

Chapter 3

SYSTEM DESIGN

3.1 Design of the Fields and Records

The V_ID is declared as a character array that can hold a maximum of 10 characters. All the fields name, address, contact number, sex are declared as characters, each having a predetermined range, only within which it is accepted during input. Hence, a typical data file record can have up to 77 bytes of information to be stored, hence occupying a maximum of 90 bytes, in the data file. This includes the 4 bytes taken up by the field delimiters (|) and the 1 byte taken up by the record delimiter (\n). The class declaration of a typical product file record and member functions is as shown in Figure 3.1:

```
class voter
{
    public:
        char V_ID[10];
        char name[30];
        char addr[40];
        char dob[8];
        char gender[6];
        void Clear();
        int Unpack(fstream&);
        int Pack(fstream&);
        void Input(int);
        void Display();
        void Append();
        void Remove(int);
        ~voter(){}
        void Assign(voter&);
}
```

Figure 3.1 Design Of Fields

3.2 User Interface

The User Interface or UI refers to the interface between the system and the user. Here, the UI is menu-driven, that is, a list of options (menu) is displayed to the user and the user is prompted to enter an integer corresponding to the choice, that represents the desired operation to be performed.

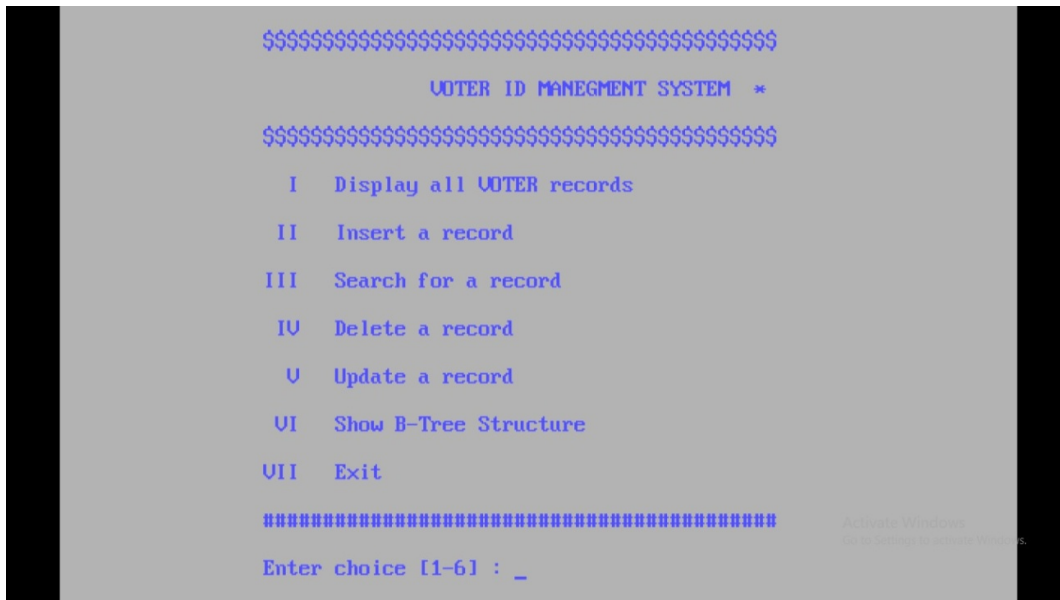


Figure 3.2 User Menu Screen

3.2.1 Insertion of a Record

If the operation desired is Addition of voter details, the user is required to enter 2 as his/her choice, from the menu displayed, after which a new screen is displayed. Next, the user is prompted to enter the Voter ID, name of the voter, address of the voter, date of birth of the voter, sex. The user is prompted for each input until the value entered meets the required criterion for each value. After all the values are accepted, a “record inserted” message is displayed and the user is prompted to press any key to return back to the menu screen. After insertion of details of the voter, every voter is given with the unique Voter ID which cannot be given to any other voters.

3.2.2 Display of a Record

If the operation desired is Display of Records, the user is required to enter 1 as his/her choice from the menu displayed after which new screen is displayed. If there are no records in the file, “no records found” message is displayed. For voter files with atleast one record, each Voter ID, followed by the details of each record within the file, with suitable headings, is displayed. In each case, the user is then prompted to press any key to return back to the menu screen.

3.2.3 Deletion of a Record

If the operation desired is Deletion, the user is required to enter 4 as his/her choice, from the menu displayed, after which a new screen is displayed. Next, the user is prompted to enter the voter name, whose file from which a record is to be deleted. If there are no records in the file, a “no records to delete” message is displayed, and the user is prompted to press any key to return back to the menu screen. If there is at least 1 record in the file, the user is prompted for the voter, whose matching record is to be deleted. The voter name entered is used as a key to search for a matching record. If none is found, a “record not found” message is displayed. If one is found, details of the particular record is displayed followed by confirmation of deletion. Then “record deleted” message is displayed. In each case, the user is then prompted to press any key to return back to the menu screen.

3.2.4 Search for a Record

If the operation desired is Search, the user is required to enter 3 as his/her choice, from the menu displayed, after which a new screen is displayed. Next, the user is prompted to enter the V_ID, whose file the record is to be searched for. The V_ID entered is used as a key to search for a matching record. If none is found, a “record not found” message is displayed. If one is found, the details of the record, with suitable headings, are displayed. In each case, the user is then prompted to press any key to return back to the menu screen.

3.2.5 Modifying or updating a Record

If the operation desired is Update, the user is required to enter 5 as his/her choice, from the menu displayed, after which a new screen is displayed. Next, the user is prompted to enter the patient id, whose file from which a record is to be modified. If there are no records in the file, a “no records to delete” message is displayed, and the user is prompted to press any key to return back to the menu screen. If there is at least 1 record in the file, the user is prompted for the V_ID, whose matching record is to be updated. Instead of deleting the old record and inserting new one, the system makes changes to the old product itself for the matching V_ID record.

The V_ID entered is used as a key to search for a matching record. If none is found, a “record not found” message is displayed and the user is then prompted to press any key to return back to the menu screen. If one is found, a “confirm permanent modification” message is displayed, after pressing Y, the user is prompted to enter the name of the voter, address of the voter, date of birth of the voter, sex to be inserted into the file. The user is prompted for each input until the value entered meets the required criterion for each value. After all the values are accepted, a “record updated” message is displayed and the user is prompted to press any key to return back to the menu screen.

3.2.6 Design of Index

The B-TREE is declared as a class, an object of which represents a node in the B - TREE. Each node contains a count of the number of entries in the node and a reference to each descendant. The maximum number of descendants is 4 while the minimum is 2. Each node has an array of objects, each an instance of the class type index as shown in Fig. 3.3. Each object of type “index” contains a page id field and an address field. The contents of the B -TREE are written to the index file on disk after each insert, delete and modify operation. To ensure efficient space utilization, and, to handle the conditions of underflow and overflow, nodes may be merged with their siblings, or split into 2 descendants, thereby creating a new root node for the new nodes created, or, entries within nodes maybe shifted to save space.

The above operations are used to maintain a balanced tree-structure after each insertion and deletion. The links of a node, beginning at the root, are traversed recursively, while displaying, and, writing to the index file, in order to maintain an ascending order among index entries. The same links are traversed when there is a request to search for a record, if present.