**TOTAL SCREENS IN THE PROJECT**

Overall , the project "**Railway Reservation System**" , will have following screens:

**1. Main Screen**

**2. View Trains**

**3. Book Ticket**

**4. View Ticket**

**5. Search Ticket No**

**6. View All Bookings**

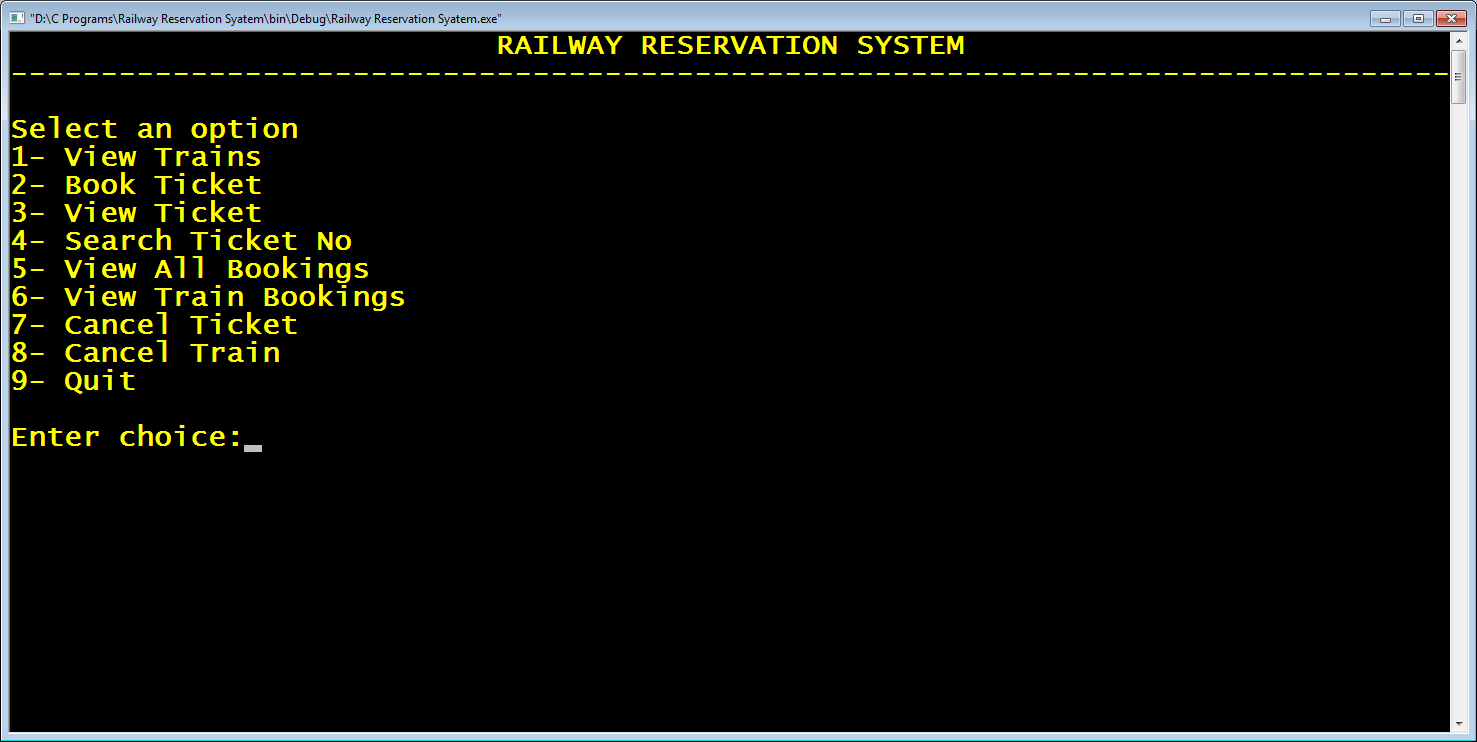
**7. View Train Bookings**

**8. Cancel Ticket**

**9. Cancel Train**

**THE MAIN SCREEN**

**============================================================================================**

****

**STEPS TO BE DONE BEFORE THE MainScreen**

Before we design the **Main Screen** , we need to do the following:

1. Add support of **conio2.h** in our project

2. Create a header file called "**rlyres.h**" which will contain **structure declarations**, **typedef statements** and **function declarations**

3.Create a source code file called "**rlyres.c**" which will contain definitions of the functions declared in the file "**rlyres.h**"

**ADDING "conio2.h" SUPPORT**

1. Create the project "**Railway Reservation System**" in CODEBLOCKS

2. Download the file **conio21.zip** from the url **https://sourceforge.net/projects/conio/**

3. Unzip the file and copy 2 files "**conio.c**" and "**conio2.h**" from it into your project folder

4. Now we can call all **BORLAND style** functions like **clrscr()**,**getch()**, **textcolor()**,**gotoxy()** etc in our project

**CREATING "rlyres.h" HEADER FILE**

The next thing we have to do is to create a header file called "**rlyres.h**" containing all the **structure** and **function declarations**.

**STRUCTURE DECLARATIONS IN "rlyres.h" HEADER FILE**

Our project will have 2 structures called "**TRAIN**" and "**PASSENGER**" .

The structure **TRAIN** will have following data members:

**train\_no** :- **string** (for storing train number)

**from**:- **string** (for storing FROM station code)

**to**:- **string**(for storing TO station code)

**fac\_fare**:- **int** (for storing fare amount of FIRST AC)

**sac\_fare**:- **int**(for storing fare amount of SECOND AC)

Following is it's declaration

***struct Train***

***{***

***char train\_no[10];***

***char from[10];***

***char to[10];***

***int fac\_fare;***

***int sac\_fare;***

***};***

The structure **PASSENGER** will have following data members:

**p\_name** :- **string** (for storing name of the passenger)

**gender**:- **char** (for storing gender of the passenger as **M** or **F**)

**train\_no**:- **string**(for storing train number)

**p\_class**:- **char** (for storing the **F** for FIRST AC and **S** for SECOND AC)

**address**:- **string**(for address of the passenger)

**age**:**int** (for storing passenger's age)

**ticketno**:- **int**(for storing ticket number)

**mob\_no**:-**string**(for storing passenger's mobile number)

Following is it's declaration

***struct Passenger***

***{***

***char p\_name[20];***

***char gender;***

***char train\_no[20];***

***char p\_class;***

***char address[30];***

***int age;***

***int ticketno;***

***char mob\_no[11];***

***};***

To make the coding easier we also have to **typedef** these structures as shown below:

***typedef struct Train Train;***

***typedef struct Passenger Passenger;***

**FUNCTION DECLARATIONS IN "rlyres.h" HEADER FILE**

Our project will have **21 functions** including the **main()** function. Initially we will declare **10 functions**

Following are their names and purposes:

**1. enterchoice()** : For displaying available options to the user and returning the choice entered by him/her to the **main( )** function

**Prototype:**

**int enterchoice()**

**2. add\_trains()** : For adding some train details to a file

**Prototype:**

**void add\_trains()**

**3. view\_trains()** : For displaying details of all the trains from the file

**Prototype:**

**void view\_trains()**

**4. book\_ticket()** : For accepting a **PASSENGER** variable as argument and booking his ticket . If booking is successful it will return the **ticket no** , otherwise it will return -1. It also saves the booked ticket to a FILE.

**Prototype:**

**int book\_ticket(Passenger)**

**5. get\_ticket\_no()** : For accepting a passenger's mobile number as argument and returning all his ticket nos. If record is not present it returns NULL.

**Prototype:**

**int\* get\_ticket\_no(char \*)**

**6. view\_ticket()** : For accepting a ticket number as argument and displaying the ticket details.

**Prototype:**

**void view\_ticket(int)**

**7. view\_all\_bookings()** : For displaying all the booked tickets.

**Prototype:**

**void view\_all\_bookings( )**

**8. view\_booking()** : For accepting a train number as argument and displaying the ticket booked for that train.

**Prototype:**

**void view\_bookings(char \*)**

**9. cancel\_ticket()** : For accepting a ticket number as argument and cancelling the ticket and if the ticket is cancelled successfully it returns 1 otherwise it returns 0

**Prototype:**

**int cancel\_ticket(int)**

**10. cancel\_train()** : For accepting a train number and deleting all the tickets of that train. If the cancellation is done successfully it returns 1 otherwise it returns 0

**Prototype:**

**int cancel\_train(char \*)**