**1)HR TRAVELS**

**import java.util.Map;**

**public class TicketBooking {**

**private Map<string, string> bookingMap = new HashMap<>();</string, string>**

**public Map<string, string> getBookingMap() {</string, string>**

**return bookingMap;**

**}**

**public void setBookingMap(Map<string, string> bookingMap) {</string, string>**

**this.bookingMap = bookingMap;**

**}**

**// Write and implement the business requirements in this class**

**public void addTicketBookingDetails(String bookingNumber, String travelMode) {**

**bookingMap.put(bookingNumber, travelMode);**

**}**

**public List findBookingNumbersByTravelMode(String travelMode) {**

**List numberList = new ArrayList<>();**

**for (Map.Entry<string, string> entry : bookingMap.entrySet()) {</string, string>**

**String no = entry.getKey();**

**String mode = entry.getValue();**

**if (mode.equalsIgnoreCase(travelMode)) {**

**numberList.add(no);}}**

**return numberList;}}**

**2)VELVET VISTA**

**FLATINFO.CLASS**

**import java.util.ArrayList;**

**import java.util.HashMap;**

**import java.util.List;**

**import java.util.Map;**

**public class FlatInfo {**

**private Map<string, string> flatMap = new HashMap<>();</string, string>**

**public void addFlatDetails(String flatId, String blockName) {**

**flatMap.put(flatId.toLowerCase(), blockName.toLowerCase());**

**}**

**public List getTheFlatNumberByBlockName(String blockName) {**

**List flatIds = new ArrayList<>();**

**for (Map.Entry<string, string> entry : flatMap.entrySet()) {</string, string>**

**if (entry.getValue().equalsIgnoreCase(blockName)) {**

**flatIds.add(entry.getKey().toUpperCase());}}**

**return flatIds;}}**

**3)SERIOUSLY SWEET(CAKEORDER.INFO)**

**import java.util.HashMap;**

**import java.util.Map;**

**public class CakeOrder {**

**private Map<string, double> orderMap = new HashMap<>();</string, double>**

**public void addOrderDetails(String orderId, double cost) {**

**orderMap.put(orderId.toLowerCase(), cost);**

**}**

**public double findTotalCostOfOrdersInTheGivenRange(double minimumCost, double maximumCost) {**

**double totalCost = 0;**

**boolean found = false;**

**for (Map.Entry<string, double> entry : orderMap.entrySet()) {</string, double>**

**double cost = entry.getValue();**

**if (cost >= minimumCost && cost <= maximumCost) {**

**totalCost += cost;**

**found = true;**

**}**

**}**

**if (!found) {**

**return -1;**

**}**

**return totalCost;**

**}**

**}**

**4)EXPERT TALK MAANGMENT**

**import java.util.Set;**

**public class ExpertCatalog {**

**private Map<string, string> expertMap = new HashMap<>();</string, string>**

**public void addExpert(String expertDetails) {**

**String[] details = expertDetails.split(":");**

**String expertName = details[0];**

**String topic = details[1];**

**expertMap.put(expertName.toLowerCase(), topic.toLowerCase());**

**}**

**public Set getExpertsByTopic(String topic) {**

**Set experts = new HashSet<>();**

**for (Map.Entry<string, string> entry : expertMap.entrySet()) {</string, string>**

**if (entry.getValue().equalsIgnoreCase(topic)) {**

**experts.add(entry.getKey());**

**}**

**}**

**return experts;**

**}**

**}**

**5)PHONE DIRECTORY**

**import java.util.HashMap;**

**public class PhoneDirectorySystem {**

**private Map<string, string> contactDirectory = new HashMap<>();</string, string>**

**public void addContactRecord(String contactData) {**

**String[] details = contactData.split(":");**

**String name = details[0];**

**String phoneNumber = details[1];**

**contactDirectory.put(name.toLowerCase(), phoneNumber);**

**}**

**public Set getPhoneNumbers(String searchKeyword) {**

**Set phoneNumbers = new HashSet<>();**

**for (Map.Entry<string, string> entry : contactDirectory.entrySet()) {</string, string>**

**if (entry.getKey().contains(searchKeyword.toLowerCase())) {**

**phoneNumbers.add(entry.getValue());**

**}**

**}**

**return phoneNumbers;**

**}**

**}**

**6)**[**Movie Streaming Platform**](https://cognizant.tekstac.com/mod/vpl/view.php?id=32693)

**import java.util.ArrayList;**

**import java.util.List;**

**public class MovieManager {**

**private List movieList = new ArrayList<>();**

**public void addMovieDetails(String movieDetails) {**

**movieList.add(movieDetails);**

**}**

**public List getMoviesByRating(double minRating) {**

**List filteredMovies = new ArrayList<>();**

**for (String movie : movieList) {**

**String[] details = movie.split(":");**

**String title = details[0];**

**double rating = Double.parseDouble(details[1]);**

**if (rating >= minRating) {**

**filteredMovies.add(title);**

**}**

**}**

**return filteredMovies;**

**}**

**}**

**7)WILD LIFE SANTUARY**

**import java.util.HashMap;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Map;**

**public class WildlifeSanctuary {**

**private Map<string, string> sanctuaryRecords = new HashMap<>();</string, string>**

**public void addAnimal(String animalData) {**

**String[] details = animalData.split(":");**

**String animalName = details[0];**

**String species = details[1];**

**sanctuaryRecords.put(animalName.toLowerCase(), species.toLowerCase());**

**}**

**public List getAnimalsBySpecies(String species) {**

**List animals = new ArrayList<>();**

**for (Map.Entry<string, string> entry : sanctuaryRecords.entrySet()) {</string, string>**

**if (entry.getValue().equalsIgnoreCase(species)) {**

**animals.add(entry.getKey());}}**

**return animals;}}**