

Name:Yashica Patodia

Roll No:19CS10067

# MASTER TEST PLAN

## UNIT TEST SCENARIOS

### STATION CLASS

- Positive Test
  - Calling constructor and then GetName() function.
    - Input:Station s("Mumbai")  
cout<<s.GetName()
    - Golden Output:Mumbai
- Negative Test
  - Calling constructor with a blank string
    - Input:Station s("")
    - Golden Output:Invalid Station

### RAILWAYS CLASS

- Positive Test

- Adding distance between two stations in a hash map and the calling GetDistance() function
  - Input:Station s1("Mumbai");Station s2("Delhi")  
 Railways& IR = Railways::GetRailways();  
 IR.AddStation(s1); IR.AddStation(s2);IR.AddDist(s1, s2, 1447);double dist=IR.GetDistance(s1, s2);
  - Golden Output:1447
- Negative Test
  - Adding same station as input
    - INPUT:Station s1("Mumbai");Railways& IR = Railways::GetRailways(); IR.AddStation(s1); IR.AddStation(s1)
    - GOLDEN OUTPUT:Duplicate Station given as input
  - Adding distance between two stations again as input
    - INPUT:Station s1("Mumbai");Station s2("Delhi");Railways& IR = Railways::GetRailways(); IR.AddStation(s1); IR.AddStation(s2);IR.AddDist(s1, s2, 1447);IR.AddDist(s1, s2, 100);
    - GOLDEN OUTPUT:Distance between same station not stored

## PASSENGER CLASS

- Positive Test

- Calling Passenger constructor and creating a passenger object and calling GetGender()function
  - INPUT:Passenger p(string("123456789123"), Date(9, 9, 2002), Gender::Male::Type(),string ("1111111111"), string("abc"), string("def"), string("ghf"), Divyaang::Blind::Type(), string("123"));Gender g=p.GetGender();
  - Golden Output:Male
- Calling Passenger constructor and creating a passenger object and calling GetDisabilityID()function
  - INPUT:Passenger p(string("123456789123"), Date(9, 9, 2002), Gender::Male::Type(),string ("1111111111"), string("abc"), string("def"), string("ghf"), Divyaang::Blind::Type(), string("123"));Divyaan &d=p.GetDisabilityID();
  - Golden Output:Blind
- Calling Passenger constructor and creating a passenger object and calling GetAge()function
  - INPUT:Passenger p(string("123456789123"), Date(9, 9, 2002), Gender::Male::Type(),string ("1111111111"), string("abc"), string("def"), string("ghf"), Divyaang::Blind::Type(), string("123"));int age=p.GetAge();
  - Golden Output:18
- Calling Passenger constructor and creating a passenger object and calling GetDateOfBirth()function

- INPUT: Passenger p(string("123456789123"),  
Date(9, 9, 2002), Gender::Male::Type(), string  
("1111111111"), string("abc"), string("def"),  
string("ghf"), Divyaang::Blind::Type(),  
string("123")); Date &d=p.GetDateOfBirth();
- GOLDEN OUTPUT: 9/9/2002

## ● Negative Test

- Creating Passenger object with No First and last Name

- INPUT: Passenger p(string("123456789123"),  
Date(9, 9, 2002), Gender::Male::Type(), string  
("1111111111"), string(""), string("def"), string(""),  
Divyaang::Blind::Type(), string("123"));
- GOLDEN OUTPUT:

- Creating Passenger object with aadhar Number has other characters beside numbers

- INPUT: Passenger p(string("1234567891ab"),  
Date(9, 9, 2002), Gender::Male::Type(), string  
("1111111111"), string(""), string("def"), string(""),  
Divyaang::Blind::Type(), string("123"));
- GOLDEN OUTPUT:

- Creating Passenger object when mobile Number has other characters beside numbers

- INPUT: Passenger p(string("123456789112"),  
Date(9, 9, 2002), Gender::Male::Type(), string  
("1111111111a"), string(""), string("def"), string(""),  
Divyaang::Blind::Type(), string("123"));
- GOLDEN OUTPUT:

- Creating Passenger object when disability ID has other characters beside numbers
  - INPUT: Passenger p(string("123456789112"), Date(9, 9, 2002), Gender::Male::Type(), string("1111111111"), string(""), string("def"), string(""), Divyaang::Blind::Type(), string("12a"));
  - GOLDEN OUTPUT:
- Creating Passenger object with invalid Aadhar Number length .
  - INPUT: Passenger p(string("12"), Date(9, 9, 2002), Gender::Male::Type(), string("1111111111"), string(""), string("def"), string(""), Divyaang::Blind::Type(), string("123"));
  - GOLDEN OUTPUT:
- Creating Passenger object with invalid mobile Number length .
  - INPUT: Passenger p(string("122222222222"), Date(9, 9, 2002), Gender::Male::Type(), string("11"), string(""), string("def"), string(""), Divyaang::Blind::Type(), string("123"));
  - GOLDEN OUTPUT:

## GENDER CLASS

- Positive Test
  - Checking for isMale() function
    - INPUT: IsMale(Gender::Male::Type())
    - GOLDEN OUTPUT: True

- Negative Test
  - Checking for isMale() function
    - INPUT:IsMale(Gender::Female::Type())
    - GOLDEN OUTPUT: False

## DATE CLASS

- Positive Tests
  - Creating a Date object and checking isValidDate()
    - INPUT:Date d(18, 9, 2001); d.ValidDate()
    - GOLDEN OUTPUT:True
  - Creating a Date object and checking isLeap()
    - INPUT:Date d(18, 9, 2012); d.isLeap()
    - GOLDEN OUTPUT:True
  - Creating a Date object and checking < operator
    - INPUT:Date d1(18, 9, 2001);Date d2(9, 9, 2001);  
(d2<d1)
    - GOLDEN OUTPUT:True
- Negative Tests
  - Creating an invalid date
    - INPUT:Date d1(29, 2, 2001);
    - GOLDEN OUTPUT:
  - Creating invalid date with wrong date of month

- INPUT:Date d1(42, 8, 2001);
- GOLDEN OUTPUT:
- Creating invalid date with wrong year
  - INPUT:Date d1(1, 8,1500);
  - GOLDEN OUTPUT:

## DIVYAANG CLASS

- Positive Tests
  - Calling the GetName() function of Divyaang class
    - INPUT:Divyaang::Blind::Type().GetName()
    - GOLDEN OUTPUT:Blind
  - Calling the GetConcession() function of Divyaang Class
    - INPUT:Divyaang::CancerPat::Type().GetConcessionFactor(BookingClasses::ACFirstClass::Type())
    - GOLDEN OUTPUT:0.5

## CONCESSION CLASS

- Positive Tests
  - Creating a Passenger object and calling GetConcession() for Blind Type

■ **INPUT:** Passenger p(string("123456789123"),  
Date(9, 9, 2002), Gender::Male::Type(), string  
("1111111111"), string("abc"), string("def"),  
string("ghf"), Divyaang::Blind::Type(),  
string("123")); DivyaangConcession::Type().GetConce  
ssionFactor(pM,  
BookingClasses::ACFirstClass::Type(),  
Divyaang::Blind::Type())

■ **GOLDEN OUTPUT:** 0.5

- Creating a Passenger object and calling  
GetDisability() for Blind Type

■ **INPUT:** Passenger p(string("123456789123"),  
Date(9, 9, 2002), Gender::Male::Type(), string  
("1111111111"), string("abc"), string("def"),  
string("ghf"), Divyaang::Blind::Type(),  
string("123")); (DivyaangConcession::Type().IsEligible(  
pM, pM.GetDisabilty()))

■ **GOLDEN OUTPUT:** True

- Creating a Passenger object and calling  
isEligible() for concession

■ **INPUT:** Passenger p(string("123456789123"),  
Date(9, 9, 2002), Gender::Male::Type(), string  
("1111111111"), string("abc"), string("def"),  
string("ghf"), Divyaang::Blind::Type(),  
string("123")); GeneralConcession::Type().IsEligible()

■ **GOLDEN OUTPUT:** True

○

## BOOKING CLASS



- Positive Tests

- Creating a booking object and checking the ComputeFare() function

- INPUT: Passenger

- ```
pDivyaang(string("111111111111"), Date(9, 9, 2001),
Gender::Male::Type(), string ("99999999999"),
string("ABC"), string("DEF"), string("GHI"),
Divyaang::Blind::Type(),
string("123"))::CreateBooking( Station("Mumbai"),
Station("Kolkata"), Date(3, 3, 2021), Date(4, 5, 2021),
BookingClasses::AC3Tier::Type(),
BookingCategory::General::Type(), pDivyaang);
```

- GOLDEN OUTPUT:2558

## BOOKINGCLASSES CLASS

- Positive Tests

- Calling GetName() function

- INPUT:BookingClasses::AC3Tier::Type().GetName)

- GOLDEN OUTPUT:AC3Tier

- Calling GetLoadFactor() function

- INPUT:BookingClasses::AC3Tier::Type().GetLoadFactor()

- GOLDEN OUTPUT:2.5

- Calling IsSitting() function

- INPUT:BookingClasses::AC3Tier::Type().IsSitting()

- GOLDEN OUTPUT:False
- Calling IsAC() function
  - INPUT:BookingClasses::AC3Tier::Type().IsAC()
  - GOLDEN OUTPUT:True
- Calling IsLuxury() function
  - INPUT:BookingClasses::AC3Tier::Type().IsLuxury()
  - GOLDEN OUTPUT:False
- Calling GetNumberOfTiers() function
  - INPUT:BookingClasses::AC3Tier::Type().GetNumberOfTiers()
  - GOLDEN OUTPUT:3
- Calling GetReservationCharge() function
  - INPUT:BookingClasses::AC3Tier::Type().GetReservationCharge()
  - GOLDEN OUTPUT:40
- Calling GetTatkalLoadFactor() function
  - INPUT:BookingClasses::AC3Tier::Type().GetTatkalLoadFactor()
  - GOLDEN OUTPUT:0.3
- Calling GetTatkalMinCharge() function
  - INPUT:BookingClasses::AC3Tier::Type().GetTatkalMinCharge()
  - GOLDEN OUTPUT:300
- Calling GetTatkalMaxCharge() function

- INPUT:BookingClasses::AC3Tier::Type().GetTatkalMaxCharge()
- GOLDEN OUTPUT:400
- Calling GetTatkalMinDist() function
  - INPUT:BookingClasses::AC3Tier::Type().GetTatkalMinDist()
  - GOLDEN OUTPUT:500

## BOOKINGCATEGORY CLASS

- Positive Tests
  - Checking GetName() function of BookingCatgory
    - INPUT:BookingCategory::Ladies::Type().GetName()
    - GOLDEN OUTPUT:Ladies
  - Checking GenerateBooking function by creating a passenger object
    - INPUT:Passenger p(string("123456789123"), Date(9, 9, 2002), Gender::Male::Type(),string ("1111111111"), string("abc"), string("def"), string("ghf"), Divyaang::Blind::Type(), string("123"))BookingCategory::Ladies::Type().GenerateBooking(Station("Delhi"), Station("Mumbai"),Date(2, 3, 2021),Date(4, 5, 2021), BookingClasses::AC3Tier::Type, pF, true, "BOOKING SUCCEEDED", 0)

- GOLDEN OUTPUT:Booking Successful

# APPLICATION TESTING

The following booking are done to test the application of the software. It covers all the possible situations of booking. In all these booking ,bookingclasses and stations are rotated so that it covers all the cases

## Positive Tests

Checking every for every Booking Category:

Introduce Passengers of all kinds,

Divyaang Passenger:

- Passenger pDivyaang(string("111111111111"), Date(9, 9, 2001), Gender::Male::Type(), string

```
("9999999999"), string("ABC"), string("DEF"),  
string("GHI"), Divyaang::Blind::Type(), string("123"));
```

Ladies Passenger:

- Passenger pLadies(string("111111111111"), Date(9, 9, 2001), Gender::Male::Type(), string ("9999999999"), string("ABC"), string("DEF"), string("GHI"), Divyaang::Blind::Type(), string("123"));

Senior Citizen Passenger:

- Passenger pSeniorCitizen(string("111111111111"), Date(9, 9, 1955), Gender::Female::Type(), string ("9999999999"), string("ABC"), string("DEF"), string("GHI"), Divyaang::Blind::Type(), string("123"))

Now creating booking for all test scenarios

- General:
  - Booking::CreateBooking( Station("Mumbai"), Station("Kolkata"), Date(3, 3, 2021), Date(4, 5, 2021), BookingClasses::AC3Tier::Type(), BookingCategory::General::Type(), pDivyaang);

## **GOLDEN OUTPUT:**

**BOOKING SUCCEEDED**

**PNR Number = 1**

**Date of Reservation = 3/Mar/2021**

**Date of Booking= 4/May/2021**

**Booking Class = AC3Tier**

**From Station = Mumbai**

**To Station = Kolkata**

**: Comfort: AC**

**: Mode: Sleeping**

**: Luxury: No**

**: Bunks: 3**

**Travel Category = General**

**Fare = 2558**

■ Ladies:

- Booking::CreateBooking(Station("Delhi"), Station("Mumbai"), Date(3, 3, 2021), Date(4, 5, 2021), BookingClasses::AC2Tier::Type(), BookingCategory::Ladies::Type(), pLadies);

**GOLDEN OUTPUT:**

BOOKING SUCCEEDED

PNR Number = 2

Date of Reservation = 3/Mar/2021

Date of Booking= 4/May/2021

Booking Class = AC2Tier

From Station = Delhi

To Station = Mumbai

: Comfort: AC

: Mode: Sleeping

: Luxury: No

: Bunks: 2

Travel Category = Ladies

Fare = 2944

■ Senior Citizen:

- Booking::CreateBooking(Station("Kolkata"), Station("Chennai"), Date(2, 3, 2021), Date(4, 5, 2021), BookingClasses::FirstClass::Type(), BookingCategory::SeniorCitizen::Type(), pSenior);

**GOLDEN OUTPUT:**

BOOKING SUCCEEDED

PNR Number = 3

Date of Reservation = 3/Mar/2021  
Date of Booking= 4/May/2021  
Booking Class = FirstClass  
From Station = Kolkata  
To Station = Chennai  
: Comfort: Non-AC  
: Mode: Sleeping  
: Luxury: Yes  
: Bunks: 2  
Travel Category = Senior Citizen  
Fare = 1543

■ Divyaanjan:

- Booking::CreateBooking(Station("Bangalore"), Station("Delhi"), Date(2, 3, 2021), Date(4, 5, 2021), BookingClasses::ACFirstClass::Type(), BookingCategory::DivyaangCat::Type(), pDivyaang);

**GOLDEN OUTPUT:**

BOOKING SUCCEEDED

PNR Number = 4

Date of Reservation = 3/Mar/2021

Date of Booking= 4/May/2021

Booking Class = ACFirstClass

From Station = Bangalore

To Station = Delhi

: Comfort: AC

: Mode: Sleeping

: Luxury: Yes

: Bunks: 2

Travel Category = Divyaanjan

Fare = 3554

■ Tatkal:

- Booking::CreateBooking(Station("Kolkata"), Station("Bangalore"), Date(2, 3, 2021), Date(3, 3, 2021), BookingClasses::ExecutiveChairCar::Type(), BookingCategory::Tatkal::Type(), pLadies);

**GOLDEN OUTPUT:**

BOOKING SUCCEEDED

PNR Number = 5

Date of Reservation = 3/Mar/2021

Date of Booking= 3/Mar/2021

Booking Class = ExecutiveChairCar

From Station = Kolkata

To Station = Bangalore

: Comfort: AC

: Mode: Sitting

: Luxury: Yes

: Bunks: 0

Travel Category = Tatkal

Fare = 5238

■ Premium Takal:

- Booking::CreateBooking(Station("Mumbai"), Station("Chennai"), Date(2, 3, 2021), Date(3, 3, 2021), BookingClasses::Sleeper::Type(), BookingCategory::PremiumTatkal::Type(), pSenior);

**GOLDEN OUTPUT:**

BOOKING SUCCEEDED

PNR Number = 6

Date of Reservation = 3/Mar/2021



Date of Booking= 3/Mar/2021  
Booking Class = Sleeper  
From Station = Mumbai  
To Station = Chennai  
: Comfort: Non-AC  
: Mode: Sleeping  
: Luxury: No  
: Bunks: 3  
Travel Category = Premium Tatkal  
Fare = 1089

## NEGATIVE TESTS

- Booking Failing due to invalid station name

### INPUT:

```
Booking::CreateBooking(Station("Mumbai"),Station("Hyder  
abad"),Date(2, 3, 2021),Date(4, 9,  
2021),BookingClasses::ACChairCar::Type(),BookingClasse  
s::ACChairCar::Type(),BookingCategory::SeniorCitizen::Ty  
pe(),pSenior);
```

### GOLDEN OUTPUT:

Booking Station invalid  
Booking Category invalid  
Station invalid  
BOOKING FAILED  
PNR Number = 7  
Date of Reservation = 2/Mar/2021  
Date of Booking= 4/May/2021  
Booking Class = SecondSitting  
From Station = Mumbai  
To Station = Hyderabad

: Comfort: Non-AC  
: Mode: Sitting  
: Luxury: No  
: Bunks: 0

Travel Category = Senior Citizen  
Fare = 15

- Booking failing due to same station booking

**INPUT:**

```
Booking::CreateBooking(Station("Mumbai"),Station("Mumbai"),Date(2, 3, 2021),Date(4, 9, 2021),BookingClasses::AC3Tier::Type(),BookingClasses::ACChairCar::Type(),BookingCategory::Ladies::Type(),pDivyang);
```

**GOLDEN OUTPUT:**

Booking to same station not possible  
Booking Category invalid  
Distance between same station not stored  
BOOKING FAILED  
PNR Number = 8  
Date of Reservation = 2/Mar/2021  
Date of Booking= 4/May/2021  
Booking Class = AC3Tier  
From Station = Mumbai  
To Station = Mumbai  
: Comfort: AC  
: Mode: Sleeping  
: Luxury: No  
: Bunks: 3

Travel Category = Ladies  
Fare = 40

- Booking failing due to invalid booking date

**INPUT:**

```
Booking::CreateBooking(Station("Kolkata"),Station("Mumbai"),Date(2, 3, 2021),Date(4, 2, 2021),BookingClasses::ACChairCar::Type(),BookingCategory::SeniorCitizen::Type(),pSenior);
```

**GOLDEN OUTPUT:**

Booking Date Invalid

BOOKING FAILED:

Booking Date Invalid

PNR Number = 9

Date of Reservation = 2/Mar/2021

Date of Booking= 4/Feb/2021

Booking Class = ACChairCar

From Station = Kolkata

To Station = Mumbai

: Comfort: AC

: Mode: Sitting

: Luxury: No

: Bunks: 0

Travel Category = Senior Citizen

Fare = 1248



