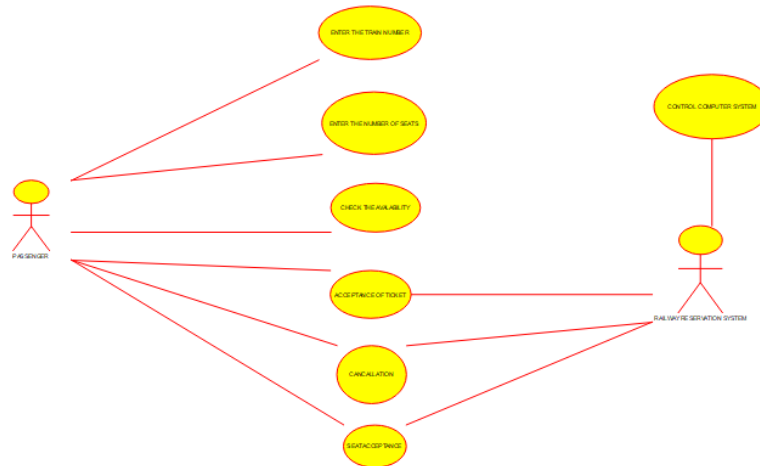
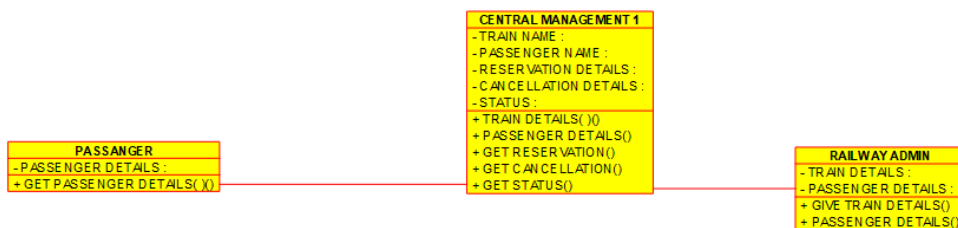


F –ticketing

Usecase:



Class diagram:



Code generate:

```

    /**
    */
    void GIVE_TRAIN_DETAILS ()
    {
    }

    /**
    */
    void PASSENGER_DETAILS ()
    {
    }

protected:

    // Static Protected attributes
    //

    // Protected attributes
    //

public:

    // Protected attribute accessor methods
    //

protected:

public:

    // Protected attribute accessor methods
    //

protected:
```

```

1
2 #ifndef RAILWAY_ADMIN_H
3 #define RAILWAY_ADMIN_H
4
5 #include <string>
6 #include vector
7
8
9
10 /**
11  * class RAILWAY_ADMIN
12  *
13  */
14
15 class RAILWAY_ADMIN
16 {
17 public:
18
19     // Constructors/Destructors
20     //
21
22
23     /**
24     * Empty Constructor
25     */
26     RAILWAY_ADMIN ();
27
28     /**
29     * Empty Destructor
30     */
31     virtual ~RAILWAY_ADMIN ();
32
33     // Static Public attributes
34     ..

```

```
49  /**
50  */
51  void GET_PASSENGER_DETAILS_ ()
52  {
53  }
54
55  protected:
56
57  // Static Protected attributes
58  //
59
60  // Protected attributes
61  //
62
63  public:
64
65
66  // Protected attribute accessor methods
67  //
68
69  protected:
70
71  public:
72
73
74  // Protected attribute accessor methods
75  //
76
77  protected:
78
79
80  private:
81
82  // Static Private attributes
83  //
84
```

```
1
2 #ifndef PASSANGER_H
3 #define PASSANGER_H
4
5 #include <string>
6 #include vector
7
8
9
10 /**
11  * class PASSANGER
12  *
13  */
14
15 class PASSANGER
16 {
17 public:
18
19     // Constructors/Destructors
20     //
21
22
23     /**
24     * Empty Constructor
25     */
26     PASSANGER ();
27
28     /**
29     * Empty Destructor
30     */
31     virtual ~PASSANGER ();
32
33     // Static Public attributes
34     //
35
36     // Public attributes
```