TecSaCon
We Build your Career

Tecsacon Technologies

IT Training | Placements

TECSACON TECHNOLOGIES

MAINFRAME PROJECT REPORT CASE STUDY REPORT

ON

FINANCE BILLING SYSTEM

Submitted in partial fulfillment for the certification of

MAINFRAME APPLICATIONS PROGRAMMING

BY

GOPI P [Batch No: TTM106]

PREETHA K [Batch No: TTM106]

Under the guidance of

MR RAMAMOORTHY V

TECSACON TECHNOLOGIES.

11/1, Second Floor, 100 feet Road, BTM Layout 2nd Stage, Bangalore - 560076 .

ABSTRACT

The electricity billing system project is developed in CICS as Frontend VS COBOL II as programming language and DB2 as backend. The main aim of the project report is to maintain customer details and generate bill to Money lender daily and monthly report for record maintenance.

The administrator or employee should enter the username and password to enter into the system. If the username and password are correct, it will login to the project.If the entered details are invalid it will show error message.

The money lender in the finance service system are entered by the employee. The entered details can be deleted and update if any change is need.The bill is prepared automatically according to the Money lender and amount is stored to the system.

When the money lender making payment it will be automatically update to the database. The bill will be generated for the money lender. The employee details are entered and managed by the administrator. The required are generated to the administrator and employee.

ACKNOWLEDGMENT

I consider it’s my privilege to express gratitude and respect to all those who guided and helped me for this completion of project case study.

I personally express my gratitude with respect and sincere thanks to our honorable director MR RAMAMOORTHY V for his cooperation and motivated me saying that, it was possible of doing this project.

I personally thank my all seniors, lab coordinator, juniors who have pointed out the loop holes in my case study and hat off all mainframe websites which helps me a lot in clearing the doubts.

This leaf of acknowledgement would not be complete without a special word of thanks to my well-wisher’s moral support, encouragement and love, which enable me to successfully bring out this project report.

TABLE OF CONTENT

1.PREAMBLE

1.1 INTRODUCTION

The project Finance Service Billing System aims at serving the department of loan by computerizing the billing system. It mainly focuses on the calculation of loan taken during the specified time and the money to be paid to finance offices. This computerized system will the overall billing system easy, accessible, comfortable and effective for money lender.

1.2 EXISTING AND PROPOSED SYSTEM

The conventional system of finance service billing system is not so effective; one staff has to visit each customer's house to note the loan records and collect the data. Then another staff has to compute the consumed loan and calculate the interest and the money to be paid. Again, the bills prepared are to be delivered to money lender. Finally, individual customers has to go to finance office to pay their dues.

Hence, the conventional finance billing system is uneconomical, requires many staff to do simple jobs and is a lengthy process overall. In order to solve this lengthy process of billing, finance service billing based computerized system is essential. This proposed finance billing system project overcome all these drawbacks with the features a for mentioned. It is beneficial to both money lender and the company which provides loan.

With the new system, there is reduction in number of staffs to be employed by the company. The working speed and the performance of the software is faster with high performance which saves time. Furthermore there is very little chance of miscalculation and being corrected by the staffs.

1.3 MODULE DESCRIPTION

The system consists of three main modules they are:

- ➢ Login
- ➢ Admin Operations
- ➢ User operations

1.3.1 LOGIN

There are two actors namely,

- ➢ Admin login
- ➢ User login

As per input the map will be redirected either to admin operations map or user operations map.

1.3.2 ADMIN OPERATIONS

There are three sub-modules. They are,

- ✓ New loan

- ✓ Loan cancellation
- ✓ Monthly report generation

NEW LOAN

This is the first option for an admin to create a new money lender with all personal and connection details. Here unique form id will be given to the new money lender and all details will be entered to that form id and on submissions unique customer id will be generated by the system itself. This customer id will be used for his/her reference in further operations.

CONNECTION TERMINATION

In this module, admin can cancel the loan provided to the customer if any violations are done by the customer. By entering the unique customer id, we can fetch all the details about the customer. By entering the status and the reason for termination admin has previlage to terminate a connection.

MONTHLY REPORT GENERATION

In this module,admin gets to know the total collection for a month. By entering the month and the year further on by submission the total collection for that particular month will be generated.

1.3.3 USER OPERATIONS

There are three sub-modules.They are

- ✓ Bills generation
- ✓ Problem Registration
- ✓ Daily collection Report Generation

BILLS GENERATION

In this module, the bills will be generated for the customer. By entering customer id and the consumed units the system will calculate the total amount that is to be paid and loan amount will be generated for the customer.

PROBLEM GENERATION

This module allows users to register a complaint. By entering the customer id and by specifying the status and the description of the problem thus registering the complaint for the further actions.

DAILY COLLECTION REPORT GENERATION

In this module, user gets to know the total collection for a day. By entering the day, the month and the year further on by submission the total collection for that particular day will be generated.

2. SYSTEM ENVIRONMENT

The system and the software specification, which are been used in this project case study are as follows.

2.1 SOFTWARE USED

System OS	:	Windows 8.1
Terminal emulator	:	MOCHA 3270
Mainframe Server	:	Live Server from US
Mainframe OS	:	z/OS 1.10

2.2 MODULES INCLUDED

Front-end	:	CICS
Back-end	:	DB2
Programming language	:	COBOL
Compiler	:	JCL Program (COBOL-DB2-CICS Compiler)

3. CASE STUDY SURVEY

3.1 COBOL

COBOL is a compiled english-Like computer programming language designed for business use. COBOL is primarily used in business, finance and administrative systems for companies and governments. COBOL is still wieldy used in legacy applications deployed on mainframe computers, Such as large-scale batch and transaction processing jobs.

COBOL code is split into four divisions (identification, environment, data and procedure) containing a rigid hierarchy of sections, paragraphs and sentences, Lacking a large standard Library, the standard specifies 43 statements, 87 functions and just one class.

NAME	COLUMNS	USAGE
Sequence Number Area	1-6	Originally used for card/line Numbers, This area is ignored by the compiler
Indicator Area	7	The following characters are allowed here: • * - a commant line • / - a comment line that will be printed on a new page of a source listing • - - a continuation line, where words or literals from the previous line are continued • D - a line enabled in finance billing bugging mode,which is otherwise ignored
Area A	8-11	This contains: DIVISIONS, SECTION and PROCEDURE headers; 01 and 77 level numbers and file/report descriptors
Area B	12-72	Any other code not allowed in Area A
Program Name area	73-80	Historically up to column 80 for punched cards, it is used toidentify the program or sequence the card belongs to

IDENTIFICATION DIVISION

The IDENTIFICATION DIVISION identifies the following code entity and contains the definition of a class or interface.

ENVIRONMENT DIVISION

The ENVIRONMENT DIVISION contains the configuration section and the input-output section. The configuration section is used to specify variable features such as currency signs, locales and character sets. The input-output section contains file related informations.

DATA DIVISION

The DATA DIVISION is split into six sections which declare different items: the file section, for file records; the WORKING STORAGE SECTION, for static variables; the local-storage section, for automatic variables; the linkage section, for parameters and the return value; the report section and the screen section, for text-based user interfaces.

DATA TYPES

Standard COBOL provides the following data types:

Data type	**Sample declaration**	**Notes**
Alphabetic	PIC A(30)	May only contain letters and spaces
Alphanumeric	PIC X(30)	May contain any characters
Boolean	PIC 1 USAGE BIT	Data stored in the form of 0s and 1s, as a binary number
Index	USAGE INDEX	Used to reference table elements
National	PIC N(30)	Similar to alphanumeric, but using an extended character set, e.g. UTF-8
Numeric	PIC 9(5) V9(5)	May contain only numbers
Object	USAGE OBJECT REFERNCE	May reference either an object or NULL
Pointer	USAGE POINTER	

PICTURE CLAUSE

A PICTURE (OR PIC) clause is a string of characters, each of which represents a portion of the data item and what it may contain. Some picture characters specify the type of the item and how many characters or digits it occupies in memory. For example, a 9 indicates a decimal digit, and an S indicates that the item is signed. Other picture characters (called insertion and editing characters) specify how an item should be formatted. Picture specifications containing only digit (9) and sign (S) characters define purely numeric data items, while picture specifications containing alphabetic (A) or alphanumeric (X) characters define alphanumeric data items. The presence of other formatting characters define edited numeric or edited alphanumeric data items.

USAGE CLAUSE

The USAGE clause declares the format data is stored in. Depending on the data type, it can either complement or be used instead of a PICTURE clause. While it can be used to declare pointers and object references, it is mostly geared towards specifying numeric types. These numeric formats are:

- Binary, where a minimum size is either specified by the PICTURE clause or by a USAGE clause such as BINARY-LONG.
- USAGE COMPUTATIONAL, where data may be stored in whatever format the implementation provides, often equivalent to USAGE BINARY
- USAGE DISPLAY, the default format, where data is stored as a string.
- Floating-point, in either an implementation-dependent format or according to IEEE 754.
- USAGE NATIONAL, where data is stored as a string using an extended character set
- USAGE PACKED-DECIMAL, where data is stored in the smallest possible decimal format (typically packed binary-coded decimal)

PROCEDURE DIVISION

PROCEDURES

The sections and paragraphs in the procedure division (collectively called procedures) can be used as labels and as simple subroutines. Unlike in other divisions, paragraphs do not need to be in sections.Execution goes down through the procedures of a program until it is terminated. To use procedures as subroutines, the PERFORM verb is used. This transfers control to be specified range of procedures and returns only upon reaching the end.

Unusual control flow can trigger mines, which cause control in performed procedures to return at unexpected times to unexpected locations. Procedures can be reached in three ways: they can be called with PERFORM , jumped to from a GO TO or through execution " falling through" the bottom of an above paragraph. Combinations of these invoke undefined behaviour, creating mines. Specifically, mines occur when execution of a range of procedures would cause control flow to go pass the last statement of a range of procedures already being performed.

STATEMENTS

COBOL 2014 has 47 statements (also called verbs), which can be grouped into the following broad categories: control flow, I/O, data manipulation and the report writer. The report writer statements are covered in the report writer section.

CONTROL FLOW

COBOL'S conditional statements are IF and EVALUATE. EVALUATE is a switch-like statement with the added capability of evaluating multiple values and conditions. This can be used to implement decision tables.

The PERFORM statement is used to define loops which are executed until a condition is true (not while, unlike other languages). It is also used to call procedures or ranges of procedures (see the procedure section for more details). CALL and INVOKE call subprograms and methods, respectively. The name of the subprogram/method is contained in a string which may be a literal or a data item. Parameters can be passed by reference, by content (where a copy is passed by reference) or by value (but only if prototype is available). Cancel unloads subprograms from memory. GO TO causes the program to jump to a specified procedure.

The GO BACK statement is a return statement and the STOP statement stops the program. The EXIT statement has six different formats: it can be used as a return statement, a break statement, a continue statement, an end marker or to leave a procedure.

Exceptions are raised by a RAISE statement and caught with a handler, or declarative, defined in the DECLARATIVES portion of the procedure division. Declaratives are sections beginning with a USE statement which specify the errors to handle. Exceptions can be names or objects. RESUME is used in a declarative to jump to the statement after the one that raised the exception or to a procedure outside the DECLARATIVES. Unlike other languages, uncaught exceptions may not terminate the program and the program can proceed unaffected.

I/O

File I/O is handled by the self-describing OPEN, CLOSE, READ and WRITE statements along with a further three: REWRITE, which updates a record; START, which select subsequent records to access by finding a record with a certain key; and UNLOCK, which releases a lock on the last record accessed.

User interaction is done using ACCEPT and DISPLAY.

DATA MANIPULATION

The following verbs manipulate data:

- INITIALIZE, which sets data items to their default values.
- MOVE, which assign values to data items.
- SET, which has 15 formats: it can modify indices, assign object references and alter table capabilities, among other functions.
- ADD, SUBTRACT, MUTIPLY, DIVISION and COMPUTE, which handle arithmetic (with COMPUTE assigning the result of a formula to a variable).
- ALLOCATE and FREE, which handle dynamic memory.
- VALIDATE, which validates and distributes data as specified in an item’s description in the data division.
- STRING and UNSTRING, which concatenate and split strings respectively.
- INSPECT, which tallies or replaces instances of specified substrings within a string.
- SEARCH, which searches a table for the first entry satisfying a condition.

Files and tables are sorted using SORT and MERGE verb merges andsort files.The RELEASE verb provides records to sort and RETURN retrieves sorted records in order.

SCOPE TERMINATION

Some statements, such as IF and READ, may themselves contain statements. Such statements may be terminated in two ways: by a period (implicit termination), which terminates all unterminated statements contained, or by a scope terminator, which terminates the nearest matching open statement.

3.2 DB2

DB2 is a database product from IBM. It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyse nd retrieve the data efficiently. In DB2 Database tables, each column has its own data type depending on developer's requirements. The data type is said to be type and range of the values in columns of a table.

Built-in data types

- ❖ DATETIME
  - TIME: It represents the time of the day in hours, minutes and seconds.
  - TIMESTAMP: It represents seven values of the date and time in the form of year, month, day, hours, minutes, seconds and microseconds.
  - DATE: It represents date of the day in three parts in the form of year, moth and day.
- ❖ STRING
  - Character
- ❖ CHAR (fixed length): Fixed length of Character Strings.
  - Varying length
- ❖ VARCHAR: Varying length character strings.
- ❖ CLOBL: Large object strings, you use this when a character string might exceed the limits of the VARCHAR data type.
  - Graphic
- ❖ GRAPHIC
  - Fixed length: Fixed length graphic strings that contains double-byte characters
  - Varying length
- ❖ VARGRAPHIC: Varying character graphic string that contains double byte characters.
- ❖ DBCLOB: Large object type
  - Binary
- ❖ BLOB (varying length): Binary string in large object

- BOOLEAN: In the form of 0 and 1
- SIGNED NUMERIC
    - Exact
- BINARY INTEGER
    - SMALLINT [16BIT]: Using this you can insert small int values into columns
    - INTEGER [32BIT]   : Using this you can insert large int values into columns.
    - BIGINT [64BIT]   : Using this you can insert larger int values into columns.
- DECIMAL
    - DECIMAL (packed)
    - DECFLOAT (decimal floating point): Using this, you can insert decimal floating point numbers.
    - Approximate
- FLOATING POINTS
    - REAL (single precision): Using this data type, you can insert single precision floating point numbers.
    - DOUBLE (double precision): Using this data type, you can insert double precision floating point numbers.

TABLES

The following syntax creates table:

Syntax: [To create a new table]

```
Create table <schema_name> <table name>

(column_namecolumn_type.....) in <tablespace_name>
```

Inserting data values in table

The following syntax inserts values in the table:

```
Insert into<table_name>(col 1,col 2,...)

Values(val 1,val2,....)
```

Retrieving values from table

The following syntax retrieves value from the table:

Syntax: [to retrieve value from a table]

```
Select *from<tab_name>
```

Retrieving values from a table including hidden columns

The following syntax retrieves values from selected columns:

Syntax: [to retrieve selected hidden columns value from a table]

```
Select col1, col2, col3 from<tab_name>
```

Altering the type of table columns

You can modify our table structure using this “alter” command as follows:

Syntax :

```
Alter table <tab_name>after column <col_name> set data type <data_type>
```

Altering column name

You can change column name as shown below:

Syntax:[To modify the column name from old name to new name of a table]

```
Alter table <tab_name>rename column<old_name> to <new_name>
```

Dropping the tables

To delete any table ,you need to use the “DROP” command as follows:

Syntax :

Drop table<tab_name>

INDEX

Index is a set of pointers, which can refer to rows in a table, blocks in MDC or ITC tables, XML data in an XML storage object that are logically ordered by the values of one or more keys. It is created on DB2 table columns to speed up the data access for the queries, and to cluster and partition the data efficiently. It can also improve the performance of operation on the view. A table with a unique index can have rows with unique keys. Depending on the table requirements, you can take different type of indexes.

TYPE OF INDEXES

- Unique and Non-Unique indexes
- Cluster and Non-clustered indexes

Creating idexes

For creating unique indexes, you use following syntax:

Create unique index <index_name> on

<table_name> (<unique_column>) include (<column_names...>)

Dropping indexes

For dropping the index, you can use the following syntax

Drop unique index<index_name> on

<table_name> (<unique_column>) include (<column_names,...>)

VIEW

A view is an alternative way of represnting the data stored in the tables. It is not an actual table and it does not have any permanent storage. View provides a way of looking at the data in one or more tables. It is a nemed specification of a result table.

Creating a view

You can create a view using th following syntax:

Syntax:

```
Create view <view_name>(<col_name>,<col_name1.....) as select<cols>..
From<table_name>
```

Modifying a view

You can modify a view using the following syntax:

```
Alter view<view_name> alter <col_name>
Add scope<table_or_view_name>
```

Dropping the view

You can drop a view using the following syntax:

```
Drop view <view_name>
```

To enforce database integrity, a set of rules is defined, called constraint. The constraints either permit or prohibit the values in the columns.

CONSTRAINTS

In a Real time database activities, the data should be added with certain restrictions. For example, in a sales database, sales-id or transaction-id shoud be unique. The constraints types are:

- Unique
- Not null
- Primary key
- Foreign key
- Check

Constraint are only associated with tables. They are applied to only particular tables. They are defined and applied at the time of table creation.

NOT NULL

It is rule to prohibit null values from one or more columns within the table.

Syntax:

```
Create table <table_name>(col_namecol_typenotnull)
```

UNIQUE CONSTRAINTS

Using these constraints, you can set values of columns uniquely. For this, the unique constraints are declared with “not null” constraints at the time of creating table

Syntax:

```
Create table <table_name>(<col><col_type>not null unique,....)
```

PRIMARY KEY

Similar to the unique constraints, you can use a “primary key” and a “foreign key” constraint to declare relationship between multiple table.

Syntax:

```
Create table<tab_name>(,....,primary key () )
```

FOREIGN KEY

A foreign key is a set of columns in a table which are required to match at least one primary key of a row in another table. It is a relational constraint or referential integrity constraint. It is a logical rule about values in multiple columns in one or more tables. It enables required relationship between the tables.

Syntax:

```
Create table<tab_name>(<col><col_type>,constraint
<const_name> foreign key (<col_name>)
References <ref_table>(<ref_col>)
```

CHECKING CONSTRAINTS

You need to use this constraint to add conditional restrictional for a specific column in a table.

Syntax:

```
Create table (primary key (),
Constraint check (condition or condition))
```

Dropping the constraint

Let us see the syntaxes for dropping various constraint.

Dropping UNIQUE constraint

Syntax:

```
Alter table<tab_name> drop unique<const_name>;
```

Dropping PRIMARY KEY

Syntax:

```
Alter table <table_name> drop primary key;
```

Dropping CHECK CONSTRAINT

Syntax:

```
Alter table <tab_name>drop check<check_const_name>;
```

Dropping FOREIGN KEY

Syntax:

Alter table<tab_name>drop foreign key <foreign_key_name>;

3.3 JCL

Jcl is used in a mainframe environment to act as a communication between a program (Example: cobol, assembler or PL/1) and the operating system. In a mainframe environment, programs can be executed in batch and online process mode. Example of a batch system can be processing the bank transactions through a VSAM (Virtual Storage Access Method) File and applying it to corresponding accounts.

Batch and online processing differ in the aspect of input, output and program exection request. In batch processing, aspects are fed into a jcl which is in turn received by the operating system.

COMPILING COBOL PROGRAMS

In order to execute a cobol program in batch mode using JCL, the program needs to be compiled and a load module is created with all the sub-program. The JCL uses the load module and the actual program at the time of execution. The load libraries are concatenated and given to the time of execution using JCLLIB or STEPLIB>

3.4 CICS

CICS is a DB/DC system which is used in online application. CICS was developed because batch operating system can executed only batch programs. CICS programs can be written in COBOL, C, C++, JAVA, etc. These days, users want information within seconds. So as to achieve this goal we need a system which can process information online. CICS allows users to communicate with back-end system to get the desired information. Example of online programs are online banking system, flight reservation, etc. Following images shows how a all the components are related to CICS:

FUNCTIONS OF CICS

The main functions performed by CICS in an application are as follow:

- CICS manage request from concurrent users in an application.
- Although, multiple users are working on CICS system but it gives a feel user that he is the single user only.
- CICS give the access to data files for reading or updating them in an application.

FEATURES OF CICS

The features of CICS are as follows

CICS is an operating system in itself as it manages its own processor storage, has its own task manager which handles execution of multiple programs, provides its own file management functions.

- CICS provides on-line environment in batch operating system. Jobs submitted are executed immediately.
- CICS is a generalized transaction processing interface.
- It is possible to have 2 or more CICS regions at the same time as CICS runs as a batch job operating system at the back-end.

4.PROJECT STRUCTUR

4.1 WORKFLOW DIAGRAM: FINANCE BILLING SYSTEM

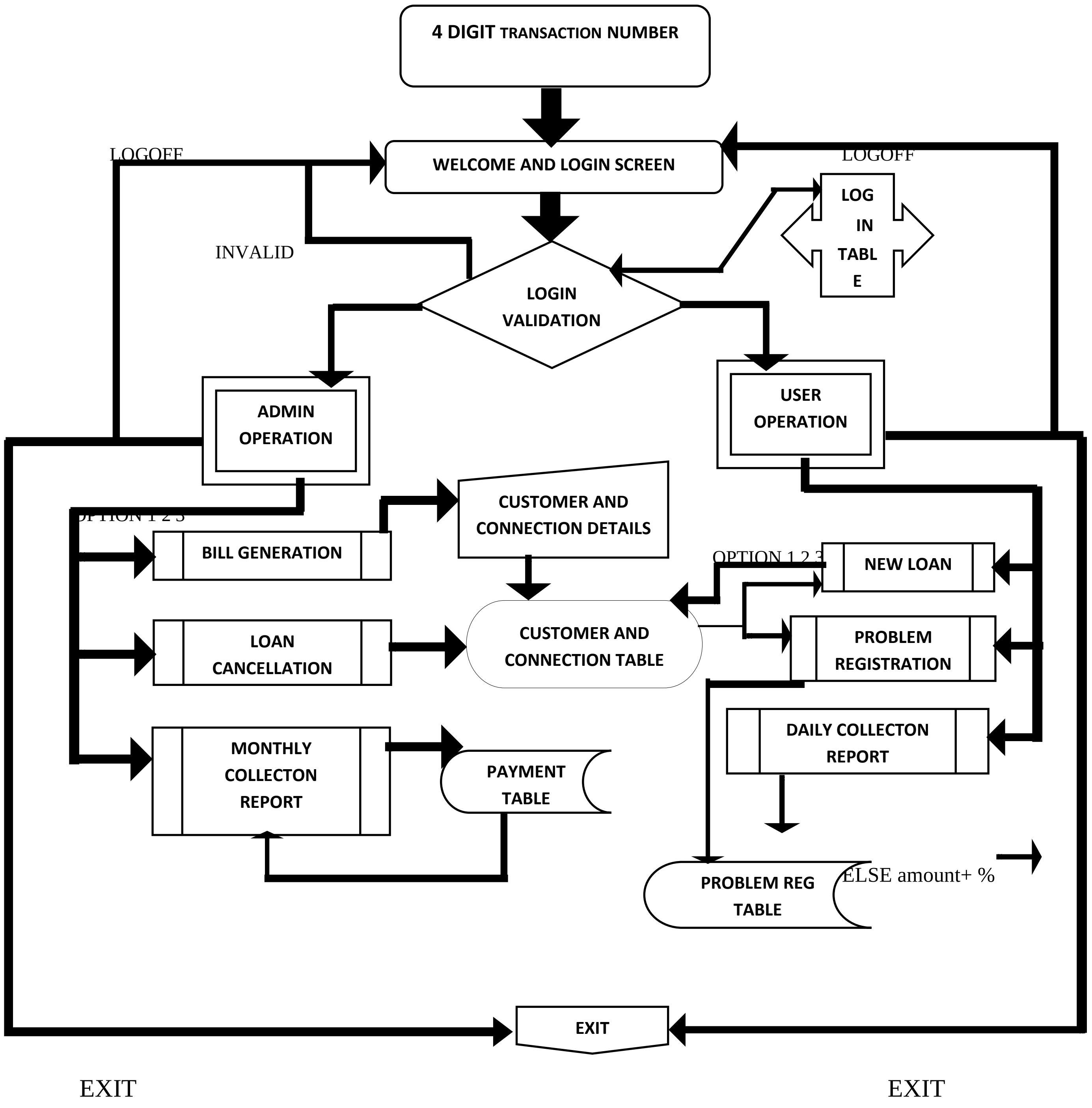


Figure 4.1 architecture flow of finance billing system case study project

Simple architecture shows the general work process of finance billing system case study, where we can analysed flow and connectivity of maps with database.

4.2 DATABASE DESIGN DIAGRAM

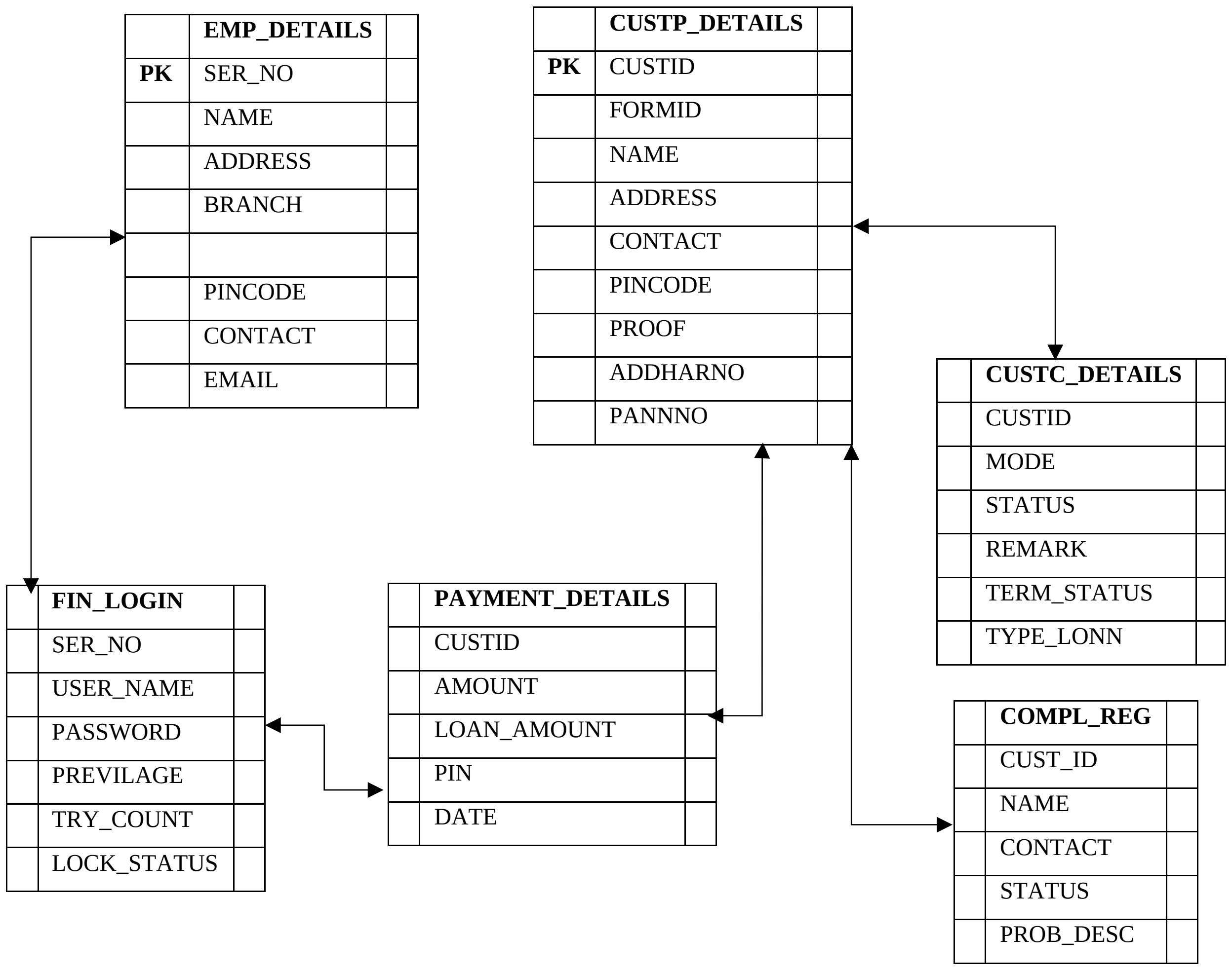


Figure 4.2 finance billing system db2 design diagram

The above diagram shows the general connectivity of table and relationships of the table and tables structure used for the case study project.

5. DATABASE

5.1 LOGIN TABLE

This table which contain login details along with privilege of the user.

SNO	COLUMN	WIDTH	DATA TYPE	CONSTRAINT
1	SER_ID	5	CHAR	FOREIGN KEY
2	USERN	10	CHAR	UNIQUE
3	PASSW	5	INTEGER	NOT NULL
4	PRIVILEGE	1	CHAR	NOT NULL

5.2 EMPLOYEE INFORMATION TABLE

This contain information about personal information about the employee of the organization.

SNO	COLUMN	WIDTH	DATA TYPE	CONSTRAINT
1	SER_ID	5	CHAR	PRIMARY KEY
2	NAME	15	CHAR	NOT NULL
3	EADDRESS	40	CHAR	NOT NULL
4	BRANCH	10	CHAR	NOT NULL
5	PINCODE	6	INT	NOT NULL
6	CONTACT	10	INT	UNIQUE
7	EMAIL	30	CHAR	UNIQUE

5.3 CUSTOMER PERSONAL DETAILS

This table which maintains information about the customers and it has a primary key i,e. CUSTID for faster access.

SNO	COLUMN	WIDTH	DATA TYPE	CONSTRAINT
1	CUSTID	6	CHAR	PRIMARY KEY
2	FORMID	6	CHAR	UNIQUE
3	NAME	15	CHAR	NOT NULL
4	ADDRESS	40	CHAR	NOT NULL
5	CONTACT	10	INT	UNIQUE
6	PINCODE	6	INT	NOT NULL
7	PROOF	5	CHAR	NOT NULL
8	AADHARNO	15	CHAR	UNIQUE
9	PANNO	10	CHAR	UNIQUE

5.4 CUSTOMER CONNECTION INFORMATION TABLE:

This table is used to maintain the customers connection information which contains CUSTID as foreign key that refer to customer information table.

SNO	COLUMN	WIDTH	DATA TYPE	CONSTRAINT
1	CUSTID	6	CHAR	FOREIGN KEY
2	MODE	2	INT	NOT NULL
3	TYPELONN	10	CHAR	NOT NULL
4	CSTATUS	10	CHAR	NOT NULL
5	REMARK	40	CHAR	NOT NULL
6	TERM_S	10	CHAR	NOT NULL

5.5 PROBLEM REGISTRATION TABLE:

The purpose of this table is to maintain those power problems thet are faced by the customer.

SNO	COLUMN	WIDTH	DATA TYPE	CONSTRAINT
1	CUSTID	6	CHAR	FOREIGN KEY
2	NAME	15	CHAR	NOT NULL
3	CONTACT	10	INT	NOT NULL
4	STATUS	10	CHAR	NOT NULL
5	PROB_DESC	40	CHAR	NOT NULL

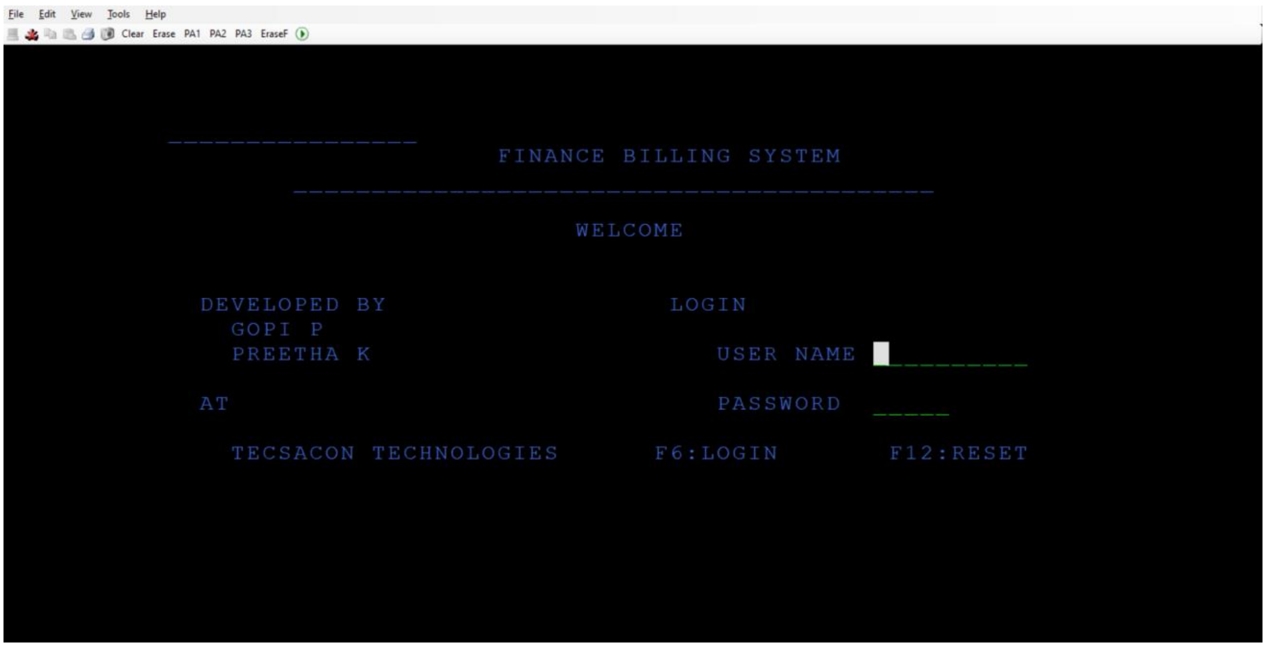
5.6 PAYMENT TABLE:

Table which contains information about customer payment details and it has CUSTID as a foreign key.

SNO	COLUMN	WIDTH	DATA TYPE	CONSTRAINT
1	CUSTID	6	CHAR	FOREIGN KEY
2	AMOUNT	8,2	DECIMAL	NOT NULL
3	LOAN	10	INT	NOT NULL
4	PIN	6	INT	NOT NULL
5	CDATE		DATE	NOT NULL
6	SER_NO	5	CHAR	NOT NULL

7.PROJECT SCREENSHOTS

7.1 WELCOME AND LOGIN SCREEN

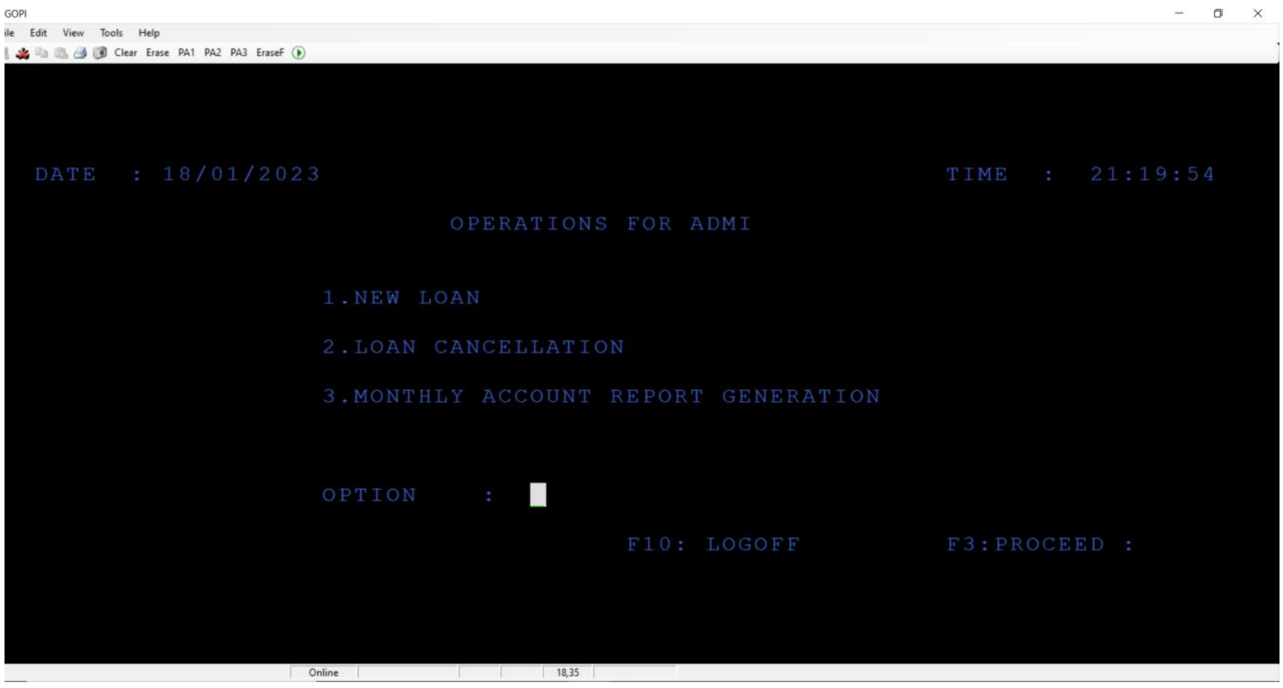


```
                    FINANCE BILLING SYSTEM

                          WELCOME

  DEVELOPED BY                     LOGIN
    GOPI P
    PREETHA K                         USER NAME __________

  AT                                  PASSWORD  _____

    TECSACON TECHNOLOGIES      F6:LOGIN        F12:RESET
```

Figure 7.1 welcome screen

After entering transaction ID this map will apppear, employee or admin needs to enter user name and password to enter in to the system, according to the username and password the user will be redirect to admin or employee operations menu.

7.2 ADMIN OPERATION MENU

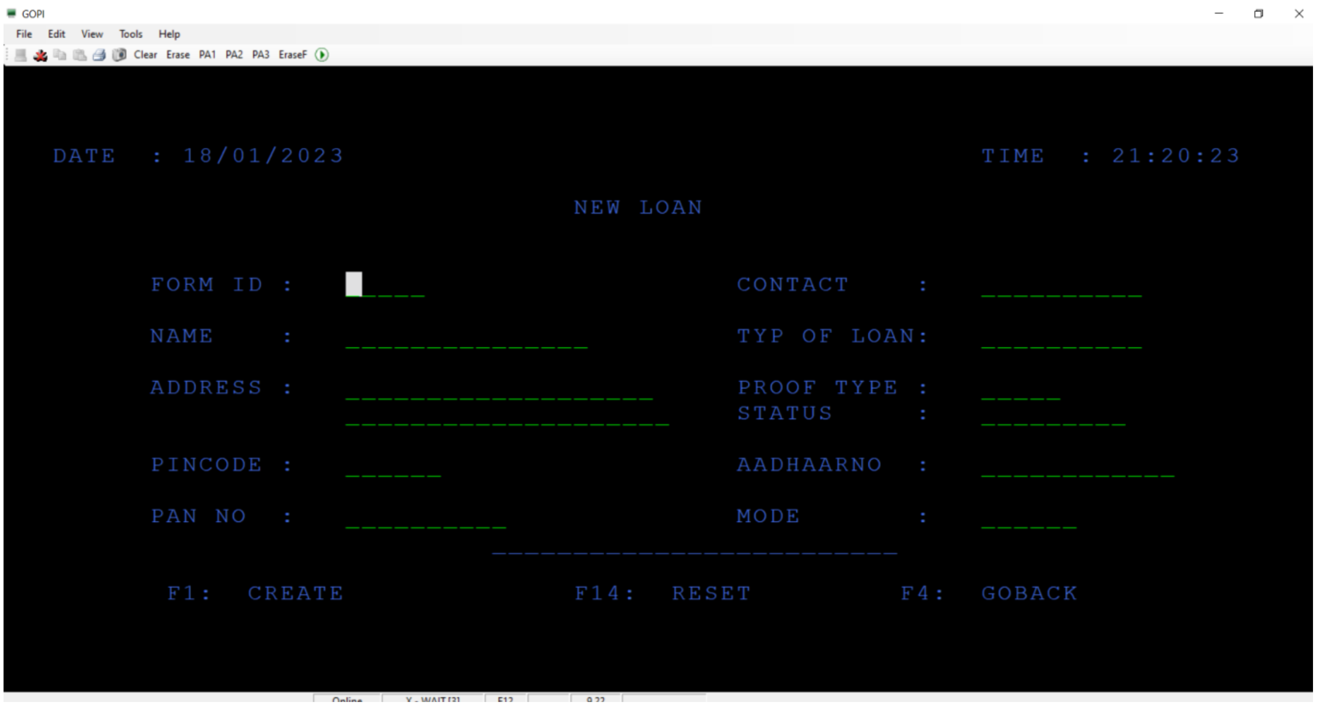


```
GOPI
ile  Edit  View  Tools  Help
Clear  Erase  PA1  PA2  PA3  EraseF

DATE   : 18/01/2023                                  TIME   :  21:19:54
                        OPERATIONS FOR ADMI

                 1.NEW LOAN

                 2.LOAN CANCELLATION

                 3.MONTHLY ACCOUNT REPORT GENERATION


                 OPTION     :
                                   F10: LOGOFF         F3:PROCEED :

Online                18,35
```

Figure7.2 Admin operation screen

After login to the system using username and password if the actor is admin then this screen will appears.

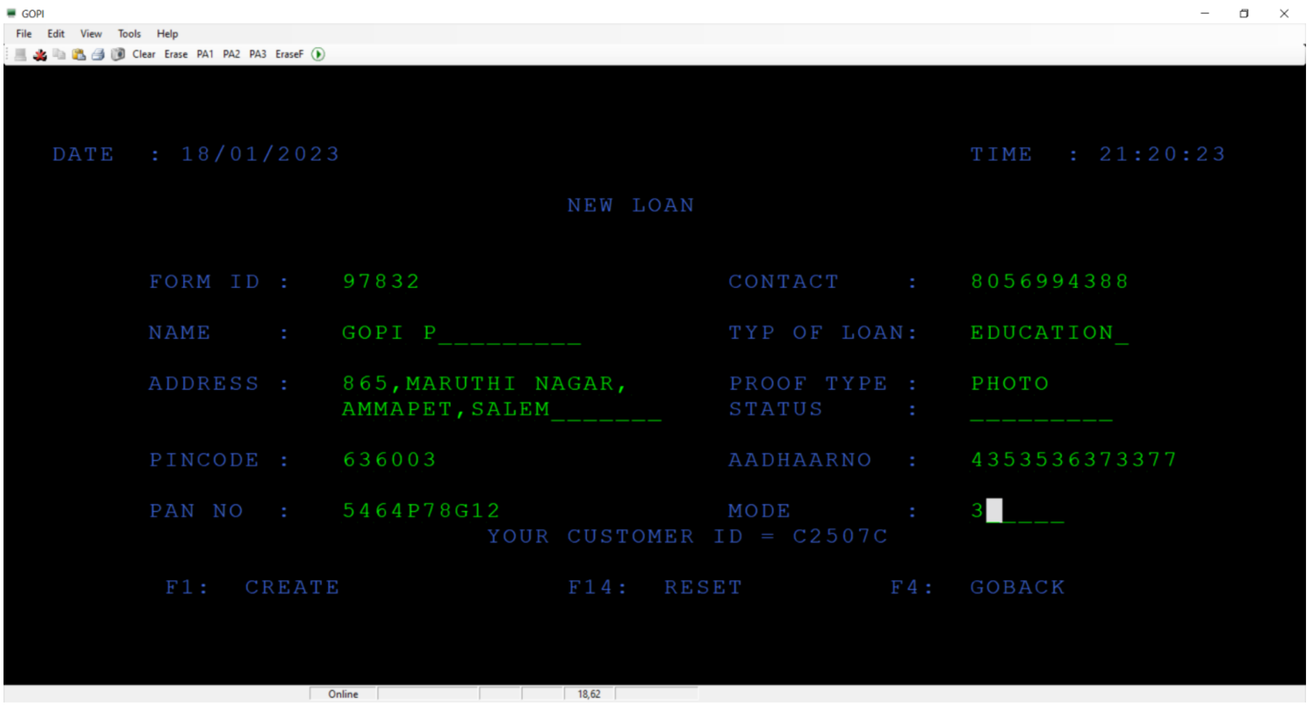
7.3 ADMIN-NEW LOAN CREATION



```
DATE   : 18/01/2023                                        TIME   : 21:20:23

                              NEW LOAN

      FORM ID :    ____                    CONTACT     :   __________

      NAME    :    _______________         TYP OF LOAN:    __________

      ADDRESS :    ___________________     PROOF TYPE :    _____
                   ____________________    STATUS     :    _________

      PINCODE :    ______                  AADHAARNO  :    ____________

      PAN NO  :    __________              MODE       :    ______
                            _________________________

       F1:  CREATE                F14:  RESET          F4:  GOBACK
```

Figure7.3 New loan creation screen

If the option is one this screen will appears.



```
DATE   : 18/01/2023                                        TIME   : 21:20:23

                              NEW LOAN

      FORM ID :    97832                   CONTACT     :   8056994388

      NAME    :    GOPI P_________         TYP OF LOAN:    EDUCATION_

      ADDRESS :    865,MARUTHI NAGAR,      PROOF TYPE :    PHOTO
                   AMMAPET,SALEM_______    STATUS     :    _________

      PINCODE :    636003                  AADHAARNO  :    4353536373377

      PAN NO  :    5464P78G12              MODE       :    3____
                          YOUR CUSTOMER ID = C2507C

       F1:  CREATE                F14:  RESET          F4:  GOBACK
```

Figure 7.4 New loan screen with data

After entering the new loan lender details user to press F1. Id will be generated for the new lender.