be allowed to submit the solution once the mentioned time for the assessment is over.

5. **No Sample Input/Output is provided as part of this document. This means that you will not be evaluated for any of the presentation related Requirements. You are free to write your own code in the main , to invoke the business method to check its correctness. main is not taken for evaluation.**

5.0 Change Log

<table>
<tr><th></th><th colspan="4">Changes Made</th></tr>
<tr><td>V1.0.0</td><td colspan="4">Initial baseline created on <dd-Mon-yy> by <Name of Author></td></tr>
<tr><td rowspan="3">Vx.y.z</td><td colspan="4"><Please refer to the configuration control tool / change item status form if the details of changes are maintained separately. If not, the template given below needs to be followed></td></tr>
<tr><td>Section No.</td><td>Changed By</td><td>Effective Date</td><td>Changes Effected</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>