NAME:P.LAKSHMI SUMANA

USN:1NH18CS132

PROJECT TITLE: TELECOM BILLING

SEC:3B

CHAPTER 1

INTRODUCTION

1.1) PROBLEM DEFINITION

The project is mainly aimed to present the requirement of the Telephone Billing. In this project the telephone bills are automatically calculated. In this project almost every work is done that is related to telephone billing system. Telecom billing provides customer with many services like adding records, listing records, deleting records, record modification, viewing customer records & all works related to bills. Customer will also be provided with payment option. Some other options like Meter readings to bill calculation and bill generation are provided. "Telephone Billing" is been developed due to the increase in requirement of the work and to develop new work culture. Thus, to overcome such issues and problems a new software has been developed to minimize manual work and improve work efficiency by saving time and to provide flexibility and user-friendly environment to the customer. As the system that was proposed previously was very difficult and was manual with lot of errors this project was developed.

1.2) OBJECTIVES:

The objective of this project named "Telephone Billing" is to minimize the manual work, save time, reduce effort and to increase the speed and efficiency of the work.

This system is developed with the below objectives:

1) The retrieval of information is easier.

2) Management of records in database and overall process is easier.

3) Customers account is highly secured by taking few measures like login system.

4) Customers are logged in by using username and password

5) Using this system data can be greatly reduced because this system is developed using Visual Basic of 6.0 as. This system looks for duplication of effort and inefficient procedures

1.3) METHODOLOGY

MODULES:

This system of “TELECOM BILLING” is carefully analysed and has been divided into two major divisions and is been presented with the following modules. This module involves the following:

- Administrator
- User

1. ADMINISTRATOR

The administrator is the super user of the application. Only admin have access into the admin page. Admin may be the owner of the shop. The administrator has all the information about all the users and about the products.

- Payment.
- Add records.
- Delete records.
- Display records.
- List records.
- Search records.

2. USER

A user does not have any super user privilege. User cannot have access into admin page. The admin page will be encrypted. User can access through admin page only if he/she knows the password and user id.

REGISTERATON:

A new user will have to register in the system by providing essential details in order to access the page. The admin must grant permission for the user. The user will be checked by the admin before logging in.

->A user must log in by using his/her user id and password to the system after registration.

->The user can then pay their pending bills after successful login.

->The user can also search for the billing history.

EXISTING SYSTEM

Previously this system was a manual one. Regardless of the topic the process involved by the system was done through registers or files. This system was full of complexities and difficulties. Separate files were maintained when a customer makes new connections. It was a tedious job to update records of customer. Few administrative works like maintaining rates of calls, addition of calls or modification of metered calls and customer entries was a very difficult task.

DRAWBACKS OF EXISTING SYSTEM

In previous system all the office works was manual. Thus, the manual work process was time consuming and slower. Few drawbacks of the existing system are listed below:

1. As the system is totally based on manual work there were chances of getting errors in process.

2. Most common and major drawback in the previous system was the speed of retrieval of data that led to delay of process.

3. Management of large data was very difficult and tremendous job.

4. one of the main drawbacks is calculations are more error prone.

5. There was a high chance of duplication of data and information.

6. Updating of data and records was very difficult job.

7. It was difficult to get different statistical data from central database.

From above facts, information and drawbacks we can clearly prove that this system needs to be computerized and hence have been decided to computerize the system. In this way “TELEPHONE BILLING” was proposed. Since the previous system was completely manual with lots of complexities, shortcomings, difficulties and errors and the data was being stored in registers, files etc. so to overcome the drawbacks of this system, a new computerized and modified system was required. A new system was required so that the information can be provided to the customer more quickly, easily and more accurately without any errors. It is not easy to do several works like maintaining rates of calls, addition or modification of metered calls & customer entries.

PROPOSED SYSTEM

The new system proposed was titled as “TELEPHONE BILLING” and was proposed to remove all the drawbacks and limitations discussed above. The vital ingredient for any operation or management of any organization is information. Hence all the systems should have the capability to provide error free information after processing the data. The system has been developed with an objective for developing a more sophisticated and clear system that can easily be handled by all kind of customers. The proposed system aims at the efficient and timely information for decision-making, integrate with other functions, and reduce work.

Important features of this proposed system are listed below:

1. Efficient user interface with high economic features are built into it.

2. Design of system and particular way to make the integration with other systems easier.

3. Customer has complete control on his/her account as it provides valid data.

4. Simple and User-friendly error messages are sent to customer wherever required.

5. Addition, deletion, listing, modification of records as is easier.

6. connecting to new users is easier.

7. Generation of bills for customers through this system can be easily handled by all kind of users. This system aims for the efficient and timely information for decision-making and reduce work.

ADVANTAGES OF THE PROPOSED SYSTEM

1. Reduces the workload for staff.

2. Reduce the delay in processing time.

3. Reduces the delay in bill generation.

4. Provides user-friendly nature for the customer in all possible ways.

5. Provides greater flexibility for the user.

6. easier to make records in database.

7. Stores data in centralized location to reduce work and increase consistency and efficiency.

1.4) EXPECTED OUT COMES

- it displays the welcome page
- next it displays the options

  1. Add records
  2. Delete records
  3. List records
  4. Modify records
  5. Payment
  6. Exit
- if we select option 1, we can add records to the database

  After choosing this option it asks for other details like

  1. Name of customer
  2. Phone number of customers
  3. Amount to be paid
- if we choose option 2, we can delete records from the database.
- if we choose option 3, we can list the records in the database.
- if we choose option 4, we can make payments.
- option 6 is used to exit from the output screen.

1.5) HARDWARE REQUIREMENTS

- Processor: any processor
- Input device: standard keyboard and mouse
- Output device: high resolution monitor

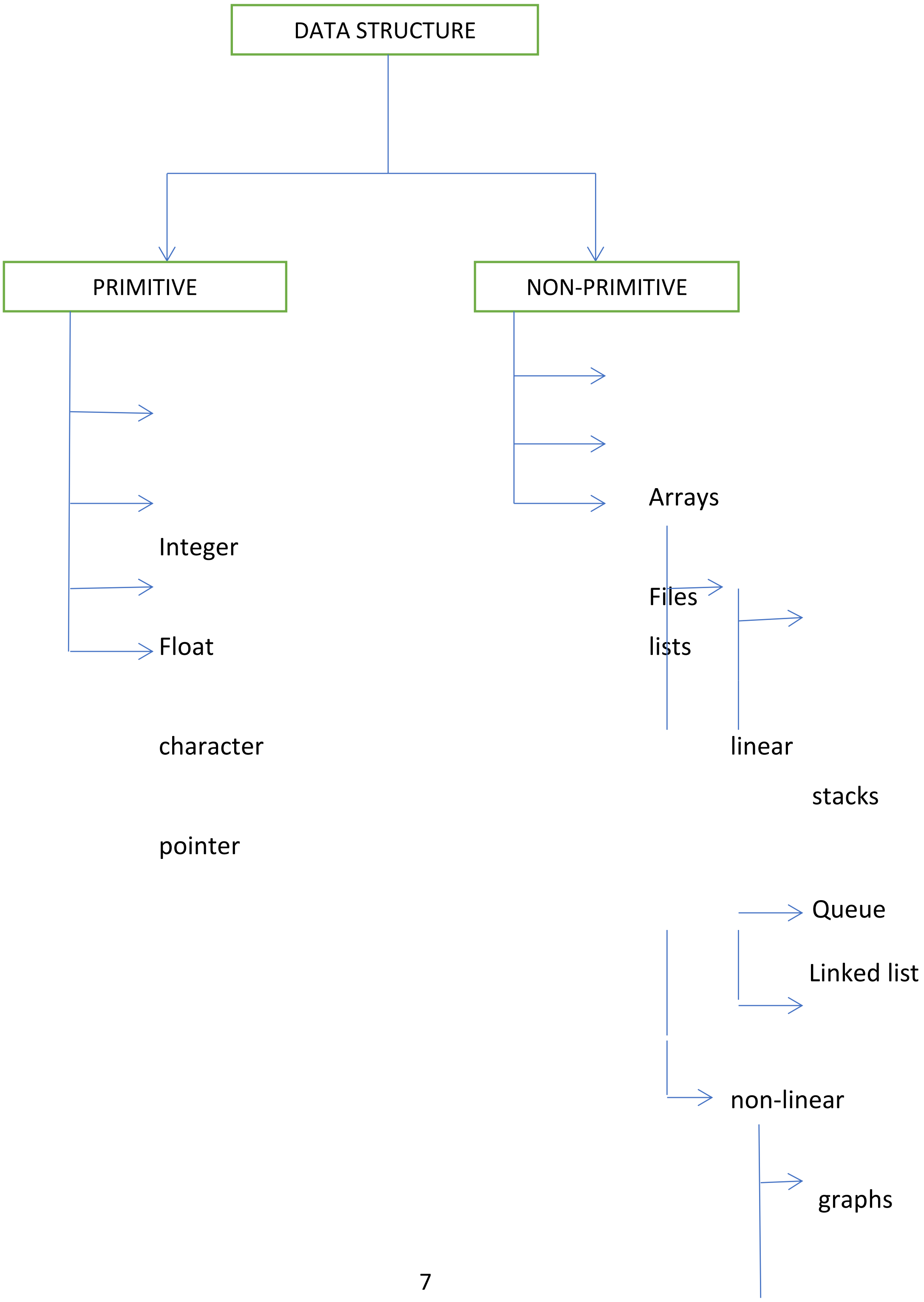
1.6) SOFTWARE REQUIREMENTS

- Operating system windows
- Turbo C++ compiler

CHAPTER 2

DATA STRUCTURE

Data structure is used to store data in a specified format.



→ Trees

FIGURE 2.1

DATA STRUCTURE

Data structure is a method of collecting, organizing, retrieving, manipulating the data.

CLASSIFICATION OF DATA STRUCTURE:

There are two types of data structure.

a) primitive data structure

b) non primitive data structure

a) PRIMITIVE DATA STRUCTURE:

It can be directly manipulated by machine instructions.

It is further classified into 3 types:

- int
- float
- char

b) NON-PRIMITIVE DATA STRUCTURE:

It cannot be directly manipulated by machine instructions.

It is also classified into 3 types:

- arrays
- lists
- files

2.1) ARRAYS:

Arrays is a collection of elements of the same data type. The arrays are stored sequentially in memory. In Java, array elements are treated as objects whereas in c they are not treated as objects.

TYPES OF ARRAYS:

- Single dimensional array
- Multi-dimensional array

SINGLE DIMENSIONAL ARRAY:

SYNTAX:

```
data_type array_name [array size];
```

Where

Data_type = type of the data (int, char, float)

Array_name = it is name of the array

Array_size = it is size of the array

EXAMPLE:

```
int a [30];
```

Where

Int is the data type

a is the variable name

30 is the size of the array

MULTI DIMENSIONAL ARRAY

SYNTAX:

```
data_type array_name [row_size] [column_size];
```

Where

Data_type = type of the data (int, char, float)

Array_name = it is name of the array

Row_size = it is size of the row

Column_size = it is the size of the column

EXAMPLE:

```
int a [30[30]];
```

Where

Int is the data type

a is the variable name

30 is the size of the row

30 is the size of the column

2.2) LIST

In non-primitive data structure, lists are further classified into 2 types. They are:

I) linear data structure

II) nonlinear data structure

LINEAR DATA STRUCTURE:

The elements are organized in linear manner; hence we call it as linear data structure. It is further classified into 3 types:

- Stack
- Queue
- Linked list

2.2.1) STACK:

It is a linear a data structure in which insertion and deletion of elements happens only at one end (top end). It follows last in first out order (LIFO). In stack, new item is added at the top of a stack. Both insertion and deletion operation are performed from one end of the stack

Different operations of stacks are:

I) push - to insert the element into the stack.

II) pop - to delete the elements from the stack.

III) display - to display the elements present in the stack.

Example:

Just imagine that you have a cookies jar. In this jar, you can add one cookie. The process when you add one cookie inside the jar from topmost position is called as push operation. Similarly, when you remove one cookie from the jar then that process of cookie removal from the topmost position is called pop operation.

The last entered cookie into the cookie jar is the first to be removed from the jar, thus a stack follows a last in first out principle (LIFO).

2.2.2) QUEUE:

It is a linear data structure in which insertion happens at one end and deletion happened at the other end. It follows Frist in Frist out order (FIFO).

Different operations of queues are:

I) insertion - to insert the element into the queue through rear end.

II) deletion - to delete the element from the queue through front end.

III) display - to display the elements present in the queue.

TYPES OF QUEUE:

- Linear queue or ordinary queue
- Circular queue
- Double ended queue
- Priority queue

1) LINEAR QUEUE

Linear queue is a queue in which insertion is possible at the rear side of the queue (i.e., end of the queue) and deletion is possible at the front side of the queue (i.e., beginning of the queue).

In linear queue, each the nodes are connected to each other nodes in a sequential manner. The first node's pointer is pointing to the value of the second node and second node's pointer is pointing to the third node and so on. But the first node of the queue has no pointer pointing towards it whereas the last node of the queue has no pointer pointing out from it.

2) CIRCULAR QUEUE

In a circular queue, each node is connected to the next node in the sequential manner but the last node's pointer is also connected to the first nodes address. Hence, the last node and the first node also gets connected which results a circular link. Therefore, it is called as circular queue.

3) DOUBLE ENDED QUEUE

The double ended queue is also known as dequeue. It is a queue in which insertion and deletion operations are possible at both the ends (i.e., front and rear side) of the queue.

Types of Double ended Queue:

- Input restricted double ended queue
- Output restricted double ended queue

INPUT RESTRICTED DOUBLE ENDED QUEUE:

It is the queue in which insertion can happen only at one end whereas deletion can happen at both the ends.

OUTPUT RESTRICTED DOUBLE ENDED QUEUE:

It is the queue in which deletion can happen only at one end whereas insertion can happen at both the ends.

4) PRIORITY QUEUE

Priority queue is the queue in which insertion operation and deletion operation happens based on the priority. The deletion Operation is performed in accordance to priority number (i.e. data item which has highest priority is removed first from the queue) and insertion is performed only in the order.

2.2.3) LINKED LIST

Linked List is a linear data structure. Linked List is also like an array but, the Linked List is not stored sequentially in the memory. Each linked list has 2 parts, the data part and the address part. Data part holds the element or data whereas Address part holds the address of the next element in the list, which is called a node.

Size of the linked list is not fixed and data items can be added at any locations in the linked list. The disadvantage of the linked list is that if we require a particular node, then we must traverse from the first node to the particular node that we require.

BASIC OPERATIONS OF LINKED LIST

- Insertion - to insert the data or information to the node of the linked list.
- Deletion - to delete the data or information of the node from the linked list.
- Display - to display the data or information present in the node of the linked list.

TYPES OF LINKED LIST

- Single linked list
- Double linked list
- Circular linked list
- Header linked list

1) SINGLE LINKED LIST

Single linked list is the linked list which has only one link. In other words, it is a collection of nodes where each node has two parts i.e., data which contains the elements and link to the next node

It can traverse only in one direction. Single linked list consumes less memory. But the disadvantage of the single linked list is that once the pointer is moved forward, we can't go back.

STRUCTURE REPRESENTION FOR SINGLE LINKED LIST:

```
struct slist
{
    int data;
    struct slist *ptr;
};
```

2) DOUBLE LINKED LIST

Double linked list is linked list which has to links. In other words, it is a collection of nodes where each node has three parts i.e., link to the previous node, data which contains the elements and link to the next node.

Double linked list consumes more memory. It can be traverse in both the direction.

STRUCTURE REPRESENTION FOR DOUBLE LINKED LIST:

```
struct slist
{
   struct dlist *prev;
   int data;
   struct dlist *next;
};
```

3) CIRCULAR LINKED LIST

Circular linked list is a collection of nodes where the last noted points back to the first node which means last node contains the address of first node.

CIRCULAR LINKED LIST IS CLASSIFIED INTO TWO TYPES:

- Circular single linked list
- Circular double linked list

4) HEADER LINKED LIST

Header linked list is a linked list which contains collections of nodes along with one extra node which is called as header node. Header node data is used to store some information such as number of nodes in linked list, address of last nodes, etc.

MAIN ADVANTAGES OF LINKEDLIST OVER ARRAYS ARE:

- Size of array is fixed; upper limit is known in advance. But in the linked list size is not fixed.
- Insertion of element is easier compared to array.
- Deletion of elements is easier compared to array.
- Memory wastage is less in linked list compared to array.

Structure of single linked list:

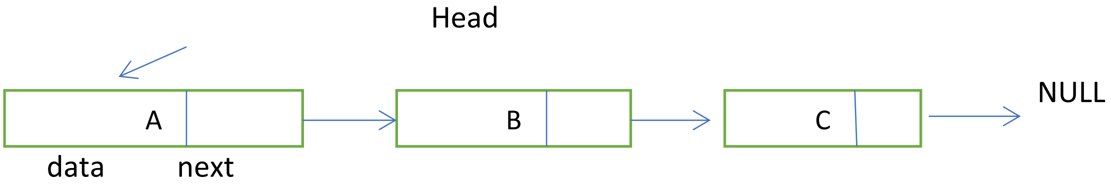


FIGURE 2.2

NON-LINEAR DATA STRUCTURE:

The elements are organized in nonlinear manner; hence we call it as nonlinear data structure. It is further classified into 2 types:

- Tree
- Graph

2.3) TREE

It is a nonlinear data structure which consists of collection of nodes organized in the hierarchical manner.

TREE TERMINOLOGY

ROOT NODE: This is the node which is at zeroth level.

INTERNAL NODE: All the nodes which has child nodes except the root node.

EXTERNAL NODE/LEAD NODE: Nodes which doesn't have child nodes

TYPES OF TREE

- Binary tree

- Strictly binary tree
- Full binary tree
- Complete binary tree

1) BINARY TREE

Binary tree is a tree which ha utmost two child nodes.

PROPERTIES OF BINARY TREE:

If the height of the binary tree is 'h', then maximum number of leaf nodes is '2^h' (i.e., 2 to the power h).

If 'h' is the height of the tree, then maximum number of nodes in the tree is '2^(h+1) - 1' (i.e., 2 to the power h+1 minus of h).

REPRESENTION OF BINARY TREE

Binary tree can be represented in two ways:

- Array representation
- Linked list representation

2) STRICTLY BINARY TREE

A tree which has zero or two child nodes is Known as strictly binary tree.

3) FULL BINARY TREE

It is binary tree in which every node should have to child nodes except leaf node.

4) COMPLETE BINARY TREE

Complete binary tree is a full binary tree till 'h-1' level but in the last level all the nodes are filled from left to right.

2.4) GRAPH:

It is a nonlinear data structure which consists of vertices and edges.

GRAPH TERMINOLOGY

- VERTEX: It is a data element which is nothing but a node. (E.g.: A, B, C)

- EDGE: It is the connection between two nodes. (E.g.: AB, BC)
- DIRECTED GRAPH: A graph which contains only directed edges.
- UNDIRECTED GRAPH: A graph which contains only undirected edges.
- MIXED GRAPH: A graph which contains both directed and undirected edges.

REPRESENTATION OF GRAPHS

They are represented in two ways:

- Array representation
- Linked list representation

2.5) MEMORY ALLOCATION

There are two types of memory allocation techniques. They are:

1) static memory allocation

2) dynamic memory allocation

STATIC MEMORY ALLOCATION:

Allocating memory to variables during compilation time.

DRAWBACK OF STATIC MEMORY ALLOCATION:

Once the memory is allocated, it can't be increased while executing program used in array.

If the entire allocated memory is not used, then memory will be wasted.

DYNAMIC MEMORY ALLOCATION:

Allocating memory during execution time or run time.in dynamic memory allocation memory can be increased while executing program used in linked list.

DYNAMIC MEMORY TECHNIQUES ARE:

- Malloc ()
- Calloc ()
- Realloc ()

- Free ()

MALLOC ()

Malloc () is also known as memory allocation. It is used to allocate single block of memory.

SYNTAX:

```
    ptr=(caste-type*) malloc(size);
```

EXAMPLE:

```
    ptr=(int*) malloc(sizeof(int));
```

CALLOC ()

Calloc () is also known as contiguous allocation. It is used to allocate multiple blocks of memory.

SYNTAX:

```
        ptr=(caste-type*) calloc (n, size);
```

EXAMPLE:

```
         ptr=(int*) calloc (5, sizeof(int));
```

REALLOC ()

Realloc () is also known as reallocation. It reallocates the memory occupied by malloc () or calloc () functions.

SYNTAX:

```
           ptr=(caste-type*) realloc (ptr, newsize);
```

EXAMPLE:

```
          ptr=(int*) malloc (15);
          ptr=(int*) realloc(ptr,30);
```

FREE ()

It is used to deallocate the memory.it frees the dynamic allocated memory.

2.6) STRUCTURE

In c program, struct is a collection of different /unsimilar data types.

SYNTAX

```
    struct tag_name
{
 type mem 1;
type mem 2;
};
```

Where

Tag_name = structure name

Type = data type (int, char)

Mem 1, mem 2 = members of the structure

EXAMPLE

```
  struct student
{
 char name [30];
char roll_number [30];
int marks;
};
```

SELF REFERENTIAL STRUCTURE

It is a structure with different types of members, in which at least one member is pointing to itself.

EXAMPLE

```
 struct slist
{
char name [30];
char father_name [30];

int sslc_roll;

int pu_roll;

Int sslc_percent;

int pu_percent;

char caste [30];

char state [30];

Int priority;

struct slist *ptr;

};
```

CHAPTER 3

ALGORITHM

Step 1: Insert and declare all header files which are required for this project.

Step 2: declare all variables which are required for project.

Step 3: create a structure to store the details of customer.

Step 4: Next we use while function and switch case for displaying few choices like

- add records
- list records
- modify records
- delete records
- payment
- exit

Step 5: we use files for storing details of customers.

Step 6: we will have separate functions for every record.

Step 7: user will choose from the above options according to his choice.

Step 8: add record is used to add record to the database. Similarly, we have function for modifying records, listing records, deleting records and payment.

Step 9: finally, after customer is done with his payment he exits from the system.

CHAPTER 4

IMPLEMENTATION

Program starts with a switch case that includes

- add records
- list records
- modify records
- search records
- delete records
- payment

4.1. Add records ()

Here we can add records by giving some details like

- customer name
- customer phone number
- amount to be paid

So, by entering the above details a record will be added to the data base.

4.2. Delete record ()

Using this function, we can delete a record from the data base by entering the customer phone number that is been added to the data base.

By entering a registered phone number, we can delete records of that particular customer.

4.3. Search record ()

Using this function, we can search a record of a customer from the data base. We need to enter phone number of a customer. After entering the phone number, the system checks for the number in the database, if the number is found then we can display the data.

4.4. List records ()

Using this function, we can delete a record of a customer from the data base. We need to enter phone number of a customer. After entering the phone number, the system checks for the number in the database, if the number is found then we can delete the data.

4.5. Modify records ()

Using this function, we can modify a record of a customer from the data base. We need to enter phone number of a customer. After entering the phone number, the system checks for the number in the database, if the number is found then we can modify the data.

4.6. Payment ()

Using this function, we can make payment of the customer. We need to enter phone number of a customer. After entering the phone number, the system checks for the number in the database if the customer has an account then it displays the amount to be paid along with the bill.

Exit

Finally, after the customer is done with his/her payment he can choose the exit option to exit.

Chapter 6:

Conclusion

Firstly, I would like to thank NEW HORIZON COLLEGE OF ENGINEERING for giving me a great opportunity to enhance my knowledge. Also hearty thanks to cs department teachers and reviewers.

This report deals with the project of “TELECOM BILLING”. This project is developed to make online phone payments. This project was completed successfully. This projected is developed to make payments easier. This system has been developed with lot of care and free of errors and at the same time it is efficient and less time consuming.

I thank my reviewer Ms. Heyshanthini ma’am for helping me to complete this project. I have learned many things about data structures especially linked list through this project.