VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

A Mini Project Report on

Travel Management System Using Indexing

Submitted in partial fulfillment of the requirements as a part of the File Structures Lab ofVI semester for the award of degree of ***Bachelor of Engineering*** *in* ***Information Scienceand Engineering****, Visvesvaraya Technological University, Belagavi*

Submitted by

M B BHAVANA	**1RN19IS082**
RAKSHITA S	**1RN19IS116**

Faculty In-charge

Dr. Prakasha S

Associate Professor

Dept. of ISE, RNSIT

ESTD : 2001

An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, R R Nagar Post

Bengaluru – 560 098

2021 – 2022

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,

Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

ESTD : 2001

An Institute with a Difference

CERTIFICATE

This is to certify that the Mini project report entitled ***TRAVEL MANAGEMENT SYSTEM USING INDEXING*** has been successfully completed by **M B BHAVANA** bearing USN **1RN19IS082** and **RAKSHITA S** bearing USN **1RN19IS116** presently VI semester students of **RNS Institute of Technology** in partial fulfillment of the requirements as a part of the **File Structure Laboratory** for the award of the degree of ***Bachelor of Engineering in Information Science and Engineering*** under **Visvesvaraya Technological University, Belagavi** during academic year **2021 – 2022.**

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of File structure Laboratory for the said degree.

Dr. Prakasha S	**Dr. Suresh L**
Faculty In charge	Professor and
Associate Professor	Head of Department
Dept. of ISE, RNSIT	Dept. of ISE, RNSIT

External Viva

Name of the Examiners	**Signature with date**
1. ______________________	______________________
2. ______________________	______________________

ABSTRACT

File Structures is the Organization of Data in Secondary Storage Device in such a way that it minimizes the access time of the information and the storage space. A File Structure is a combination of representations for data in files and of operations for accessing the data. A File Structure allows applications to read, write and modify data. It might also support finding the data that matches some search criteria or reading through the data in some particular order. The type of data we can store within the file system is Char, Int, Double, Float, and Boolean.

The main objective of the project travelling agency is to make avail to the customers all sorts of travelling services. A host of services such as registration, display, search, modify etc. are provided. In the registration step, the client has to provide his personal details. In the option of display all the client information is read like name, phone, cost etc. in the search tab, if information of a particular client is required, then that be obtained.

In the modify option, customer details can be changed or updated. In the delete option, record of the particular client such as his name, address and other details are deleted from the database. Modifications are also made to execute additional tasks. Also the project canbe extended to provide number of places and availability of vehicles.

In this project "Travelling agency a C++ Project" facilities like registration, search, display,modification, delete etc. are provided. The software searches the client data in the database it has created. Moreover the software designed here is used to arrange travel services, givenew and improved services and also identify travel related cost savings.

Travelling Agency a C++ Project
Travel agencies have to deal with a lot of customers daily. Hence it is very important to maintain records of all the customers. For this purpose we require software that makes it easy to maintain the customer details. The records of various customers can be stored in a single file. This software can be used in large number of travelling agencies where in records of their trips can be added. Travelling-Agency-a-C++-Project

ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out this Project work.

We would like to express our thanks to our Principal **Dr. M. K. Venkatesha** for his support and inspired us towards the attainment of knowledge.

We wish to place on record our words of gratitude to **Dr. Suresh L,** Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind our Project work.

We would like to express our profound and cordial gratitude to our Faculty In-charge, **Dr. Prakasha S**, Associate Professor, Department of Information Science and Engineering for his valuable guidance in preparing Project report.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped us to carry out the project work.

And lastly, we would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in carrying out this Project work.

M B BHAVANA	**(1RN19IS082)**
RAKSHITA S	**(1RN19IS116)**

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

AVL	- Adelson-Velskii and Landis
GUI	- Graphic User Interface
RAM	- Random Access Memory

Chapter 1

INTRODUCTION

1.1 Introduction to File Structures

File Structure is a combination of representations of data in files and operations for accessing the data. It allows applications to read, write and modify data. It supports finding the data that matches some search criteria or reading through the data in some particular order.

1.1.1 History

Early work with files presumed that files were on tape, since most files were. Access was sequential and the cost of access grew in direct proportion to the size of the file. As files grew intolerably large for unaided sequential access and as storage devices such as hard disks became available, indexes were added to files.

The indexes made it possible to keep a list of keys and pointers in a smaller file that could be searched more quickly. With key and pointer, the user had direct access to the large, primary file. But simple indexes had some of the same sequential flaws as the data file, and as the indexes grew, they too became difficult to manage, especially for dynamic files in which the set of keys changes. In the 19th century there were moves to codify indexing.

The Index Society was formed in London in 1877 with the aim of creating 'a general index of universal literature'. Dr. Henry Benjamin Wheatley, after whom the Wheatley Medal is named, wrote what is an indexer? In 1878. This society continued until 1890. Soon afterwards women began to enter the field. Eventually the Society of Indexers was formed in Great Britain in 1957.

The user can also store the trips and modify the trips as their need. The database contains two tables, one for storing user info and another one store trip info. Hashing is a good way to get what we want with a single request, with files that do not change size greatly over time. Hashed indexes were used to provide fast access to files. But until recently, hashing did not work well with volatile, dynamic files. Extendible dynamic hashing can retrieve information with 1 or 2 disk accesses, no matter how big the file is.

1.1.2 About the File

There is one important distinction that in file structures and that is the difference between the logical and physical organization of the data. On the whole a file structure will specify the logical structure of the data that is the relationships that will exist between data items independently of the way in which these relationships may actually be realized within any computer. It is this logical aspect that we will concentrate on.

The physical organization is much more concerned with optimizing the use of the storage medium when a particular logical structure is stored on, or in it. Typically for every unit of physical store there will be a number of units of the logical structure (probably records) to be stored in it.

A file system consists of two or three layers. Sometimes the layers are explicitly separated, and sometimes the functions are combined. The logical file system is responsible for interaction with the user application.

It provides the application program interface (API) for file operations- open, close, read etc., and passes the requested operation to the layer below it for processing. The logical file system "manages open file table entries and per-process file descriptors. This layer provides file access, directory operations, and security and protection.

The second optional layer is the virtual file system. "This interface allows support for multiple concurrent instances of physical file systems, each of which is called a file system implementation.

The third layer is the physical file system. This layer is concerned with the physical operation of the storage device (e.g. disk). It processes physical blocks being read or written. It handles buffering and memory management and is responsible for the physical placement of blocks in specific locations on the storage medium. The physical file system interacts with the device drivers or with the channel to drive the storage device.

1.1.3 Various Kinds of storage of Fields and Records

A field is the smallest, logically meaningful, unit of information in a file.

Field Structures

The four most common methods as shown in Fig. 1.1 of adding structure to files to maintain the identity of fields are:

- Force the fields into a predictable length.
- Begin each field with a length indicator.
- Place a delimiter at the end of each field to separate it from the next field.
- Use a “keyword=value” expression to identify each field and its contents.

Method 1: Fix the Length of Fields

In the above example, each field is a character array that can hold a string value of some maximum size. The size of the array is 1 larger than the longest string it can hold. Simple arithmetic is sufficient to recover data from the original fields.

The disadvantage of this approach is adding all the padding required to bring the fields up to a fixed length, makes the file much larger. We encounter problems when data is too long to fit into the allocated amount of space. We can solve this by fixing all the fields at lengths that are large enough to cover all cases, but this makes the problem of wasted space in files even worse. Hence, this approach isn’t used with data with large amount of variability in length of fields, but where every field is fixed in length if there is very little variation in field lengths.

Method 2: Begin Each Field with a Length Indicator

We can count to the end of a field by storing the field length just ahead of the field. If the fields are not too long (less than 256 bytes), it is possible to store the length in a single byte at the start of each field. We refer to these fields as length-based.

Method 3: Separate the Fields with Delimiters

We can preserve the identity of fields by separating them with delimiters. All we need to do is choose some special character or sequence of characters that will not appear within a field and then insert that delimiter into the file after writing each field. White-space characters (blank, new line, tab) or the vertical bar character, can be used as delimiters.

Method 4: Use a "Keyword=Value" Expression to Identify Fields

This has an advantage the others don’t. It is the first structure in which a field provides information about itself. Such self-describing structures can be very useful tools for organizing files in many applications. It is easy to tell which fields are contained in a file even if we don't know ahead of time which fields the file is supposed to contain. It is also a good format for dealing with missing fields. If a field is missing, this format makes it obvious, because the keyword is simply not there. It is helpful to use this in combination with delimiters, to show division between each value and the keyword for the following field. But this also wastes a lot of space: 50% or more of the file's space could be taken up by the keywords.

Record Structures

The five most often used methods for organizing records of a file are

- Require the records to be predictable number of bytes in length.
- Require the records to be predictable number of fields in length.
- Begin each record with a length indicator consisting of a count of the number of bytes that the record contains.
- Use a second file to keep track of the beginning byte address for each record.
- Place a delimiter at the end of each record to separate it from the next record.

Method 1: Make the Records a Predictable Number of Bytes (Fixed-Length Record)

A fixed-length record file is one in which each record contains the same number of bytes. In the field and record structure shown, we have a fixed number of fields, each with a predetermined length, that combine to make a fixed-length record.

Fixing the number of bytes in a record does not imply that the size or number of fields in the record must be fixed. Fixed-length records are often used as containers to hold variable numbers of variable-length fields. It is also possible to mix fixed and variable-length fields within a record.

Method 2: Make Records a Predictable Number of Fields

Rather than specify that each record in a file contains some fixed number of bytes, we can specify that it will contain a fixed number of fields.

Method 3: Begin Each Record with a Length Indicator

We can communicate the length of records by beginning each record with a filed containing an integer that indicates how many bytes there are in the rest of the record. This is commonly used to handle variable-length records.

Method 4: Use an Index to Keep Track of Addresses

We can use an index to keep a byte offset for each record in the original file. The byte offset allows us to find the beginning of each successive record and compute the length of each record. We look up the position of a record in the index, then seek to the record in the data file.

Method 5: Place a Delimiter at the End of Each Record

It is analogous to keeping the fields distinct. As with fields, the delimiter character must not get in the way of processing. A common choice of a record delimiter for files that contain readable text is the end-of-line character (carriage return/ new-line pair or, on Unix systems, just a new line character: „\n".

1.1.4 Application of File Structure

Relative to other parts of a computer, disks are slow. 1 can pack thousands of megabytes on a disk that fits into a notebook computer. The time it takes to get information from even relatively slow electronic random-access memory (RAM) is about 120 nanoseconds. Getting the same information from a typical disk takes 30 milliseconds. So, the disk access is a quarter of a million times longer than a memory access. Hence, disks are very slow compared to memory.

On the other hand, disks provide enormous capacity at much less cost than memory. They also keep the information stored on them when they are turned off. Tension between a disk's relatively slow access time and its enormous, nonvolatile capacity, is the driving force behind file structure design.

Chapter 2

SYSTEM ANALYSIS

Systems analysis is the process of observing systems for troubleshooting or development purposes. It is applied to information technology, where computer-based systems require defined analysis according to their makeup and design.

2.1 Analysis of the Project

The application deals with the Travel System, Structure are used to store the fields and records. The main objective of the project Traveling agency is to make avail to the customers all sorts of travelling services. A host of services such as registration, display, search, modify etc. are provided. In the registration step, the client has to provide his personal details. In the option of display all the client information is read like name, phone, cost etc. in the search tab, if information of a particular client is required, then that shall be obtained.

2.2 Operations Performed on a File

- **Insertion of a record into a file**

  Insertion function adds new records into the file. In this process the inserted node descends to the leaf where the key fits. If the node has an empty space, insert the key/reference pair into the node. If the node is already full, split it into two nodes, distributing the keys evenly between the two nodes.

- **Display of the records in the file**

  The display function provides information about the kind of values that are the resultof executing a statement or expression. This information is useful for understandinghow a program or script works. For example, this statement assigns a uint8 vector of values 1 2 3 4 to the variable named a.

- **Deletion of a record from a file**

  Deletion is the process in which on the parent node to remove the split key that previously separated these merged nodes unless the parent is the root and we are removing the final key from the root, in which case the merged node becomes the new root.

- **Modification of records in a file**

The system can also be used to modify existing records from all productioninventory details. The user is prompted for a key, which is used as to search the records. Once the record is found we rewrite details of the corresponding product Hence modify () is delete () followed by the read (). If the record is not found then it returns -1 value and display 'RECORD DOES NOT EXISTS' message on the console.

2.3 Technique Used

Indexing is the way to get an unordered table into an order that will maximize the query's efficiency while searching. When a table is un-indexed, the order of the rowswill likely not be discernible by the query as optimized in any way, and your query will therefore have to search through the rows linearly.

In other words, the queries will have to search through every row to find the rows matching the conditions. As you can imagine, this can take a long time. Looking through every single row is not very efficient.

B-Tree is a self-balancing search tree. In most of the other self-balancing search trees (like AVL and Red-Black Trees), it is assumed that everything is in main memory. To understand the use of B-Trees, we must think of the huge amount of data that cannot fit in main memory. When the number of keys is high, the data is read from disk in the form of blocks. Disk access time is very high compared tothe main memory access time. The main idea of using B-Trees is to reduce the number of disk accesses. Most of the tree operations (search, insert, delete, max, min, etc) require O(h) disk accesses where h is the height of the tree.

Chapter 3

SYSTEM DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces. Systems design is the process of defining the architecture, modules, interfaces, and data fora system to satisfy specified requirements. Systems design could be seen as the applicationof systems theory to product development.

3.1 Design of the Fields and records

We use variable length record and fixed size fields. In variable length record, it will take the minimum space need for that field and at the end of the record there will be a delimiter placed indicating that it is the end of the record. The fields are separated using a delimiter ('|') whereas hash ('#') denotes end of a record.

A delimiter is a sequence of one or more characters used to specify the boundary between separate, independent regions in plain text or other data streams. In this application, there are 3 files which stores the data they are:

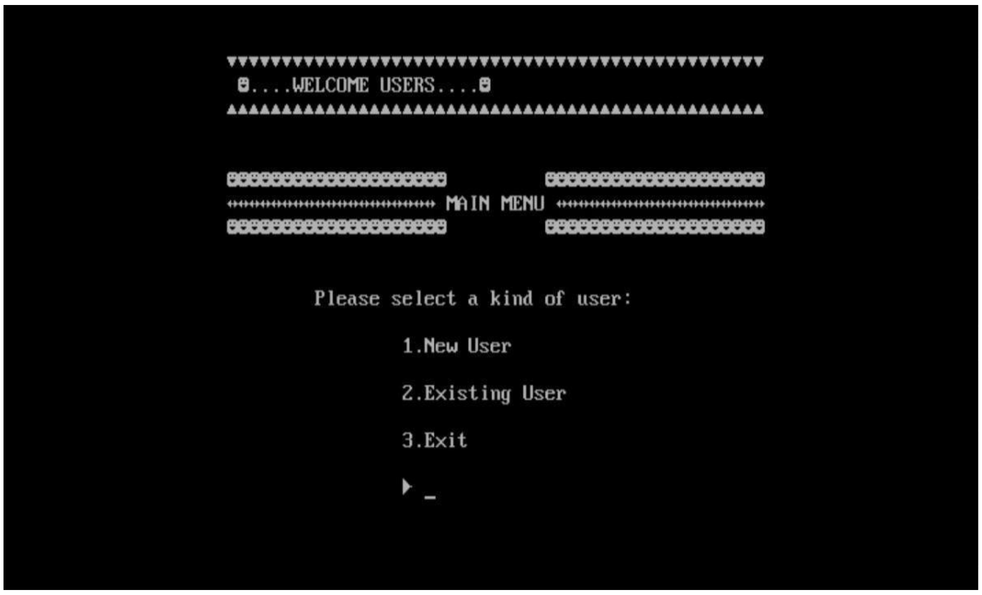
- Transactions File
- PersonalDetails File
- TravelDetails File

These files have respective attributes which identifies the uniqueness of each record.

- Transactions File: It consists of the all the transaction details that are made by the customer while receiving or issuing the book from/to Librarian. Attributes which identifies the uniqueness of each record are ID, Name, Issued-date, Returned date.
- PersonalDetails File: It consists of all the information of the customer who are registeredwith the Library. The attributes which identifies the uniqueness of each record are ID, Name, Email and Phone.
- TravelDetails File: It consists of information of the book which are available in the libraryand as well as the book details such as author name, genre. Attributeswhich denotesits uniqueness of each record are ID, Name, Author and Book- available.

3.2 User Interface

The user interface (UI), in the industrial design field of human–computer interaction, is the space where interactions between humans and machines occur. The goal of this interaction is to allow effective operation and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operators' decision-making process.



```
▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
 ☻....WELCOME USERS....☻
▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲


☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻          ☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻
↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔ MAIN MENU ↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔
☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻          ☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻


        Please select a kind of user:

                1.New User

                2.Existing User

                3.Exit

                ► _
```

Figure 3.1 User Interface

3.2.1 Insertion of a record

If the operation desired is Insertion that is to the read the data, the user is required to enter respective option new user/ existing as his/her choice, from the menu displayed, after which a new screen is displayed. For the personal details, the user is prompted to enter the followed by the Name, Address, Contact number, No. of people traveling, member details, First name, Age and Gender.

For the travel details entry, the user is prompted to enter the boarding point, destination, date of departure, deck, cabin, swimming pool, Gym, sports, salon, spa, Theatre, and there is a Back option to return back to main menu.

For Existing user, the user is prompted to enter the given ID that was previously issued in the journey. After all the values are accepted it is then stored in the data file and the user can press back option to return back to main menu.

3.2.2 Display of a record

The records are displayed based on the user options, which can be later viewed by the user. The record is displayed in various ways. The personal details and the travelling details are stored of each individual which is further helpful in modification and deletion of a record. One way is to display one record with all the details together. This option displays the records in sorted order. The other option is to display the record in the form of list under common header containing different fields. Also, various levels of the records are displayed separately.

The display function provides information about the kind of values that are the result of executing a statement or expression. This information is useful for understanding how a program or script works. Here, the display option shows the contents that are stored under personal and travel details such as name, contact number, address, gender, age , member details, whereas travel details consists of location, date of departure, deck, cabins etc.

3.2.3 Deletion of a record

If the operation desired is Deletion that is to the delete the data, the user is required to enter respective option personal or travel as his/her choice, from the menu displayed, after which a new screen is displayed. Next, the user is prompted to enter the new user or existing user, whose record from the file is to be deleted. If there are no records in the file, an “Error while opening file” message is displayed, and the user is prompted to press back key to return back to the menu screen.

Here, this delete function works as for example the no. of people travelling is four if one member data has to be deleted then his/her data shall be removed from the record reading the message “Error opening the file” .In each case, the user is then prompted to press any key to return back to the menu screen.

3.2.4 Modification of a record

If the operation desired is Modify, the user is required to enter respective option personal / travel details as his/her choice, from the menu displayed, after which a new screen is displayed.The Modify operation is implemented as a deletion followed by an insertion.

The system can also be used to modify existing records from all production inventory details. The user is prompted for a key, which is used as to search the records. Once the record is found we rewrite details of the corresponding product Hence modify () is delete () followed by the read (). MODIFY RECORD loads the record, if it is not already loaded for the current process, and displays the current input form. If there is no current record, then MODIFY RECORD does nothing. MODIFY RECORD does not affect the current selection.

Here, the modify option enables editing of a record suppose a contact number of a user has to be changed then the modify option will help the user to modify/ edit the information that was previously recorded by the user.

Chapter 4

IMPLEMENTATION

Implementation is the process of defining how the system should be built, ensuring that it is operational and meets quality standards. It is a systematic and structured approach for effectively integrating a software-based service or component into the requirements of end users.

4.1 About C++

C++ is a general-purpose object-oriented programming (OOP) language, developed by Bjarne Strousrup, and is an extension of the C language. It is therefore possible to code C++ in a "C style" or "object-oriented style." In certain scenarios, it can be coded in either way and is thus an effective example of a hybrid language.

C++ is considered to be an intermediate-level language, as it encapsulates both high- and low-level language features. Initially, the language was called "C with classes" as it had all the properties of the C language with an additional concept of "classes." However, it was renamed C++ in 1983.

The main highlight of C++ is a collection of predefined classes, which are data types that can be instantiated multiple times. The language also facilitates declaration of user-defined classes. Classes can further accommodate member functions to implement specific functionality. Multiple objects of a particular class can be defined to implement the functions within the class. Objects can be defined as instances created at run time. These classes can also be inherited by other new classes which take in the public and protected functionalities by default.

C++ includes several operators such as comparison, arithmetic, bit manipulation and logical operators. One of the most attractive features of C++ is that it enables the overloading of certain operators such as addition. A few of the essential concepts within the C++ programming language include polymorphism, virtual and friend functions, templates, namespaces and pointers.

4.2 Code

Source code is the fundamental component of a computer program that is created by a programmer. It can be read and easily understood by a human being.

4.2.1 Insertion Code

The following pseudo code represents the insertion Code.

```
void persdetails::p_input(int cd) //input func() of class1
{
    trvlcode=cd;
    num=0;
    clrscr();
    cout<<"\n\n\t PERSONAL DETAILS \n";
    cout<<"\n\t\t* Please fill in the details:\n\n\n\t\t\t1.Family Name: ";
    gets(familyname);
    cout<<"\n\t\t\t2.Address: ";
    gets(add);
    cout<<"\n\t\t\t3.Contact Number(10 Digit Mobile Number) : ";
    gets(phnum);
    cout<<"\n\n\n\t\tEnter The No of People Travelling: ";
    cin>>numppl;
    clrscr();
    if(numppl>0)
    {
      cout<<"\n\t\tPlease Enter The Details of each Member/Members: "<<endl;
      cout<<"\t\t\n";
      for(int i=0;i<numppl;i++)
      {
         cout<<endl<<"\n\t\tMember "<<i+1;
         cout<<"\n\n\t\tFirst Name: ";
         gets(name[i]);
         cout<<"\n\t\tAge: ";
         cin>>age[i];
         cout<<"\n\t\tGender(M/F): ";
         cin>>gender[i];
         cout<<"\n\t\tPassport Number: ";
         gets(passnum[i]);
         if(age[i]>5)
         {
             num++; //to calculate no of travellers below 5 yrs
         }
      }
  }
}
```

4.2.2 Display Code

The following code represents the display Code.

```
void family(int c,int &flag) //to display familyname+to check for record
{
  flag=0;
  clrscr();
  ifstream ifl("PersonalDetails.txt",ios::binary);
  if(!ifl)
  cout<<"\nError";
  ifl.read((char*)&pob,sizeof(pob));
  while(!ifl.eof())
  {
     if(pob.givecode()==c)
     {
        flag=1;
        break;
     }
  ifl.read((char*)&pob,sizeof(pob));
  }
  if(flag==1)
  {
      cout<<"\n\n\t\t **";
      pob.givefam();
      cout<<"'s FAMILY DATABASE **";
  }
  else
  {
      cout<<"\nError in locating record!!"; }
      ifl.close();
  }
```

4.2.3 Deletion Code

The following code represents the deletion code. We sketch how deletion works in various cases.

```
void deletion(int c)  //common delete func(){
ofstream ofl2("temp1.txt",ios::binary);
if(!ofl2)
cout<<"Error While Opening File";
ifstream ifl4("PersonalDetails.txt",ios::binary);
if(!ifl4)
cout<<"Error While Opening File";
ifl4.read((char*)&pob,sizeof(pob));
while(!ifl4.eof()){ if(pob.givecode()!=c){
ofl2.write((char*)&pob,sizeof(pob));}
ifl4.read((char*)&pob,sizeof(pob));}
remove("PersonalDetails.txt");rename("temp1.txt","PersonalDetails.txt");
ofl2.close();ifl4.close();
ofstream ofl3("temp2.txt",ios::binary);
if(!ofl3)
cout<<"\nError While Opening File";
ifstream ifl5("TravelDetails.txt",ios::binary);
if(!ifl5)
cout<<"\nError While Opening File";
ifl5.read((char*)&tob,sizeof(tob));
while(!ifl5.eof()){
if(tob.gtcode()!=c){
ofl3.write((char*)&tob,sizeof(tob));}
ifl5.read((char*)&tob,sizeof(tob));}
ofl3.close();ifl5.close();
remove("TravelDetails.txt");
rename("temp1.txt","TravelDetails.txt");
cout<<"\n\n\t\tDeletion Completed!";getch();
}
```

4.2.4 Modification Code

The following code represents the modify code. It helps in editing the record.

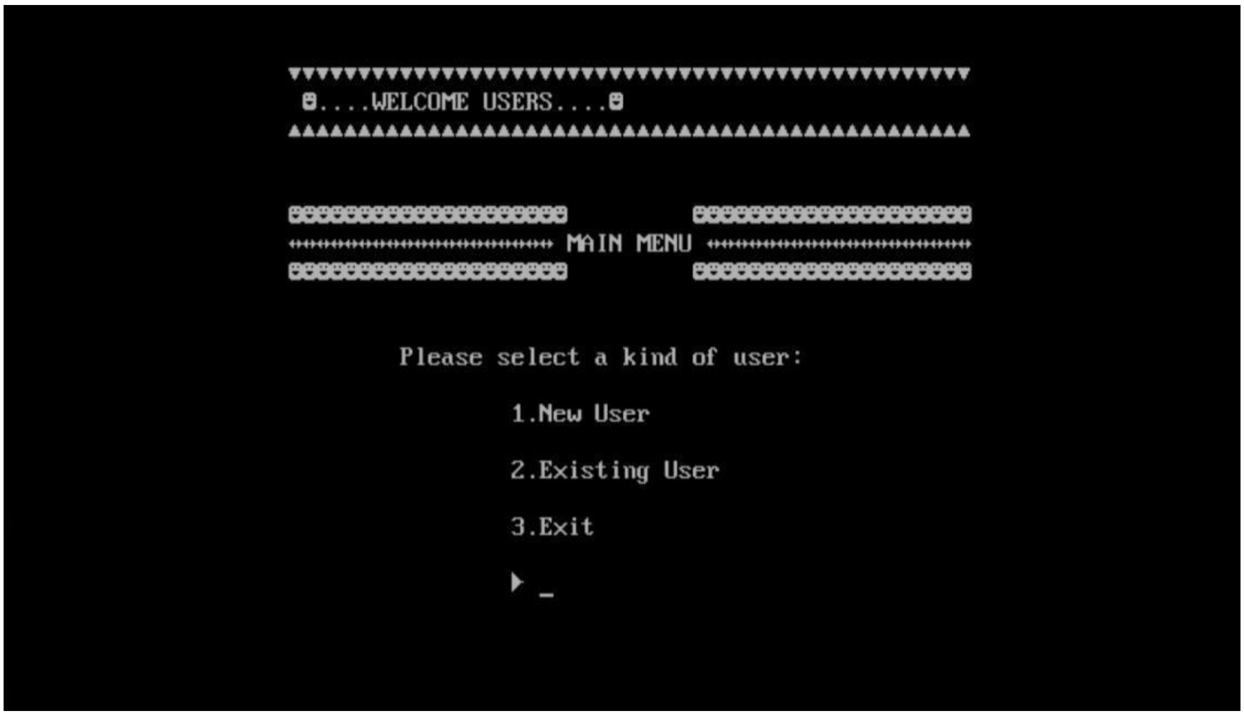
Modification code:

```
void editp(int c)  //to edit persdetails
{
 ofstream ofl2("temp1.txt",ios::binary);
 if(!ofl2)
 cout<<"Error While Opening File";
 ifstream ifl4("PersonalDetails.txt",ios::binary);
 if(!ifl4)
 cout<<"Error While Opening File";
 ifl4.read((char*)&pob,sizeof(pob));
 while(!ifl4.eof())
  {
if(pob.givecode()==c)
{
  clrscr();
  cout<<"Please Enter the New details of the record"<<endl;
  pob.p_input(c);
  ofl2.write((char*)&pob,sizeof(pob));
  cout<<"\n\t\t\tModification Succesful!!!";
  ifl4.read((char*)&pob,sizeof(pob));
}
else
{
  ofl2.write((char*)&pob,sizeof(pob));
  ifl4.read((char*)&pob,sizeof(pob));
}
 }
 remove("PersonalDetails.txt");
 rename("temp1.txt","PersonalDetails.txt");
 ifl4.close();
 ofl2.close();
 getch();
}
```

4.3 Discussion of Results

All the menu options provided in the application and its operations have been presented in as snapshots. A detailed view of all the snapshots of the output screen is given here.

4.3.1 Menu Operations



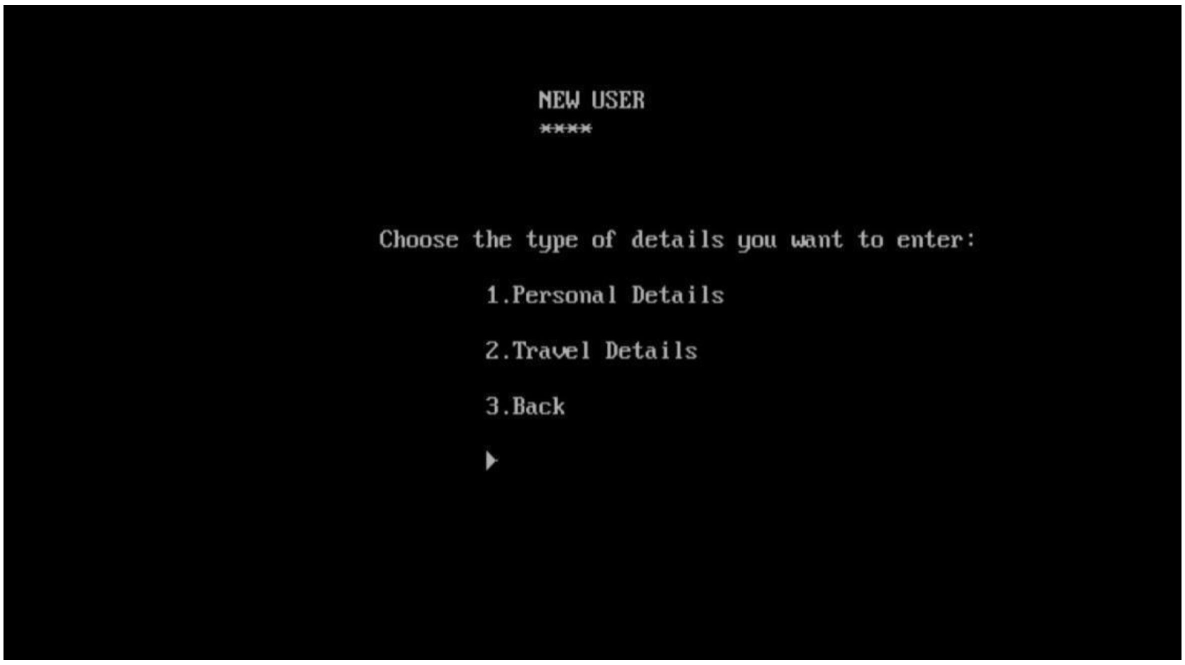
```
▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼
 ☻....WELCOME USERS....☻
▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲

☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻          ☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻
↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔ MAIN MENU ↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔↔
☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻          ☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻☻

        Please select a kind of user:

                1.New User

                2.Existing User

                3.Exit

                ► _
```

Figure 4.1 Menu Operations

The figure 4.1 shows that the user is prompted to select the kind of user i.e. new user, Existing user and Exit. After the procedure press any key to go back to the main menu.

4.3.2 New User Operations

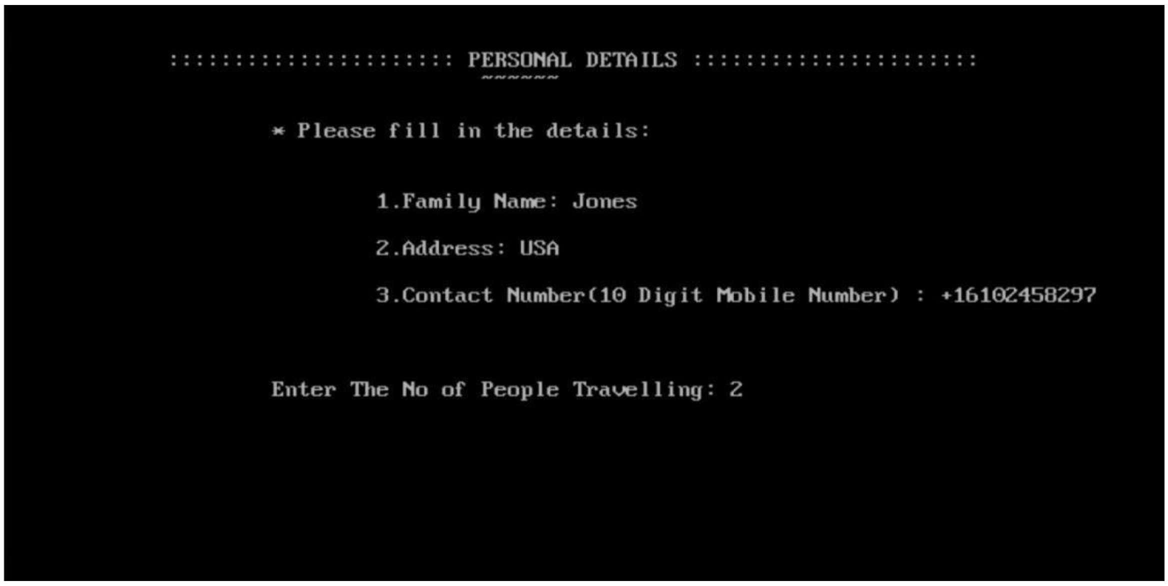
New user operation consists of personal details, travel details and back option. Insertion can be done on the following operations, we can choose the type of detail to enter for example in personal details the user is prompted to enter basic details such as name age gender etc.



```
NEW USER
****

Choose the type of details you want to enter:

        1.Personal Details

        2.Travel Details

        3.Back

        ▶
```

Figure 4.2 New Users

4.3.3 Personal details

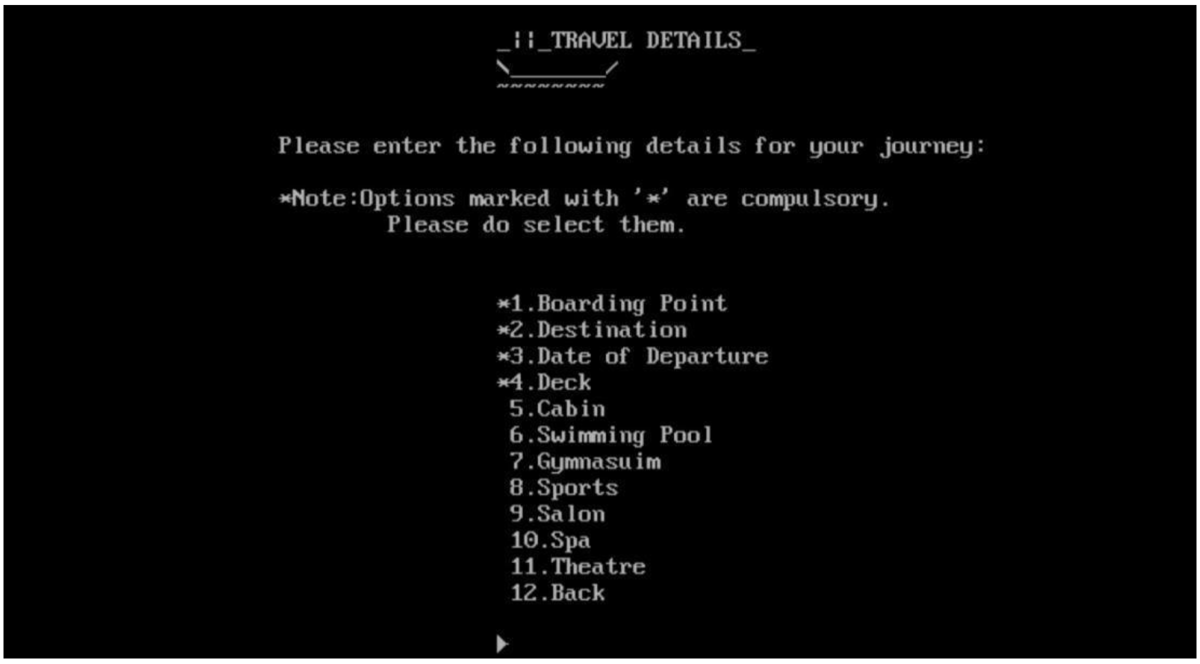


```
:::::::::::::::::::::: PERSONAL DETAILS ::::::::::::::::::::::
                       ~~~~~~

        * Please fill in the details:

                1.Family Name: Jones

                2.Address: USA

                3.Contact Number(10 Digit Mobile Number) : +16102458297


        Enter The No of People Travelling: 2
```

Figure 4.3 Personal details

Personal details consists of information such as Family Name, Address, Contact number (any 10 digit number) and to enter the details of number of people travelling. After the procedure the user should press any key to go back to the menu page.

4.3.4 Travel details

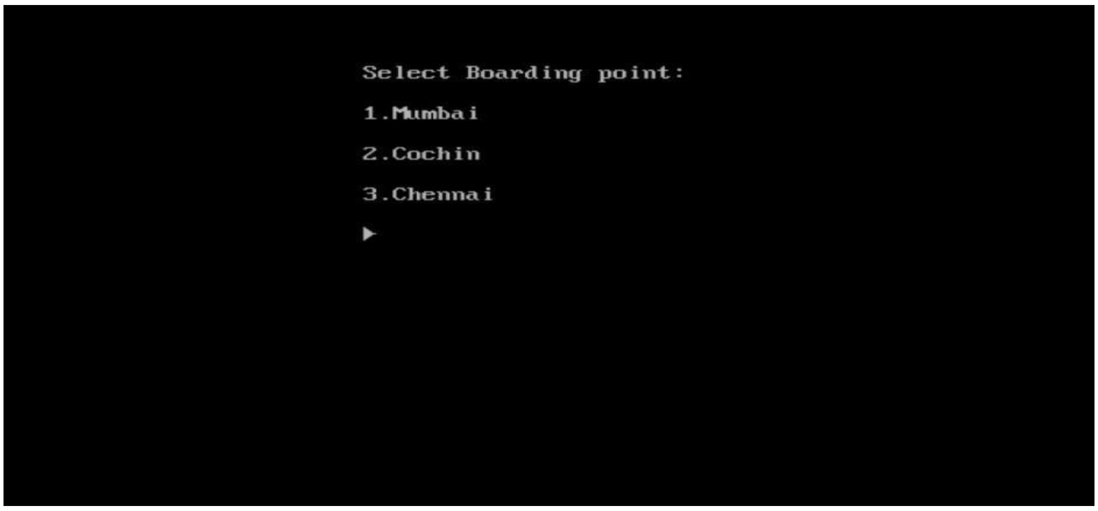


```
_¦¦_TRAVEL DETAILS_
\________/
~~~~~~~~

Please enter the following details for your journey:

*Note:Options marked with '*' are compulsory.
        Please do select them.


*1.Boarding Point
*2.Destination
*3.Date of Departure
*4.Deck
 5.Cabin
 6.Swimming Pool
 7.Gymnasuim
 8.Sports
 9.Salon
 10.Spa
 11.Theatre
 12.Back

►
```

Figure 4.4 Travel details

Fig 4.4 explains about the travel details that consist of boarding point, destination, date of departure, deck, cabin, swimming pool, gymnasium, sports, salon, spa, theatre. The user can choose any option and enter the details into it. After the procedure the user can press back key to go back to main menu.

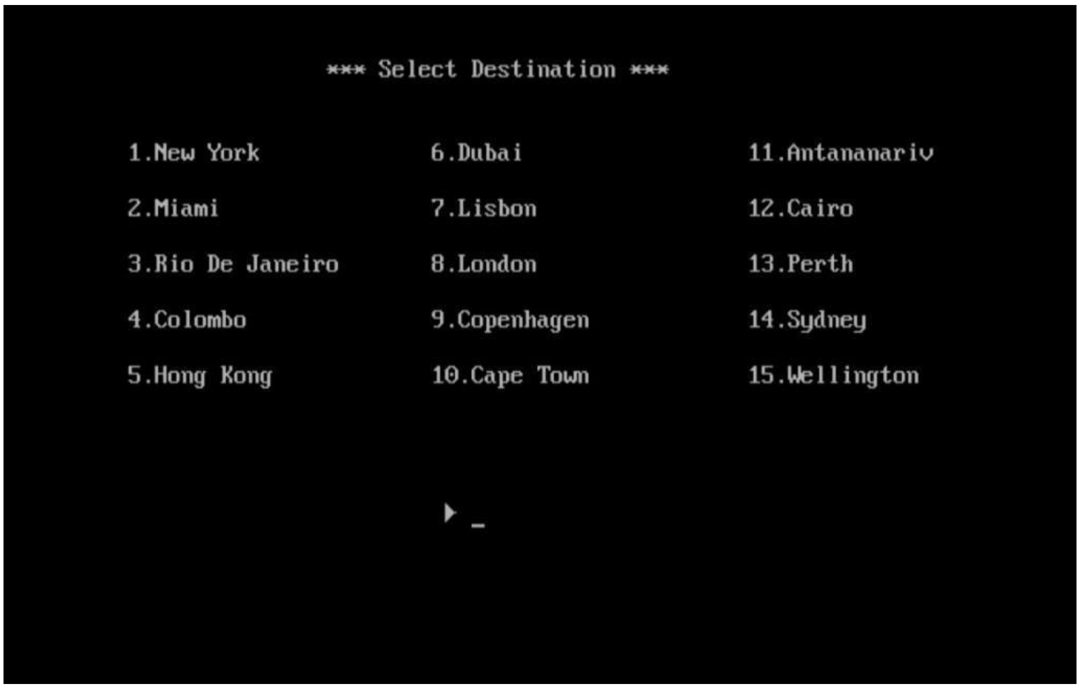
4.3.5 Source and Destination



```
Select Boarding point:

1.Mumbai

2.Cochin

3.Chennai

►
```

Figure 4.5.1 Source Operation

Fig 4.5.1 shows that the user is prompted to select the boarding point therefore the options are Mumbai, Cochin, Chennai. Press any key to go back to the menu page.

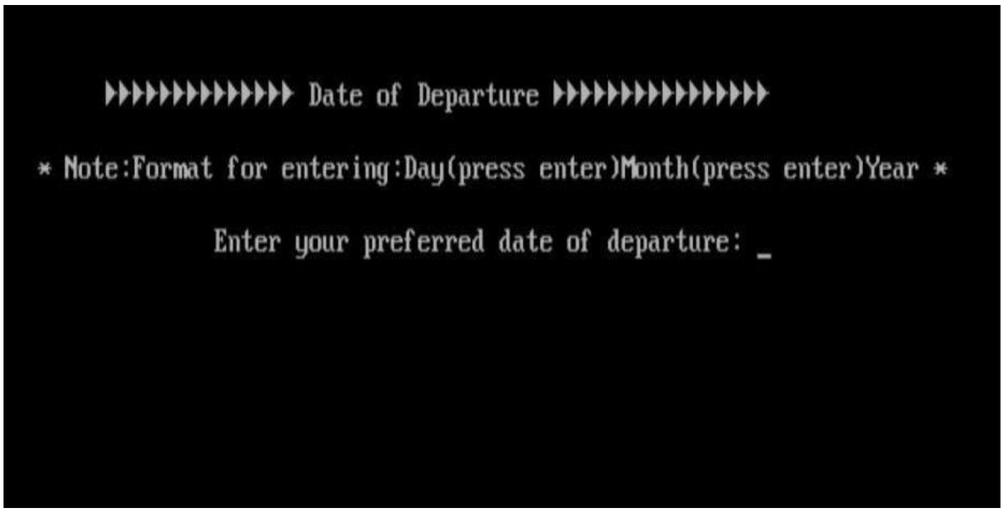


```
                    *** Select Destination ***

    1.New York          6.Dubai             11.Antananariv

    2.Miami             7.Lisbon            12.Cairo

    3.Rio De Janeiro    8.London            13.Perth

    4.Colombo           9.Copenhagen        14.Sydney

    5.Hong Kong         10.Cape Town        15.Wellington



                             ▶ _
```

Figure 4.5.2 Destination Operation

Fig 4.5.2 shows that the user is prompted to select the destination point therefore there are several options given as shown in the figure. Press any key to go back to the menu page.

4.3.6 Date of Departure

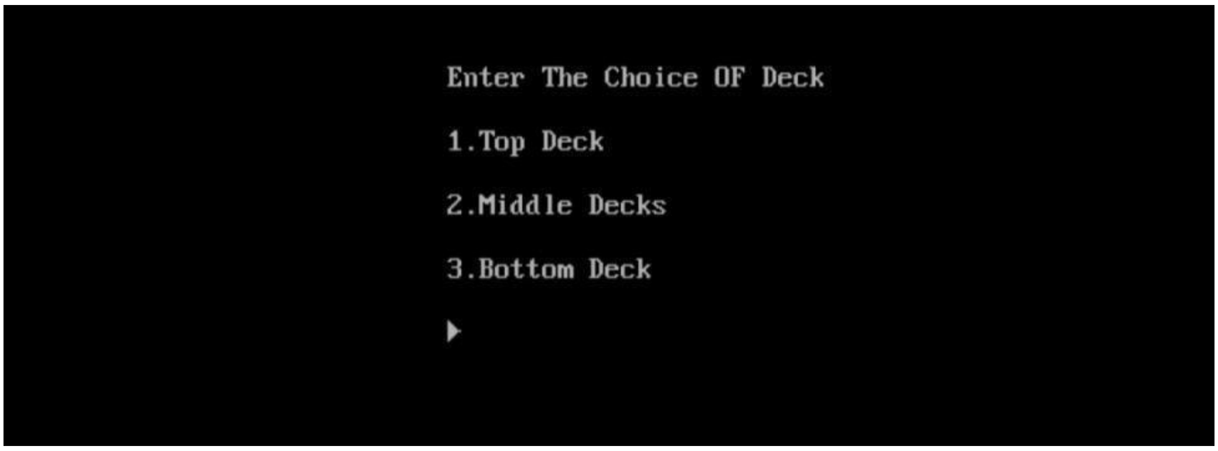


```
      ▶▶▶▶▶▶▶▶▶▶▶▶▶▶ Date of Departure ▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶▶

  * Note:Format for entering:Day(press enter)Month(press enter)Year *

              Enter your preferred date of departure: _
```

Figure 4.6 Date of Departure

Fig 4.6 shows that the user is prompted to select the day, month and year and enter the necessary details acording to the format. Also the user should enter the preferred date of departure. Press any key to go back to the menu page.

4.3.7 Choice of Deck

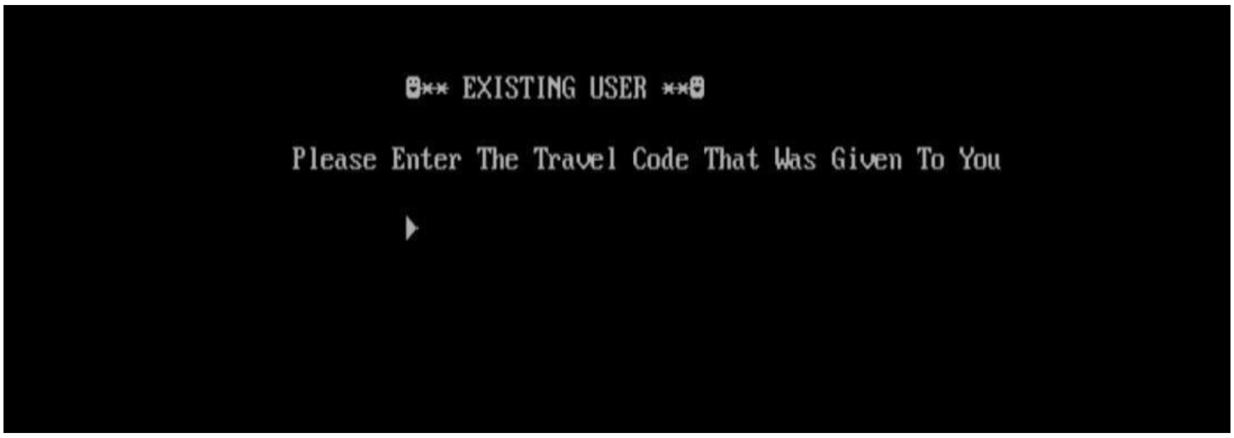


```
Enter The Choice OF Deck

1.Top Deck

2.Middle Decks

3.Bottom Deck

▶
```

Figure 4.7 Choice of deck

Figure 4.7 shows that the user is promoted to enter the choice of deck. It consists of Top Deck, Middle Decks and Bottom Deck. Press any key to go back to the menu page.

4.3.8 Existing User



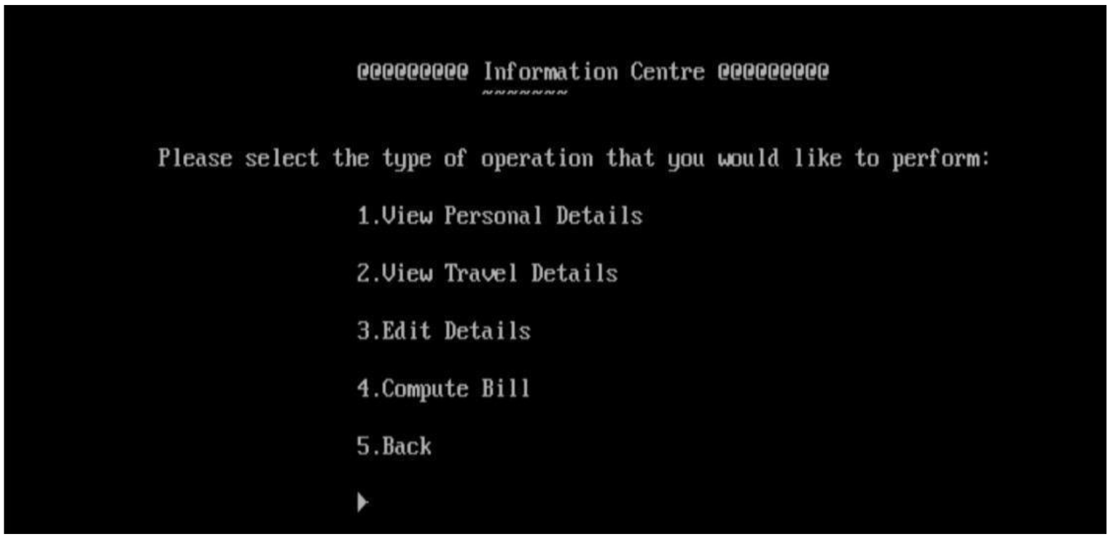
```
      ☻** EXISTING USER **☻

Please Enter The Travel Code That Was Given To You

      ▶
```

Figure 4.8 Existing User

Figure 4.8 shows that the user is prompted to enter the travel code that was previously given to the users.

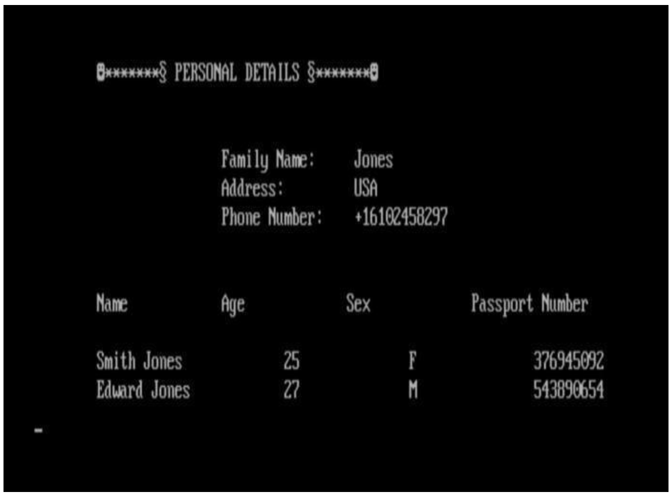
4.3.9 Information center and view details

Figure 4.9 shows that the user is prompted to select the type of operation that the user likes to perform for example to view personal and travel details and edit and compute bill.

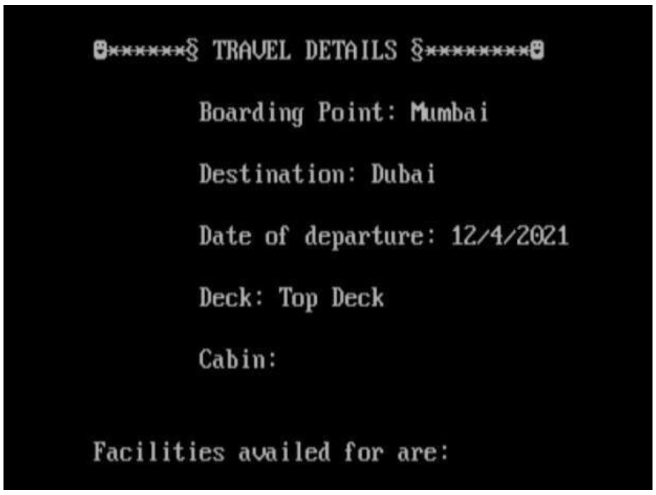


```
            @@@@@@@@@ Information Centre @@@@@@@@@
                      ~~~~~~~

   Please select the type of operation that you would like to perform:

                1.View Personal Details

                2.View Travel Details

                3.Edit Details

                4.Compute Bill

                5.Back

                ▸
```

Figure 4.9 Information center and view details



```
@*******§ PERSONAL DETAILS §*******@


              Family Name:     Jones
              Address:         USA
              Phone Number:    +16102458297


Name          Age          Sex          Passport Number

Smith Jones        25           F            376945092
Edward Jones       27           M            543890654
-
```



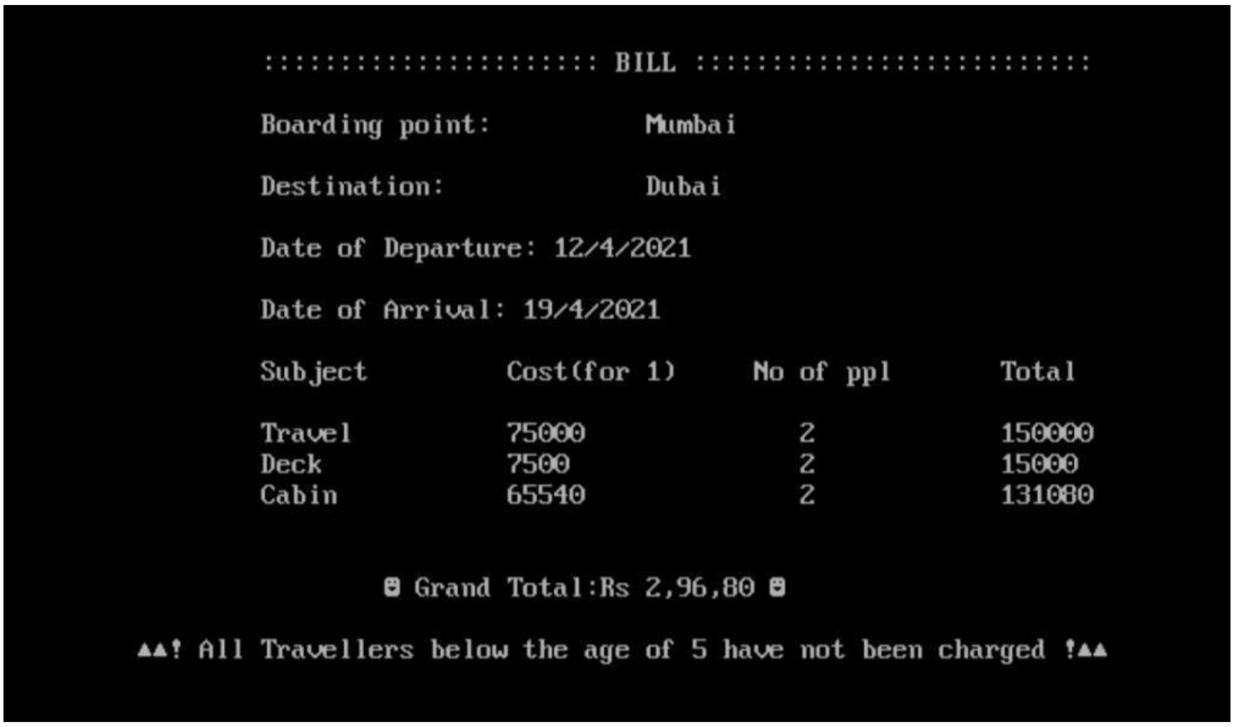
```
@*******§ TRAVEL DETAILS §*********@

        Boarding Point: Mumbai

        Destination: Dubai

        Date of departure: 12/4/2021

        Deck: Top Deck

        Cabin:


Facilities availed for are:
```

Figure 4.9.1 View Personal details Figure 4.9.2 View Travel details

Figure 4.9.1 shows that the user is prompted to view the personal details that include Family name, Address, Phone number also including gender and age of the user.
Figure 4.9.2 shows that the user is prompted to view the travel details that are, boarding point,destination, date of departure, deck, cabin etc.

4.3.10 Billing Operations

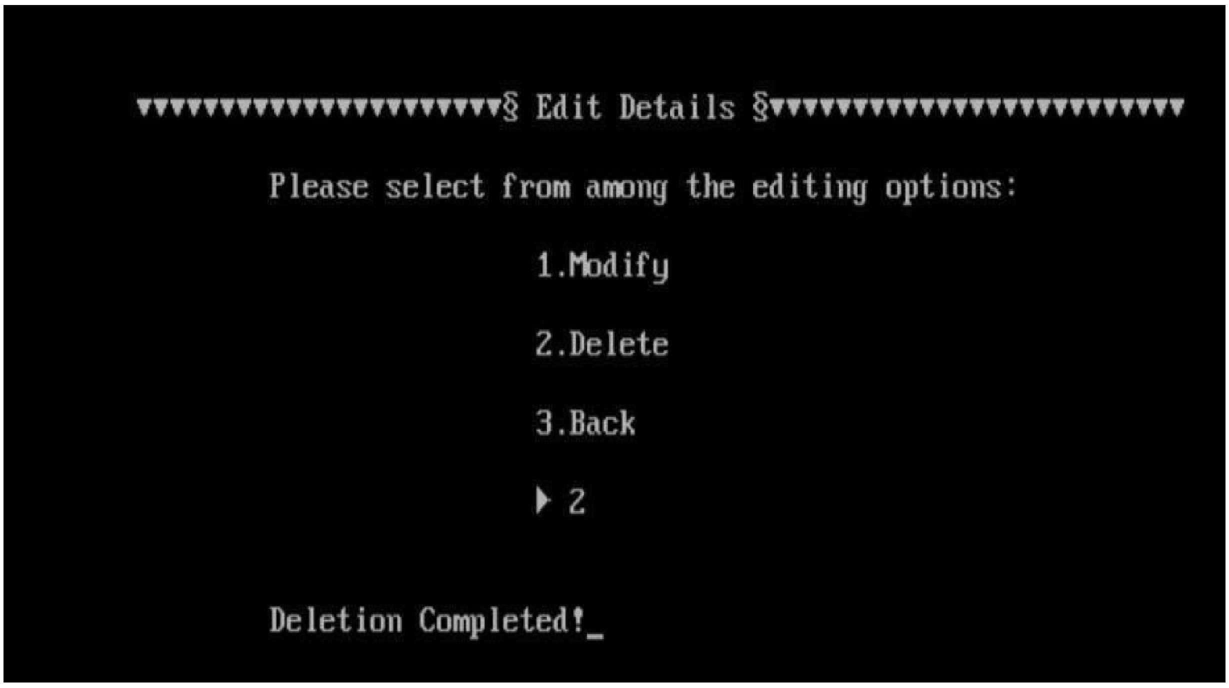
Figure 4.10 shows the detailed information of the source and destination, departure and arrival, number of people travelling and cost of the users. Children below age of 5 are not charged.



```
:::::::::::::::::::::: BILL :::::::::::::::::::::::::

Boarding point:          Mumbai

Destination:             Dubai

Date of Departure: 12/4/2021

Date of Arrival: 19/4/2021

Subject         Cost(for 1)     No of ppl       Total

Travel          75000             2             150000
Deck            7500              2             15000
Cabin           65540             2             131080


        ☻ Grand Total:Rs 2,96,80 ☻

▲▲! All Travellers below the age of 5 have not been charged !▲▲
```

Figure 4.10 Billing Operations

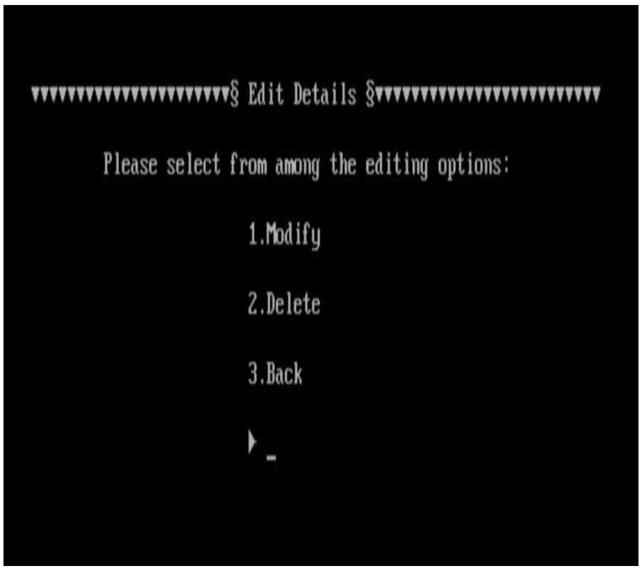
4.3.11 Modification Operation



```
▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼§ Edit Details §▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼

    Please select from among the editing options:

                 1.Modify

                 2.Delete

                 3.Back

                 ► 2


    Deletion Completed!_
```

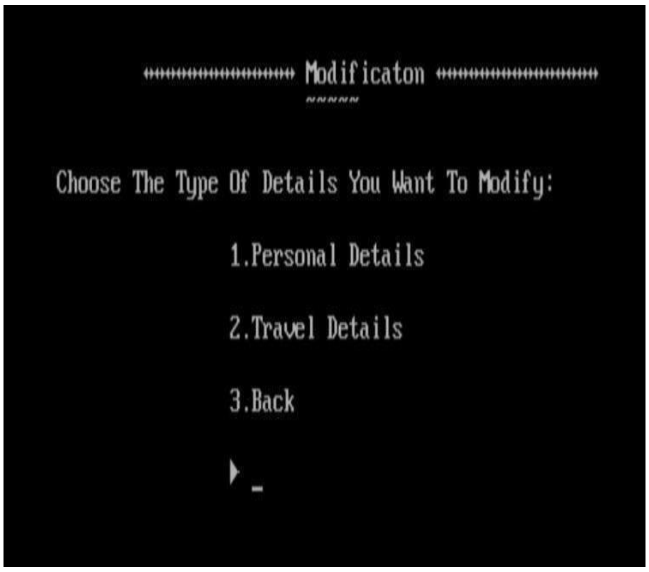
Figure 4.11 Deletion operation

Figure 4.11 shows that the user is prompted to select from modify and delete the data, the User has to enter 2 to delete the data, therefore the required data shall be deleted.



```
▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼§ Edit Details §▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼▼

        Please select from among the editing options:

                        1.Modify

                        2.Delete

                        3.Back

                        ▸ _
```

Figure 4.11.1 Edit details



```
         ♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦ Modificaton ♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦♦
                          ~~~~~

  Choose The Type Of Details You Want To Modify:

                1.Personal Details

                2.Travel Details

                3.Back

                ▸ _
```

Figure 4.11.2 Modify details

Figure 4.11.1 shows that the user is prompted to select the options; if the user enters 1 then the second screen is displayed for the user to choose the details to modify i.e. personal details and travel details. Therefore, the user can modify accordingly. After the procedure, the user can enter option 3 to exit the page and it redirects to the main menu.

Chapter 5

CONCLUSIONS AND FUTURE ENHANCEMENTS

- The project explained here is very useful in maintaining the customer's details and can work on the existing records of customer details. The workload of the travel agency manager is also considerably reduced. In the near future, the online registration can be implemented.
- Travel drives economic prosperity and sustained development in many sectors. Also at the individual organization level.
- As well as a very important economic activity- earning foreign currency for destinations and contributing to economic development in developed and developing nations.

Future Enhancements are as follows:

- The project can further be enhanced by using Hashing technique to further reduce the number of accesses and improve the performance.
- Hashing technique allows record access in only one or very few runs with the help of a hashing function.
- Using such advanced techniques for better indexing, we can implement complex functions such as comprehensive balance sheets and budgeting.

REFERENCES

[1] Michael J Folk, Bill Zoellick, Greg Riccardi: File Structures – An Object Oriented Approach with C++, 3rd Edition, Pearson Education, 1998

[2] K.R. Venugopal, K.G. Srinivas, P.M. Krishnaraj: File Structures Using C++, Tata McGraw-Hill, 2008.

[3] Scot Robert Ladd: C++ Components and Algorithms, BPB Publications, 1993

[4] Raghu Ramakrishan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw Hill, 2003

[5] www.geeksforgeeks.com

[6] www.codebind.com

[7] www.includehelp.com