
Test Case Compliance

for

Travel in good health system

Version 1.0 approved

Prepared by
Abhishek Gandhi | 19CS10031
Shashvat Gupta | 19CS30042
Sajal Chhamunya | 19CS10051

Instructors :
Prof. Sourangshu Bhattacharya
Prof. Abir Das
Omprakash Chakraborty

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

18 March 2021

Input:

```
let m = Management.getInstance();
let c1 = new Customer("sajal", "sajalchh", "hello", "9293945535");
let c2 = new Customer("shaswat", "shaswat621", "kgpianboy", "9293945235");
let c3 = new Customer("abhishek", "gandhi", "baba", "91885665571")

let t1 = new Train("mumbai rajdhani", "12951", new Map<string, Time>(), new Map<string,
number>());
let t2 = new Train("Howrah Express", "12833", new Map<string, Time>(), new Map<string,
number>());

let s1 = new Station("Mumbai");
let s2 = new Station("Delhi");
let s3 = new Station("Kolkata");
let s4 = new Station("Raipur");
let s5 = new Station("Nagpur");
let s6 = new Station("Ahmedabad");
let s7 = new Station("Durg");
let s8 = new Station("Surat");
let s9 = new Station("Kota");

t1.addStation(s1, new Time(7, 53));
t1.addStation(s2, new Time(8, 35));
t1.addStation(s8, new Time(3, 31));
t1.addStation(s9, new Time(16, 30));

t2.addStation(s1, new Time(0, 15));
t2.addStation(s3, new Time(23, 59));
t2.addStation(s4, new Time(22, 55));
t2.addStation(s5, new Time(17, 45));
t2.addStation(s6, new Time(10, 5));
t2.addStation(s7, new Time(22, 10));

let r1 = new Restaurant("sajal dhaba", "dhaba", "123");
let r2 = new Restaurant("club mahindra", "mahindra", "123");
let r3 = new Restaurant("dominos", "domi", "123");
```

```
let r4 = new Restaurant("foodbuz", "food", "123");
let r5 = new Restaurant("swastam", "swami", "123");
let r6 = new Restaurant("Mio Amore", "mio_me", "123");
let r7 = new Restaurant("Arabian Nights", "nights", "123");
let r8 = new Restaurant("KFC", "kfc", "123");
```

```
m.updateRestarantStatus(r1, 1);
m.updateRestarantStatus(r2, 1);
m.updateRestarantStatus(r3, 1);
m.updateRestarantStatus(r4, 1);
m.updateRestarantStatus(r5, 1);
m.updateRestarantStatus(r6, 1);
m.updateRestarantStatus(r7, 1);
m.updateRestarantStatus(r8, 1);
```

```
let a1 = new Agent("ramesh", "ramesh", "123", r1);
let a2 = new Agent("ramu", "ramu", "123", r2);
let a3 = new Agent("mahesh", "mahesh", "123", r3);
let a4 = new Agent("ganesh", "ganesh", "123", r4);
let a5 = new Agent("sham", "sham", "123", r5);
let a6 = new Agent("shamu", "shamu", "123", r6);
let a7 = new Agent("lalit", "lalu", "123", r7);
```

```
r1.addStation(s1, new Time(0, 30));
r2.addStation(s2, new Time(0, 30));
r2.addStation(s3, new Time(0, 40));
r3.addStation(s4, new Time(0, 30));
r4.addStation(s5, new Time(0, 30));
r5.addStation(s6, new Time(0, 30));
r6.addStation(s7, new Time(0, 20));
r7.addStation(s8, new Time(0, 50));
r7.addStation(s9, new Time(0, 10));
r8.addStation(s1, new Time(0, 60));
r8.addStation(s2, new Time(0, 40));
r8.addStation(s5, new Time(0, 10));
```

```
r1.addItem("cheese fries", 600, 8);
r1.addItem("veggie potato burger", 60, 0);
r1.addItem("chicken burger", 60, 0);
r1.addItem("veg whopper", 60, 0);
r1.addItem("Mc veggie", 60, 0);
r1.addItem("coke", 60, 5);
r1.addItem("veg momos", 60, 4);
```

```
r2.addItem("veg rolls", 60, 2);  
r2.addItem("dahi chaat", 60, 3);  
r2.addItem("hara bhara kabab", 60, 12);  
r2.addItem("chicken rolls", 60, 2);
```

```
r3.addItem("Margerita", 60, 1);  
r3.addItem("Paneer pizza", 60, 1);  
r3.addItem("corn pizza", 60, 1);  
r3.addItem("lava cake", 60, 21);
```

```
r4.addItem("butterchoch cake", 260, 5);  
r4.addItem("chocolate cake", 560, 5);  
r4.addItem("pineapple cake", 600, 5);  
r4.addItem("oreo milkshake", 60, 13);
```

```
r5.addItem("samosa", 430, 6);  
r5.addItem("raj kajori", 450, 6);  
r5.addItem("allu tikki", 500, 6);  
r5.addItem("masala soda", 40, 22);
```

```
r6.addItem("Brownie", 60, 21);  
r6.addItem("coffee", 60, 22);  
r6.addItem("Chicken nuggets", 60, 20);  
r6.addItem("pastry", 60, 21);
```

```
r7.addItem("hot sour soup", 200, 15);  
r7.addItem("biryani", 60, 21);  
r7.addItem("chicken biryani", 360, 10);  
r7.addItem("green salad", 100, 14);
```

```
r8.addItem("chicken burger", 200, 0);  
r8.addItem("chicken noodles", 60, 9);  
r8.addItem("fish nuggets", 360, 17);  
r8.addItem("prawns", 700, 20);
```

All Testing Inputs are written in Unit_Testing.ts

Output

-----UnitTestingManagementAndAccount-----

0

Management

Manager

Password Verification working

Management

Checking Application Array: Found

updateRestaurantStatus : Working

-----UnitTestingTime-----

Correct initialization

LessthanequalWorking

update working fine

-----UnitTestingItems-----

Name Verified

Price Verified

item type Verified

Restaurant Verified

-----UnitTestingMenu-----

get price functionality working

Add Item functionality working properly

Remove functionality working properly

-----UnitTestingOrders-----

order allotment checked

Get order details working properly both in restaurant and order

get Receipt working properly

-----UnitTestingRestaurants-----

adding Item and Adding Station working properly

agent allotment working properly

-----UnitTestingAgents-----

order allotment working properly
restaurant allotment working properly

-----UnitTestingStations-----

[15, 22]

Add Station and get Station working

Add Item and get Item working

-----UnitTestingTrains-----

adding Station and return functions working properly

train number allotment correct

train name allotment correct