# Kiosk Transaction Management & Utility Bills Collection System

**Aleya Khatun**

A dissertation submitted in fulfillment of the requirements for the

Degree of

**Master of Science in Computer Science and Engineering**

**Department of Computer Science and Engineering**

**East West University, Bangladesh**

May 14, 2015

## Abstract

Traditional approaches in various banks to manage transaction after kiosk operation still on papers. Inter branch utility bill also collected in a manual way. Calculate summery on records, separate bill type, find out mismatch transaction etc. managed in manual way. These are time consuming, and inefficient system leading to problems of inefficient utilization of resources and problems to bank user.

Kiosk transaction Management & Utility Bills Collection System is a comprehensive, integrated information system designed to manage transaction after kiosk operation in bank such as find out auto mismatch transaction, auto calculate summery on bill amount, vat amount, generate different type wise report, reduce inter branch ibca transaction etc.

This software provides massive, integrated systems that support the bank to manage kiosks transaction and inter branch utility bills transaction in efficient way. Banks are becoming more reliant on the ability Kiosk transaction Management & Utility Bills Collection System for better solution and improved services and practices.

The manual handling of the record is time consuming and highly prone to error. This software also store transaction in database and reduce time to produce different type wise report. This software reduces manual work and intervention, reduce time and increase efficiency.

# Declaration

This is certified that this project work is an original work and is done by me. Neither it nor part of it has been submitted for the requirement of any degree or diploma or for any other purposes except for publication.

Signature of the candidate

________________________

Aleya Khatun

# Letter of Acceptance

The project entitled Kiosk Operation Management & Utility Bills Collection System submitted by Aleya Khatun, ID No: 2013-3-96-011 to the Department of Computer Science and Engineering, East West University, Dhaka 1212, Bangladesh is accepted by the Department for the partial fulfillment of requirements for the degree of MS in Computer Science and Engineering on 14thMay - 2015.

**BOARD OF EXAMINERS**

1. ______________________________ Supervisor

**Shaila Sharmeen**

Senior Lecturer

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

2. ______________________________

Chairperson

**Dr. Shamim Hasnat Ripon**

Associate Professor &Chairperson

Department of Computer Science and Engineering

East West University

Dhaka, Bangladesh

## Acknowledgements

Here, I would like to express my feelings to those who have offered and supported me to complete this project work.

Firstly, I would like to convey my heartiest thanks and insightful indebtedness to my supervisor **Shaila Sharmeen** for her valuable contribution, constant guidance, intuitive advice, helpful criticism, valuable suggestions, commendable support, and endless patience for the completion of this project work. I am very much grateful to her and feel very proud to have worked with her because it was not possible for me to complete this work without her inspiring enthusiasm and encouragement.

Special thanks go to Monzur Morshed Khan, EO, Programmer, Standard Bank Limited and also Special thanks should be given to my friends who helped me in many ways. Finally, words alone cannot express the thanks my Parents for them encouragement and assistance.

## Abbreviations & Acronyms

| **Abbreviations** | **Description** |
|---|---|
| SQL | Structure Query Language |
| PL/SQL | Procedural Language/ Structure Query Language |
| Oracle Forms | Oracle Formss |
| SDLC | Software Development Life Cycle |

## Table of Contents

## List of Figures

## List of Tables

Chapter 1

# Introduction

## 1.1 Introduction

Handling walk-in bill payments is a vital customer service function for utility service providers. However, traditional cashiering methods are time consuming and take away resources from customer services and revenue generating activities. Rapidly increasing personnel and back office costs are an ever-growing concern, and since a quarter of households worldwide do not have sufficient access to bank accounts, many customers can be forced to wait in long queues simply to pay their everyday bills. These challenges have led many executives working for utility companies and government service providers to look for ways to serve their customers more efficiently, without hampering staff and increasing operational costs.

## 1.2 Utility Bill Payment system in Bangladesh

As more and more utility services are provided to the mass people of the country, the more it is getting difficult for people to follow the traditional way of paying the utility bills standing in big queues. Alternatives should be adapted as the technology is available. To ease the payment of utility bills the electronic payment should be introduced to the majority of people. When the Electronic Utility Bill System is introduced people will be able to pay their daily utility bills through their home PCs, kiosks, mobile phones, cyber café etc. They can still have the option of paying the bills in the public offices. Banks will also play an important role as the transactions would take place using ATM machines, POS, credit/debit cards, Internet Banking etc. Some utility service providers and some banks are already carrying out these tasks. After kiosk operation various banks managed transaction still on papers. Inter branch utility bill also collected in a manual way.

My project is based on the above concept i.e. automation of kiosk transaction after kiosk operation and automation of bills collected in bank. The project has been developed keeping in-view the following aspects:

(i) Store transaction in database after kiosk operation.

(ii) Automatic calculate bill amount

(iii) Find out mismatch transaction

(iv) Find out bill, account, posting type wise report

(v) Reduce per bill wise IBCA transaction in bank

"Kiosk Transaction Management & Utility Bills Collection System " is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to bank. It has been designed to improve the quality and management of bank management in the areas of automated bill generation, managerial and financial reports. This System enables to develop organization and improve its effectiveness and quality of work.

## 1.3 Project Objective and Scope

### 1.3.1 Objective

Self-service payment channels and unattended kiosks have the advantages of speed, convenience, and availability. Among self-service kiosks, Automated Teller Machines (ATMs) have the highest presence and acceptance among consumers. They continue to be a strong viable alternative for banks as compared to branch locations and are now being increasingly used to provide a variety of services, other than dispensing cash. The single biggest benefit of an ATM is its potential to act as a self-contained, accessible and secure extended arm of financial institutions. It is necessary for the banks to keep track of its day-to-day kiosk transaction.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase transaction in bank. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

Thus keeping the working of the manual system as the basis of our project. I have developed an automated system, named as "Kiosk Transaction Management & Utility Bills Collection System".

The main objective of my project is to provide:

- Gain operational efficiency in the area of Kiosk Transaction Management & Utility Bills Collection System

- Minimize manual intervention

- Establish effective tool for management monitoring

- Ensure adequate level of internal control

- Simplify operation

- Save time and ensure savings

- Ensure security of data at every level of user-system interaction

- Provides robust & reliable storage and backup facilities

### 1.3.2 Scope

This System can be used in any bank for maintaining kiosk transaction and inter branch utility bill transaction for store transaction, calculate amount, find out mismatch transaction and all category wise statement.

## 1.4 Motivation

At present after kiosk operation various banks managed transaction still on papers. Inter branch utility bill also collected in a manual way. Calculate summery on records, separate bill type, find out mismatch transaction etc. managed in manual way. These are time consuming, and inefficient system leading to problems of inefficient utilization of resources and problems to bank user.

With these problems in mind and in order to manage information systematically, I decided to develop computerized Kiosk Transaction & Utility Bills Collection System which will make the best use of the increasing availability of information technology and help to bank in transaction management in an efficient, effective manner.

## 1.5 Contribution

In the present era of globalization and advanced technology efficient record keeping cannot be over emphasized. Imagine the scenario when the manual processes and manual modes of instruction get replaced with electronic systems. One of such replacement can be done in the area of Kiosk Transaction Management & Utility Bills Collection System within a bank. Developing Kiosk Transaction Management & Utility Bills Collection System software would benefit the banking transaction management who can have effortless access to the data securely and more easily.

Such software has features that allow the users to manage the kiosk and inter branch transaction and incorporate security features that prevent misuse of the stored information. This software allows the bank user:

- To store kiosk transaction in database
- To Setup kiosk physical amount transaction
- To Mark mismatch kiosk transaction
- To Collect utility bill to inter branch in bank
- To Search bill type wise transaction
- To Search posting date wise transaction

- ➢ To Auto calculate transaction amount
- ➢ To Calculate summery on record wise
- ➢ To prepare various types of reports

This system will be a cost-effective system which:

- ✓ Enhances cost control via increased efficiency. The data is assembled in a highly systematic manner, paving way for easy access to crucial information such as bill type wise, posting type wise transaction etc.
- ✓ Provides a number of advantages to a bank to manage transaction. This system will be cost-effective and scalable.

## 1.6 Organization of Report

Chapter 2 describes the Survey of existing model overview of Kiosk Transaction Management & Utility Bills Collection Systems. In this section, a brief introduction of current manual kiosk transaction management, drawbacks of current manual system, establishing the Need of new system. Also discuss on my proposed system and its features briefly.

In the Chapter 3 proposed model and implementation plan has been discussed. In proposed model I have followed the software engineering process like system requirements, Entity relationship diagram, Data flow diagram, data specification, database model diagram etc. The paper has been discussed with the Computerized kiosk transaction management and utility bill collection system requirements. Step by step implementation plan aligning with the model has been discussed. An application has been developed and how this can be aligned with the model has been discussed in this chapter.

In the Chapter 4 the user manual in included, here software functional activities with screen shot describes. How the system is working, how to operate System, and System snapshot are mentioned.

In the Chapter 5 describes conclusion and feature works of the project and small discussions were there. Brief discussion about the advantages of the model and how this has been developed has been discussed. A very discussion about the feature works and how it can be done has been discussed in this chapter.

# Survey of Existing Models

## 2.1 Introduction

I surveyed number of branch of Standard bank Ltd. This branch locations in Dhaka city. For utility bill collection such as wasa,DPDC, palli bidyut somity etc. this branch use per transaction wise ibca (inter branch credit advice) transaction. Some branch manage kiosk transaction on papers. As part of contentious improvement for efficient management of day to day operation, I realized that it needs to automate different operational processes.

## 2.2 Definition of Problem

The manual handling of the record is time consuming and highly prone to error. The purpose of this project is to automate, the process management of transaction like as summery on record, separate bill type wise transaction, find out mismatch transaction etc.

I have tried my best to make the System as simple as possible using Structured & Modular technique & Menu oriented interface. I have tried to design the software in such a way that user may not have any difficulty in using this software & further expansion is possible without much effort. Even though I cannot claim that this work to be entirely exhaustive, the main purpose of my project is to perform banks activity in computerized way rather than manually which is time consuming.

I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error.

## 2.3 Drawbacks of Current Manual System

1. The current manual system has a lot of paper work.

2. Inconsistency in data entry, miscopying information.

3. With the increase in records, it will become a massive job to maintain.

4. Time consuming and costly to produce reports.

5. Large ongoing staff training cost.

6. Lack of security for the records, anyone can easily disarrange the records.

7. Duplication of data entry.

8. System is dependent on good individuals.

9. Limited Accessibility.

## 2.4 Establishing the Need of New System

1. Problem of Reliability: Current manual system is not reliable. It seems to vary in quality from one month to the, next. Sometimes it gives good output, but sometimes the output is worst.

2. Problem of Accuracy: There are too many mistakes in reports.

3. Problem of Timeliness: In the current system the reports and output produced is mostly late and in most of the cases it is useless because it is not on time.

4. Problem of Validity: The output and reports mostly contains misleading information.

5. Problem of Economy: The current system is very costly. We have to spend lots of money to keep the system up and going, but still not get the desired results.

6. Problem of Capacity: The current system is suffering from problem of capacity also. The staff for organization is very less and the workload is too much. Few peoples cannot handle all the work.

## 2.5 My Proposed System

- **Marked mismatch transaction**
  If there have any difference occurrence between physical amount and deposit amount, those transactions are auto marked as mismatch transaction.

- **Reduce inter branch credit advice**
  In branch per bill wise ibca transaction are generated. These are time consuming, and inefficient system and problems to bank user. User can post all bill transaction in this system and at the day end user can perform bill type wise single ibca transaction.

- **Summery calculation on record wise**

  User can calculate record wise summery in this system such as total Bill amount, VAT amount, revenue amount, net bill amount, physical amount etc.

- **Searching different type wise transaction**
  User can search posting type wise, bill type wise transaction.

- **Separate matched and mismatched transaction**
  By this system user can separate matched and mismatched transaction. User can generate only matched transaction or mismatched transaction or both.

- **Generate different type wise report**

By this system user can generate different type wise report. User can generate report such as account type wise, bill type wise, account number wise, payment type wise, bill mode wise, posting date wise etc.

# Proposed Model and Implementation Plan

## 3.1 Introduction

Kiosk Transaction Management & Utility Bills Collection System provides the benefits of streamlined operations, enhanced administration & control, help to maintain kiosk transactions, mismatch amount, calculate bill amount, vat amount and generate report bill type wise or all together.

This system is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to banks. More importantly it is backed by reliable and dependable support.

This system is designed for banks, to maintain a huge transaction. It is an integrated end-to-end system that provides relevant information about transaction and critical financial accounting, in a seamless flow.

## 3.2 Literature Review

In a bank where many financial transaction related to utility bills related are generated and need to maintain a proper way.

Various kiosks and utility bills transaction are maintained in bank:

- Encrypted text file generated by kiosk machine are decrypted by the software in xls and text file format.
- After the calculate bill amount, revenue amount, vat amount in xls file that are matched with physical amount.
- If any mismatch found in deposited money input by the customer in kiosk machine and physical amount then that transaction is not completed that day.
- Transaction is done after the mismatch problem solve by the communication to customer.
- Many customers to pay their everyday utility bills come in bank. In bank utility bill transaction are done by IBCA transaction.
- Whenever a transaction between inter-branch or Branch-Head office takes place, it is advised to the relevant branch or Head Office through inter branch advices namely

IBDA( Inter-Branch Debit Advice)or IBCA ( Inter-Branch Credit Advice) duly recorded in IBCA/IBDA issued and received register.

These are the various jobs that need to be done in a bank. All these works are done on papers. All this work is done manually by the lot of operational bank staff. For manually work lot of papers are needed to be handled and taken care of.

**"Kiosk Transaction Management & Utility Bills Collection System"** has been designed to computerize the following functions that are performed by the system:

**I. Upload kiosk transaction in database**

**II. Setup kiosk physical amount transaction**

**III. Mark mismatches kiosk transaction**

**IV. Collect utility bill in bank without kiosk**

**V. Search bill type wise transaction**

**VI. Search posting date wise transaction**

**VII. Auto calculate transaction amount**

a. Bill amount

b. VAT amount

c. Revenue amount

d. Net bill amount

e. Physical amount

**VIII. Record wise summery calculation**

**VII. Reporting**

a. Kiosk transaction report

b. Manual transaction report

c. Bill type wise report

d. Account type wise report

e. Posting date wise report

f. All kiosk and manual transaction report

g. Only matched transaction report

h. Only mismatched transaction report

i. Payment type wise report

## 3.3 Research Methodology

The project "Kiosk Transaction Management & Utility Bills Collection System" is based on the database, networking techniques. As there are many areas where we keep the records in database for which we are using ORACLE 10g database which is one of the best and the easiest way to keep information. This project uses Oracle forms developer (10g) as the front-end and has connectivity with Oracle database. The operation part is web based application.

**Minimum Hardware requirements: (Computer)**

| | | |
|---|---|---|
| Processor | : | Intel Core i3, 3.5 GHz or above |
| RAM | : | 2 GB RAM or above |
| Cache Memory | : | 3 MB L3 |
| Hard Disk | : | 500 GB (minimum 3 GB free space) |
| Printer | : | LaserJet Printer |

**SOFTWARE**

| | | |
|---|---|---|
| Operating System | : | Windows XP, Windows 7 or 8 |
| Font-End Tool | : | Oracle Forms developer (10g) |
| Back-End | : | Oracle 10g |

**FRONT END**

I have used Oracle Forms developer (10g) in the Font-End Tool. It is web based, event driven, platform independent.

**BACK END**

I have used Oracle 10g. Oracle 10g provides efficient/effective solution for major database tech.

- Large database and space management.
- Many concurrent database users.
- High transaction processing requirement
- High Availability
- Industry accepted standards
- Manageable security
- Portability

## 3.4 System Analysis

### 3.4.1 Principals of the System Analysis

- ➢ Understand the problem before I begin to create the analysis model
- ➢ Develop prototypes that enable a user to understand how human machine interaction will occur
- ➢ Record the origin of and the reason for every requirement
- ➢ Use multiple views of requirements like building data, function and behavioral models
- ➢ Work to eliminate ambiguity

System Analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system. One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

• What type of information is required?

• What are the constraints on the investigation?

• What are the potential problems that may make the task more difficult?

Keeping the above questions in mind and considering the survey conducted to determine the need of the system; the total system was designed and can be described as under:

The three major parts of the system are:

**1. Providing Information**

The system is effectively used to provide large variety of information to the interested customer. The major purpose of the site is to easily provide access to records of various Job seekers &users of matrimonial such as resume & profile of boys and girls those who want to search a life partner with quick update to latest modifications in the records. This thing is not at all possible in printed material, which are updated only once a few weeks. It also gives information about the general usage of the system for first time visitors. The system itself works as a information provider for company & life partner seekers.

**2. Preliminary Investigation**

System development, a process consisting of two major steps of system analysis and design, start when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The system development life cycle is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system. The system development life cycle consists of the following activities:

- ➢ Preliminary investigation
- ➢ Determination of system requirements
- ➢ Design of system
- ➢ Development of software
- ➢ System testing
- ➢ Implementation, evaluation, and maintenance

A request to take assistance from information system can be made for many reasons, but in each case someone in the organization initiates the request is made, the first system activity the preliminary investigation begins. This activity has three parts:

- ➢ Request clarification
- ➢ Feasibility study
- ➢ Request approval

**3. Request clarification**

Many requests from employees and users in the organizations are not clearly defined, therefore it becomes necessary that project request must be examined and clarified properly before considering systems investigation.

### 3.4.1.1Feasibility Study

The feasibility study proposes one or more conceptual solution to the problem set of the project. In fact, it is an evaluation of whether it is worthwhile to proceed with project or not.

1. Evaluation of feasibility of such solutions. Such evaluation often indicates shortcomings in the initial goals. This step is repeated as the goals are adjusted and the alternative solutions are evaluated.

Feasibility analysis usually considers a number of project alternatives, one that is chosen as the most satisfactory solution. These alternatives also need to be evaluated in a broad way without committing too many resources. Various steps involved in feasibility analysis are:

2. To propose a set of solution that can realize the project goal. These solutions are usually descriptions of what the new system should look like.

Four primary areas of interest in feasibility study are:

**1. Economic Feasibility:**

An evaluation of development cost weighed against the ultimate income of benefit derived from the development system of product. In economic feasibility, cost benefit analysis is done in which expected cost and benefits are evaluated.

### 3.4.1.2 Cost and Benefit Analysis

Developing an IT application is an investment. Since after developing that application it provides the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks, because in some cases an estimate can be wrong. And the project might not actually turn out to be beneficial.

Cost benefit analysis helps to give management a picture of the cost, benefits and risks. It usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and compares them with the expected costs.

In performing cost and benefit analysis it is important to identify cost and benefits factors. Cost and benefits can be categorized into the following categories:

1. Development Costs – Development costs is the costs that are incurred during the development of the system. It is one time investment.

2. Operating Costs – Operating Costs are the expenses required for the day to day running of the system. Examples of Operating Costs are Wages, Supplies and Overheads.

3. Hardware/Software Costs – It includes the cost of purchasing or leasing of computers and it's peripherals. Software costs involve required S/W costs.

4. Personnel Costs – It is the money spent on the people involved in the development of the system.

5. Facility Costs – Expenses that are incurred during the preparation of the physical site where the system will be operational. These can be wiring, flooring, acoustics, lightning, and air-conditioning.

6. Supply Costs – These are variable costs that are very proportionately with the amount of use of paper, ribbons, disks, and the like

**Benefit:**

We can define benefits as:

Profit or Benefit = Income – Costs

Benefits can be accrued by:

- Increasing income, or
- Decreasing costs, or
- Both

### 3.4.1.3 Technical Feasibility

Technical Feasibility includes existing and new H/W and S/W requirements that are required to operate the project using oracle forms. The basic S/W requirement is oracle forms in which the front end of the Kiosk Transaction Management & Utility Bills Collection System project has been done. The basic entry forms are developed in oracle forms and the data is stored in the oracle 10g database.

### 3.4.1.4 Operational Feasibility

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance from users that will affect the possible application benefits? The essential questions that help in testing the technical feasibility of a system are following:

Does management support the project?

- Are the users not happy with current business practices? Will it reduce the time considerably? If yes, then they will welcome the change and the new system.
- Have the users involved in the planning and development of the project? Early involvement reduced the probability of resistance towards the new system.
- Will the proposed system really benefit the organization? Does the overall response increase? Will accessibility of information be lost? Will the system affect the customers in considerable way?
-

### 3.4.2 System Life Cycle

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means software development life cycle.

Following are the different phases of software development cycle:

- System study
- Feasibility study
- System analysis
- System design
- Coding
- Testing
- Implementation
- Maintenance

### 3.4.3 Identification of Need

The following steps that give the detailed information of the need of proposed system are:

**Performance:** During past several decades, the records are supposed to be manually handled for all activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the Kiosk Transaction Management & Utility Bills Collection System, the computerized system is to be undertaken. The computerized project is fully computerized and user friendly even that any of the user can see the report and status of it as well as customer transaction history.

**Efficiency:** The basic need of this system is efficiency. The system should be efficient so that whenever a user upload transaction in the database the system is updated automatically. These records will be useful for other users instantly.

**Control:** The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator to set up branch, user, division etc and the other members have the rights to perform transaction.

**Security:** Security is the main criteria for the proposed system. Since illegal access may corrupt the database. So security has to be given in this project.

### 3.4.4 Software Requirement Specification (SRS)

What are the benefits of a SRS?

The IEEE 830 standard defines the benefits of a good SRS:

• Establish the basis for agreement between the customers and the suppliers on what the software product is to do. The complete description of the functions to be performed by the software specified in the SRS will assist the potential users to determine if the software specified meets their needs or how the software must be modified to meet their needs. [NOTE: We use it as the basis of our contract with our clients all the time].

• Reduce the development effort. The preparation of the SRS forces the various concerned groups in the customer's organization to consider rigorously all of the requirements before design begins and reduces later redesign, recoding, and retesting. Careful review of the requirements in the SRS can reveal omissions, misunderstandings, and inconsistencies early in the development cycle when these problems are easier to correct.

• Provide a basis for estimating costs and schedules. The description of the product to be developed as given in the SRS is a realistic basis for estimating project costs and can be used to obtain approval for bids or price estimates. [NOTE: Again, we use the SRS as the basis for our fixed price estimates]

• Provide a baseline for validation and verification. Organizations can develop their validation and Verification plans much more productively from a good SRS. As a part of the development contract, the SRS provides a baseline against which compliance can be measured. [NOTE: We use the SRS to create the Test Plan].

• Facilitate transfer. The SRS makes it easier to transfer the software product to new users or new machines. Customers thus find it easier to transfer the software to other parts of their organization, and suppliers find it easier to transfer it to new customers.

• Serve as a basis for enhancement. Because the SRS discusses the product but not the project that developed it, the SRS serves as a basis for later enhancement of the finished product. The SRS may need to be altered, but it does provide a foundation for continued production evaluation.

What should the SRS address?

The basic issues that the SRS writer(s) shall address are the following:

• Functionality

What is the software supposed to do?

My System aims at automating the manual system being used in the bank for maintaining the kiosk and utility bill transaction.

• External interfaces.

How does the software interact with people, the system's hardware, other hardware, and other software?

My system uses various forms and reports through which users can interact with the system. As we can have a centralized database so our system interact it with database software through network and as network comes in, it uses network peripheral likes switches etc.

### 3.4.5 Software Engineering Paradigm

Software engineering paradigm can be defined as a development strategy that encompasses the process, methods, tools layers and generic phases such as definition phase, development phase and support phase to solve real lives problems in an industry, research institute etc. used by software engineers or team of engineers.

The most useful software engineering paradigm that I incorporate in developing my project is the "Spiral Model". This process model is suitable and useful for my project because the following reasons:

•The all requirements of the project are not well understood by me at the beginning.

•Relatively small but too complex project.

•This project is a complete new research. I did not previously implement the logics and all necessary factors governing the project.

•The target of this process model is that a complete system will be developed after the number of refinement.

This approach to software development begins at the system level and progresses towards through analysis, design, coding, testing and support. The details discussion of these steps is beyond the consideration of the project but all these steps have been applied throughout the SDLC of my project.

## 3.5 System Design

The design document that we will develop during this phase is the blueprint of the software. It describes how the solution to the customer problem is to be built. Since solution to complex problems isn't usually found in the first try, iterations are most likely required. This is true for software design as well. For this reason, any design strategy, design method, or design language must be flexible and must easily accommodate changes due to iterations in the design. Any technique or design needs to support and guide the partitioning process in such a way that the resulting sub-problems are as independent as possible from each other and can be combined easily for the solution to the overall problem. Sub-problem independence and easy combination of their solutions reduces the complexity of the problem. This is the objective of the partitioning process. Partitioning or decomposition during design involves three types of decisions: -

Define the boundaries along which to break;

Determine into how money pieces to break; and

Identify the proper level of detail when design should stop and implementation should start. Basic design principles that enable the software engineer to navigate the design process suggest a set of principles for software design, which have been adapted and extended in the following list:

Free from the suffer from "tunnel vision." A good designer should consider alternative approaches, judging each based on the requirements of the problem, the resources available to do the job.

The design should be traceable to the analysis model. Because a single element of the design model often traces to multiple requirements, it is necessary to have a means for tracking how requirements have been satisfied by the design model.

The design should not repeat the same thing. Systems are constructed using a set of design patterns, many of which have likely been encountered before. These patterns should always be chosen as an alternative to reinvention. Time is short and resources are limited! Design time should be invested in representing truly new ideas and integrating those patterns that already exist.

The design should "minimize the intellectual distance" between the software and the problem as it exists in the real world. That is, the structure of the software design should (whenever possible) mimic the structure of the problem domain. The design should exhibit uniformity and integration. A design is uniform if it appears that one person developed the entire thing. Rules of style and format should be defined for a design team before design work begins. A design is integrated if care is taken in defining interfaces between design components.

The design activity begins when the requirements document for the software to be developed is available. This may be the SRS for the complete system, as is the case if the waterfall model is being followed or the requirements for the next "iteration" if the iterative enhancement is being followed or the requirements for the prototype if the prototyping is being followed. While the requirements specification activity is entirely in the problem domain, design is the first step in moving from the problem domain toward the solution domain. Design is essentially the bridge between requirements specification and the final solution for satisfying the requirements.

The design of a system is essentially a blueprint or a plan for a solution for the system. We consider a system to be a set of components with clearly defined behavior that interacts with each other in a fixed defined manner to produce some behavior or services for its environment. A component of a system can be considered a system, with its own components. In a software system, a component is a software module. The design process for software systems, often, has two levels. At the first level, the focus is on deciding which modules are needed for the system, the specifications of these modules, and how the modules should be interconnected. This is what is called the system design or top-level design. In the second level, the internal design of the modules, or how the specifications of the module can be satisfied, is decided. This design level is often called detailed design or logic design. Detailed design essentially expands the system design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

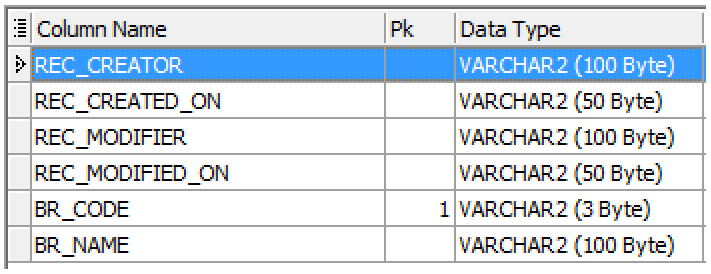
Because the detailed design is an extension of system design, the system design controls the major structural characteristics of the system. The system design has a major impact on the testability and modifiability of a system, and it impacts its efficiency. Much of the design effort for designing software is spent creating the system design.

The input to the design phase is the specifications for the system to be designed. Hence, reasonable entry criteria can be that the specifications are stable and have been approved, hoping that the approval mechanism will ensure that the specifications are complete, consistent, unambiguous, etc. The output of the top-level design phase is the architectural design or the system design for the software system to be built. This can be produced with or without using a design methodology. Reasonable exit criteria for the phase could be that the design has been verified against the input specifications and has been evaluated and approved for quality.

A design can be object-oriented or function-oriented. In function-oriented design, the design consists of module definitions, with each module supporting a functional abstraction. In object-oriented design, the modules in the design represent data abstraction (these abstractions are discussed in more detail later). In the function-oriented methods for design and describe one particular methodology the structured design methodology in some detail. In a function- oriented design approach, a system is viewed as a transformation function, transforming the inputs to the desired outputs. The purpose of the design phase is to specify the components for this transformation function, so that each component is also a transformation function. Hence, the basic output of the system design phase, when a function oriented design approach is being followed, is the definition of all the major data structures in the system, all the major modules of the system, and how the modules interact with each other. Once the designer is satisfied with the design he has produced, the design is to be precisely specified in the form of a document. To specify the design, specification languages are used. Producing the design specification is the ultimate objective of the design phase. The purpose of this design document is quite different from that of the design notation. Whereas a design represented using the design notation is largely to be used by the designer, a design specification has to be so precise and complete that it can be used as a basis of further development by other programmers. Generally, design specification uses textual structures, with design notation helping in understanding.

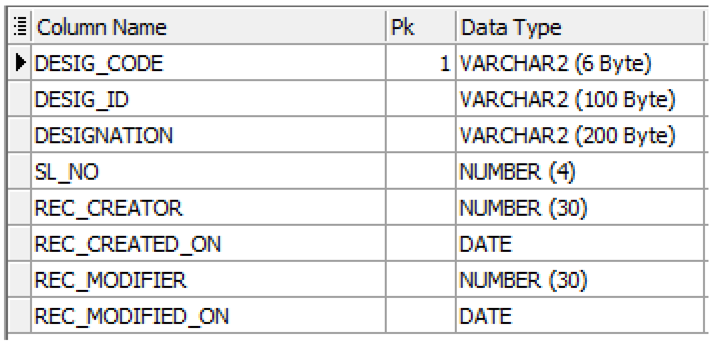
### 3.5.1 Data Modeling

Table 3.5.1 Branch



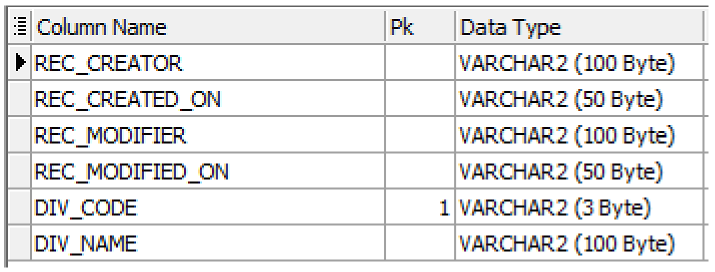
| Column Name | Pk | Data Type |
|---|---|---|
| REC_CREATOR | | VARCHAR2 (100 Byte) |
| REC_CREATED_ON | | VARCHAR2 (50 Byte) |
| REC_MODIFIER | | VARCHAR2 (100 Byte) |
| REC_MODIFIED_ON | | VARCHAR2 (50 Byte) |
| BR_CODE | 1 | VARCHAR2 (3 Byte) |
| BR_NAME | | VARCHAR2 (100 Byte) |

Table 3.5.2 Designation



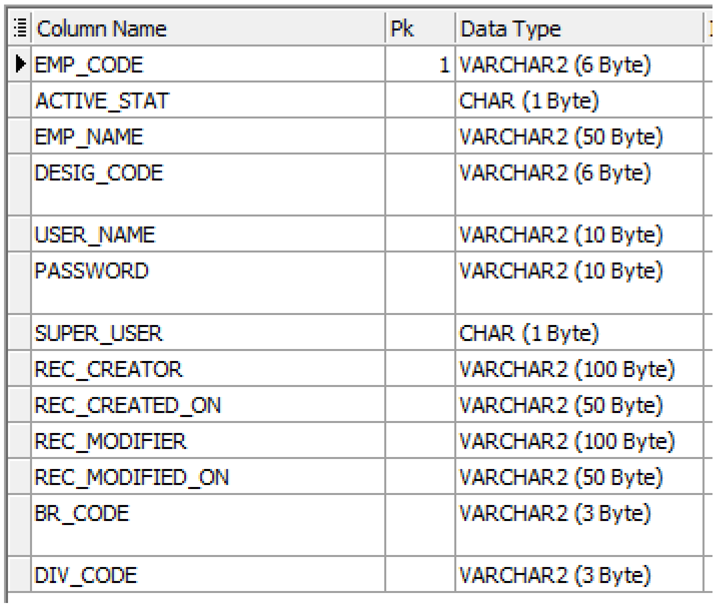
| Column Name | Pk | Data Type |
|---|---|---|
| DESIG_CODE | 1 | VARCHAR2 (6 Byte) |
| DESIG_ID | | VARCHAR2 (100 Byte) |
| DESIGNATION | | VARCHAR2 (200 Byte) |
| SL_NO | | NUMBER (4) |
| REC_CREATOR | | NUMBER (30) |
| REC_CREATED_ON | | DATE |
| REC_MODIFIER | | NUMBER (30) |
| REC_MODIFIED_ON | | DATE |

Table 3.5.3 Division



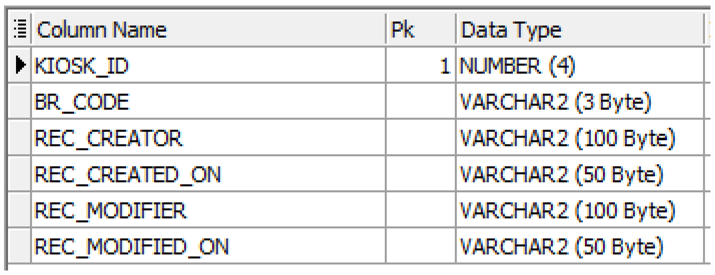
| Column Name | Pk | Data Type |
|---|---|---|
| REC_CREATOR | | VARCHAR2 (100 Byte) |
| REC_CREATED_ON | | VARCHAR2 (50 Byte) |
| REC_MODIFIER | | VARCHAR2 (100 Byte) |
| REC_MODIFIED_ON | | VARCHAR2 (50 Byte) |
| DIV_CODE | 1 | VARCHAR2 (3 Byte) |
| DIV_NAME | | VARCHAR2 (100 Byte) |

Table 3.5.4 Employees



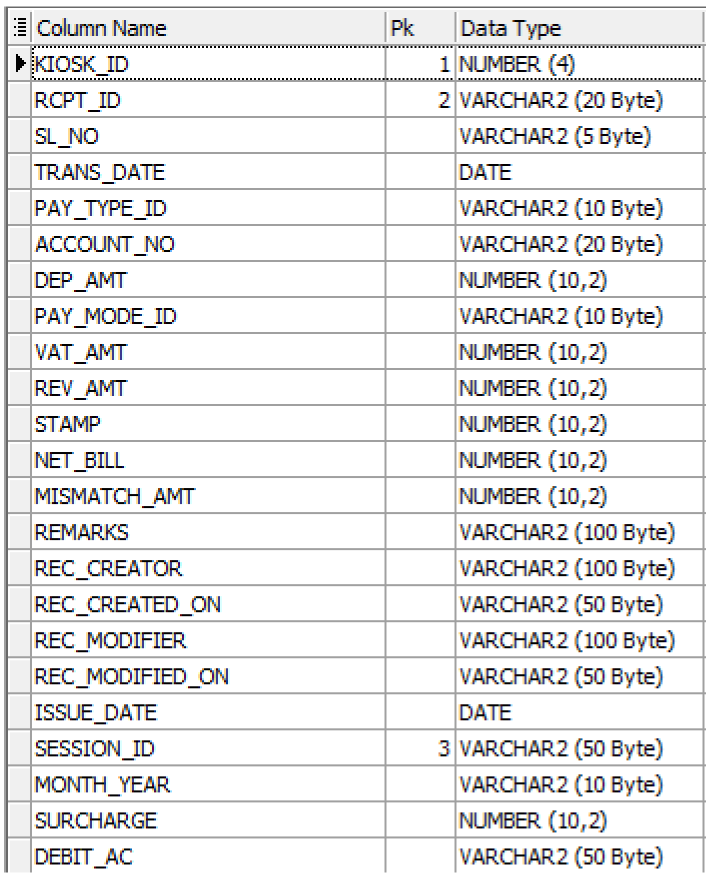
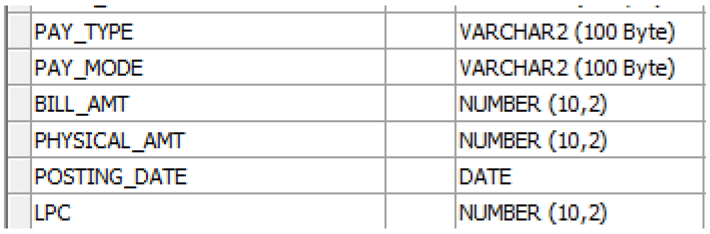
| Column Name | Pk | Data Type |
|---|---|---|
| EMP_CODE | 1 | VARCHAR2 (6 Byte) |
| ACTIVE_STAT | | CHAR (1 Byte) |
| EMP_NAME | | VARCHAR2 (50 Byte) |
| DESIG_CODE | | VARCHAR2 (6 Byte) |
| USER_NAME | | VARCHAR2 (10 Byte) |
| PASSWORD | | VARCHAR2 (10 Byte) |
| SUPER_USER | | CHAR (1 Byte) |
| REC_CREATOR | | VARCHAR2 (100 Byte) |
| REC_CREATED_ON | | VARCHAR2 (50 Byte) |
| REC_MODIFIER | | VARCHAR2 (100 Byte) |
| REC_MODIFIED_ON | | VARCHAR2 (50 Byte) |
| BR_CODE | | VARCHAR2 (3 Byte) |
| DIV_CODE | | VARCHAR2 (3 Byte) |

Table 3.5.5 Kiosk_Info



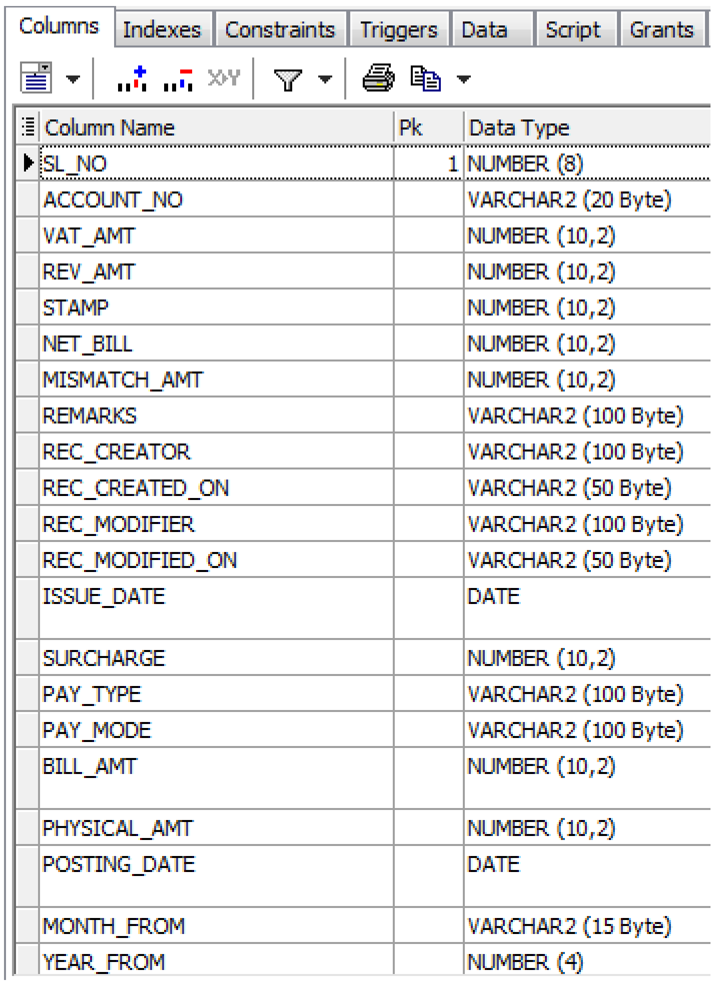
| Column Name | Pk | Data Type |
|---|---|---|
| KIOSK_ID | 1 | NUMBER (4) |
| BR_CODE | | VARCHAR2 (3 Byte) |
| REC_CREATOR | | VARCHAR2 (100 Byte) |
| REC_CREATED_ON | | VARCHAR2 (50 Byte) |
| REC_MODIFIER | | VARCHAR2 (100 Byte) |
| REC_MODIFIED_ON | | VARCHAR2 (50 Byte) |

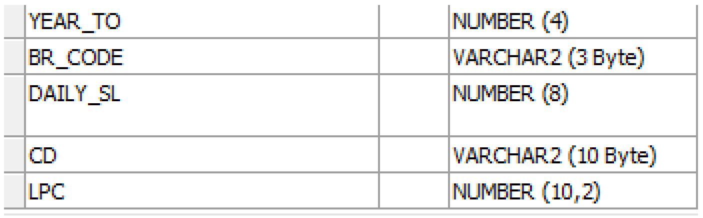
Table 3.5.6 Trans_Info

| Column Name | Pk | Data Type |
|---|---|---|
| KIOSK_ID | 1 | NUMBER (4) |
| RCPT_ID | 2 | VARCHAR2 (20 Byte) |
| SL_NO | | VARCHAR2 (5 Byte) |
| TRANS_DATE | | DATE |
| PAY_TYPE_ID | | VARCHAR2 (10 Byte) |
| ACCOUNT_NO | | VARCHAR2 (20 Byte) |
| DEP_AMT | | NUMBER (10,2) |
| PAY_MODE_ID | | VARCHAR2 (10 Byte) |
| VAT_AMT | | NUMBER (10,2) |
| REV_AMT | | NUMBER (10,2) |
| STAMP | | NUMBER (10,2) |
| NET_BILL | | NUMBER (10,2) |
| MISMATCH_AMT | | NUMBER (10,2) |
| REMARKS | | VARCHAR2 (100 Byte) |
| REC_CREATOR | | VARCHAR2 (100 Byte) |
| REC_CREATED_ON | | VARCHAR2 (50 Byte) |
| REC_MODIFIER | | VARCHAR2 (100 Byte) |
| REC_MODIFIED_ON | | VARCHAR2 (50 Byte) |
| ISSUE_DATE | | DATE |
| SESSION_ID | 3 | VARCHAR2 (50 Byte) |
| MONTH_YEAR | | VARCHAR2 (10 Byte) |
| SURCHARGE | | NUMBER (10,2) |
| DEBIT_AC | | VARCHAR2 (50 Byte) |
| PAY_TYPE | | VARCHAR2 (100 Byte) |
| PAY_MODE | | VARCHAR2 (100 Byte) |
| BILL_AMT | | NUMBER (10,2) |
| PHYSICAL_AMT | | NUMBER (10,2) |
| POSTING_DATE | | DATE |
| LPC | | NUMBER (10,2) |

Table 3.5.7 Trans_Manual





| Column Name | Pk | Data Type |
|---|---|---|
| SL_NO | 1 | NUMBER (8) |
| ACCOUNT_NO | | VARCHAR2 (20 Byte) |
| VAT_AMT | | NUMBER (10,2) |
| REV_AMT | | NUMBER (10,2) |
| STAMP | | NUMBER (10,2) |
| NET_BILL | | NUMBER (10,2) |
| MISMATCH_AMT | | NUMBER (10,2) |
| REMARKS | | VARCHAR2 (100 Byte) |
| REC_CREATOR | | VARCHAR2 (100 Byte) |
| REC_CREATED_ON | | VARCHAR2 (50 Byte) |
| REC_MODIFIER | | VARCHAR2 (100 Byte) |
| REC_MODIFIED_ON | | VARCHAR2 (50 Byte) |
| ISSUE_DATE | | DATE |
| SURCHARGE | | NUMBER (10,2) |
| PAY_TYPE | | VARCHAR2 (100 Byte) |
| PAY_MODE | | VARCHAR2 (100 Byte) |
| BILL_AMT | | NUMBER (10,2) |
| PHYSICAL_AMT | | NUMBER (10,2) |
| POSTING_DATE | | DATE |
| MONTH_FROM | | VARCHAR2 (15 Byte) |
| YEAR_FROM | | NUMBER (4) |
| YEAR_TO | | NUMBER (4) |
| BR_CODE | | VARCHAR2 (3 Byte) |
| DAILY_SL | | NUMBER (8) |
| CD | | VARCHAR2 (10 Byte) |
| LPC | | NUMBER (10,2) |

## 3.5.2 Scheduling

Scheduling of a software project does not differ greatly from scheduling of any multi- task engineering effort. Therefore, generalized project scheduling tools and techniques can be applied with little modification to software projects.

Program evaluation and review technique (PERT) and critical path method (CPM) are two project scheduling methods that can be applied to software development. Both techniques are driven by information already developed in project planning activities.

<u>**Estimates of Effort**</u>

- A decomposition of the product function.
- The selection of the appropriate process model and task set.
- Decomposition of tasks.

Interdependencies among tasks may be defined using a task network. Tasks, sometimes called the project Work Breakdown Structure (WBS) are defined for the product as a whole or for individual functions.

Both PERT and CPM provide quantitative tools that allow the software planner to (1) determine the critical path-the chain of tasks that determines the duration of the project; (2) establish "most likely" time estimates for individual tasks by applying statistical models; and (3) calculate "boundary times" that define a time window" for a particular task.

Boundary time calculations can be very useful in software project scheduling. Slippage in the design of one function, for example, can retard further development of other functions. It describes important boundary times that may be discerned from a PERT or CPM network: (I) the earliest time that a task can begin when preceding tasks are completed in the shortest possible time, (2) the latest time for task initiation before the minimum project completion time is delayed, (3) the earliest finish-the sum of the earliest start and the task duration, (4) the latest finish- the latest start time added to task duration, and (5) the total float-the amount of surplus time or leeway allowed in scheduling tasks so that the network critical path maintained on schedule. Boundary time calculations lead to a determination of critical path and provide the manager with a quantitative method for evaluating progress as tasks are completed.

Both PERT and CPM have been implemented in a wide variety of automated tools that are available for the personal computer. Such tools are easy to use and take the scheduling methods described previously available to every software project manager.

## 3.6 Code Efficiency

Reviewing of Code efficiency for a module is carried out after the module is successfully compiled and all the syntax errors eliminated. Code efficiency review is extremely cost-effective strategies for reduction in coding errors in order to produce high quality code. Normally, two types of efficiency are carried out on the code of a module - code optimization and code inspection. The procedure and final objective of these two efficiency techniques are very different as discussed below.

## 3.7 Optimization of Code

Code optimization is an informal code analysis technique. In this technique, after a module has been coded, it is successfully compiled and all syntax errors are eliminated. Some members of the development team are given the code a few days before the optimization meeting to read and understand the code. Each member selects some test cases and simulates execution of the code by hand (i.e. trace execution through each statement and function execution). The main objectives of the optimization are to discover the algorithmic and logical errors in the code. The members note down their findings to discuss these in optimization meeting where the coder of the module is also present.

Even though a code optimization is an informal analysis technique, several guidelines have evolved over the years for making this naïve technique more effective and useful. Of course, these guidelines are based on personal experience, common sense, and several subjective factors. Therefore are based on personal experience, common sense, and several subjective factors. Therefore, guidelines should be considered as examples rather than as rules to be applied dogmatically. Some of these guidelines are the following:

The team performing the code optimization should not be either too big or too small. Ideally, it should consist of three to seven members.

## 3.8 Testing

One of the purposes of the testing is to validate and verify the system. Verification means checking the system to ensure that it is doing what the function is supposed to do and Validation means checking to ensure that system is doing what the user wants it to do.

### 3.8.1 Testing Phase

No program or system design is perfect; communication between the user and the designer is not always complete or clear, and time is usually short. The result is errors and more errors. Theoretically, a newly designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the user's requirements. This is the best chance to detect and correct errors before the system is implemented. The purpose of system testing is to consider all the likely variations to which it will be subjected and then push the system to its limits. If we implement the system without proper testing then it might cause the problems.

1. Communication between the user and the designer.

2. The programmer's ability to generate a code that reflects exactly the system specification.

3. The time frame for the design.

Theoretically, a new designed system should have all the pieces in working order, but in reality, each piece works independently. Now is the time to put all the pieces into one system and test it to determine whether it meets the requirements of the user. The process of system testing and the steps taken to validate and prepare a system for final implementation are:

### 3.8.2 Level of Testing

The different types of testing are as follows:

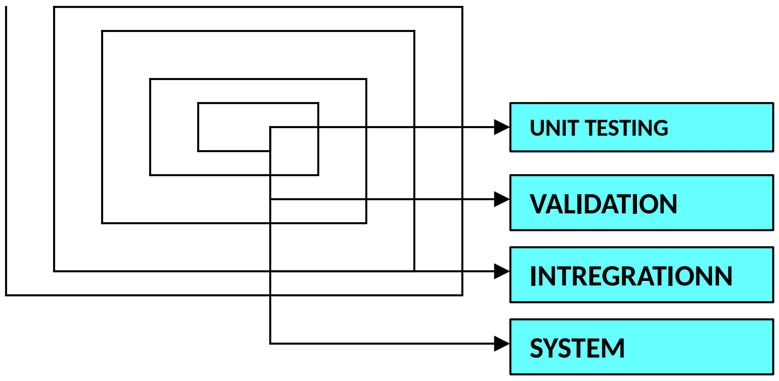


Figure: 3.1 Level of Testing

1<u>. Unit Testing:</u>

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

2<u>. Integration Testing:</u>

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

3. System Testing:

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

System testing consists of the following steps:

1. Program(s) testing.

2. String testing.

3. System testing.

4. System documentation.

5. User acceptance testing.

4. Field Testing:

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and the real world are checked. This type of testing is very rarely used. So far our project is concerned, we haven't tested our project using the field testing.

Acceptance Testing:

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfills the goals or not.

Why System Testing?

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing results in two types of problems:

1. The time lag between the cause and the appearance of the problem.

2. The effect of system errors on the files and records within the system.

Another reason for system testing is its utility as a user-oriented vehicle before implementation.

Activity Network for System Testing

The test plan entails the following activities:

1. Prepare test plan.

2. Specify conditions for user acceptance testing.

3. Prepare test data for program testing.

4. Prepare test data for transaction path testing.

5. Plan user training.

6. Compile/assemble programs.

7. Prepare job performance aids.

8. Prepare operational documents.

Prepare Test

A workable test plan must be prepared in accordance with established design specifications. It includes the following items:

• Outputs expected from the system.

• Criteria for evaluating outputs.

• A volume of test data.

• Procedure for using test data.

• Personnel and training requirements.

Specify Conditions for User Acceptance Testing

Planning for user acceptance testing calls for the analyst and the user to agree on conditions for the test.

Prepare Test Data for Program Testing

As each program is coded, test data are prepared and documented to ensure that all aspects of the program are properly tested.

Prepare Test Data for Transaction Path Testing

This activity develops the data required for testing every condition and transactions to be introduced into the system. The path of each transaction from origin to destination is carefully tested reliable results.

Plan User Training

User training is designed to prepare the user for testing and converting the system. User involvement and training take place parallel with programming for three reasons:

- The system group has time available to spend on training while the programs are being written.
- Initiating a user-training program gives the systems group a clearer image of the user's interest in the new system.
- A trained user participates more effectively in system testing.

The training plan is followed by preparation of the user training manual and other text materials.

Compile / Assemble Programs

All programs have to be compiled / assembled for testing.

Prepare Job Performance Aids

In this activity the materials to be used by personnel to run the system are specified and scheduled. This includes a display of materials.

Prepare Operational Documents

During the test plan stage, all operational documents are finalized including copies of the operational formats required by the candidate system.

Systems testing

The computer department to ensure that the system functions as specified does this testing. This testing is important to ensure that a working system is handed over to the user for acceptance testing.

Acceptance testing

The user to ensure that the system functions, as the user actually wanted performs this testing. With prototyping techniques, this stage becomes very much a formality to check the accuracy and completeness of processing. The screen layouts and output should already have been tested during the prototyping phase.

An error in the program code can remain undetected indefinitely. To prevent this from happening the code was tested at various levels. To successfully test a system, each condition, and combinations of conditions had to be tested. Each program was tested and linked to other programs. This unit of program is tested and linked to other units and so on until the complete system has been tested.

The purpose of testing is to ensure that each program is fully tested. To do so a test plan had to be created. The test plan consists of a number of test runs such as the valid paths through the code, and the exception and error handling paths. For each test run there is a list of conditions tested, the test data used and the result expected. The test plan was then reviewed to check that each path through the code is tested correctly. It is the responsibility of the programmer to collect the data that will produce the required test condition.

### 3.8.3 Verification and Validation (V&V)

The objectives of verification, validity activities are to assess and improve the quality of the work products generated during development and modification of the software. Quality depends upon the various attributes like correctness, completeness, consistency, reliability, usefulness, usability, efficiency and conformance to standards.

The terms verification and validation are used synonymously.  These are defined as under: -

Verification: "Are we building the product right?"

Validation: "Are we building the right product?"

Verification activities include proving, testing, and reviews. Validation is the process of evaluating software at the end of the software development to ensure compliance with the software requirements. Testing is a common method of validation. Clearly, for high reliability we need to perform both activities. Together, they are often called V&V activities.

The major V&V activities for software development are inspection, reviews, and testing (both static and dynamic). The V&V plan identifies the different V&V tasks for the different phases and specifies how these tasks contribute to the project V&V goals. The methods to be used for performing these V&V activities, the responsibilities and milestones for each of these activities, inputs and outputs for each V&V task, and criteria for evaluating the outputs are also specified.

The two major V&V approaches are testing and inspections. Testing is an activity that can be generally performed only on code. It is an important activity and is discussed in detail in a later chapter. Inspection is a more general activity that can be applied to any work product, including code. Many of the V&V tasks are such that for them, an inspection type of activity is the only possible way to perform the tasks (e.g. trace ability and document evaluation). Due to this, inspections play a significant role in verification.

## 3.9 System Implementation

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. System implementation refers to the fourth phase of the systems development life cycle, in which the information system is programmed, tested, installed and supported. During the life cycle, the reporting requirements are specified and reports produced.

### 3.9.1 System Implementation Maintenance and Review

As we know, creating software is one thing and the implementation of the created software is another. The process of implementing software is much difficult as compared to the task of creating the project. First we have to implement the software on a small scale for removing the bugs and other errors in the project and after removing them we can implement the software on a large scale. Before we think in terms of implementing the Software on a large basis, we must consider the Hardware requirements.

Whenever we develop software or project a certain hardware and software is being used by the programmer for developing the project. The hardware and software to be used by the programmer for developing the project should be such that it would result in the development of a project, which would satisfy all the basic needs for which the project has been created by the programmer. The Hardware should be such that cost constraints of the Client should also be taken into account without affecting the performance.

### 3.9.2 Hardware Evaluation Factors

When we evaluate computer hardware, we should first investigate specific *physical and performance* characteristics for each hardware component to be acquired. These specific questions must be answered concerning many important factors. These *hardware evaluation factors* questions are summarized in the below figure.

Notice that there is much more to evaluating hardware than determining the fastest and cheapest computing device. For e.g. the question of possible obsolescence must be addressed by making a technology evaluation. The factor of *ergonomics* is also very important. Ergonomics is the science and technology that tries to ensure those computers and other technologies are "user-friendly", that is safe, comfortable and easy to use. *Connectivity is* another important evaluation factor, since so many computer systems are now interconnected within wide area or local area telecommunications networks.

Hardware evaluation factors:

1) Performance

2) Cost

3) Reliability

4) Availability

5) Compatibility

6) Modularity

7) Technology

8) Ergonomics

9) Connectivity

10) Environmental requirements

11) Software

12) Support

### 3.9.3 Software Evaluation Factors

Software can be evaluated according to many factors similar to the hardware evaluation. Thus the factors of *performance, cost, reliability, compatibility, modularity, technology, ergonomics, and support* should be used to evaluate proposed software acquisitions. In addition, however, *the software evaluation factors* are summarized in below figure. For e.g. some software packages require too much memory capacity and are notoriously slow, hard to use, or poorly documented. They are not a good selection for most end users, even if offered at attractive prices.

1) Efficiency: is the software a well-written system of computer instructions that does not use much memory capacity or CPU time?

2) Flexibility: can it handle its processing assignments easily without major modifications?

3) Security: does it provide control procedures for errors, malfunctions and improper use?

4) Language: do our computer programmers and users write it in a programming language that is used?

5) Documentation: is the s/w well documented? Does it include helpful user instructions?

6) Hardware: does existing hardware have the features required to best use this software?

7) Other characteristics of hardware such as its performance, what about the cost, how much is reliable and etc.

## 3.10 Security Measures

Security Performed in Kiosk Transaction Management & Utility Bills Collection System are as follows:

1. User Name & Password security implemented so that no unauthorized person

2. Cannot handle any operation without username and Password

3. Only authorized person can log-on the system

4. Only authorized person can update the records

5. Only authorized person can handle the reservation

6. Only authorized person can print the report

It has two kinds of users:

1. Administrator

2. User

**Administrator:** He has complete authority of operating the software. The User Name and Password provided to the Administrator in this project is:

| | |
|---|---|
| Username: | admin |
| Password: | 123 |

**User1:** When this user logs onto the system, he cannot found rights to setup branch, user create etc.

| | |
|---|---|
| Username: | user |
| Password: | 123 |

## 3.11 Data Flow Diagram

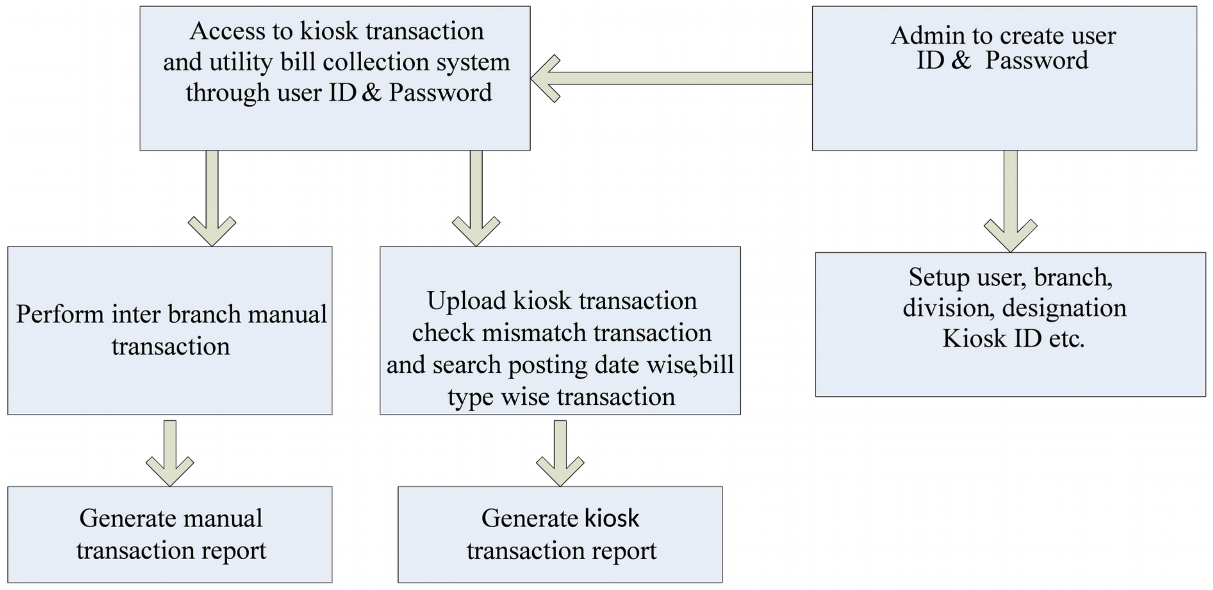

Access to kiosk transaction and utility bill collection system through user ID & Password
Admin to create user ID & Password
Perform inter branch manual transaction
Upload kiosk transaction check mismatch transaction and search posting date wise,bill type wise transaction
Setup user, branch, division, designation Kiosk ID etc.
Generate manual transaction report
Generate kiosk transaction report

Figure: 3.2 Data Flow Diagram

## 3.12 Entity Relationship Diagram

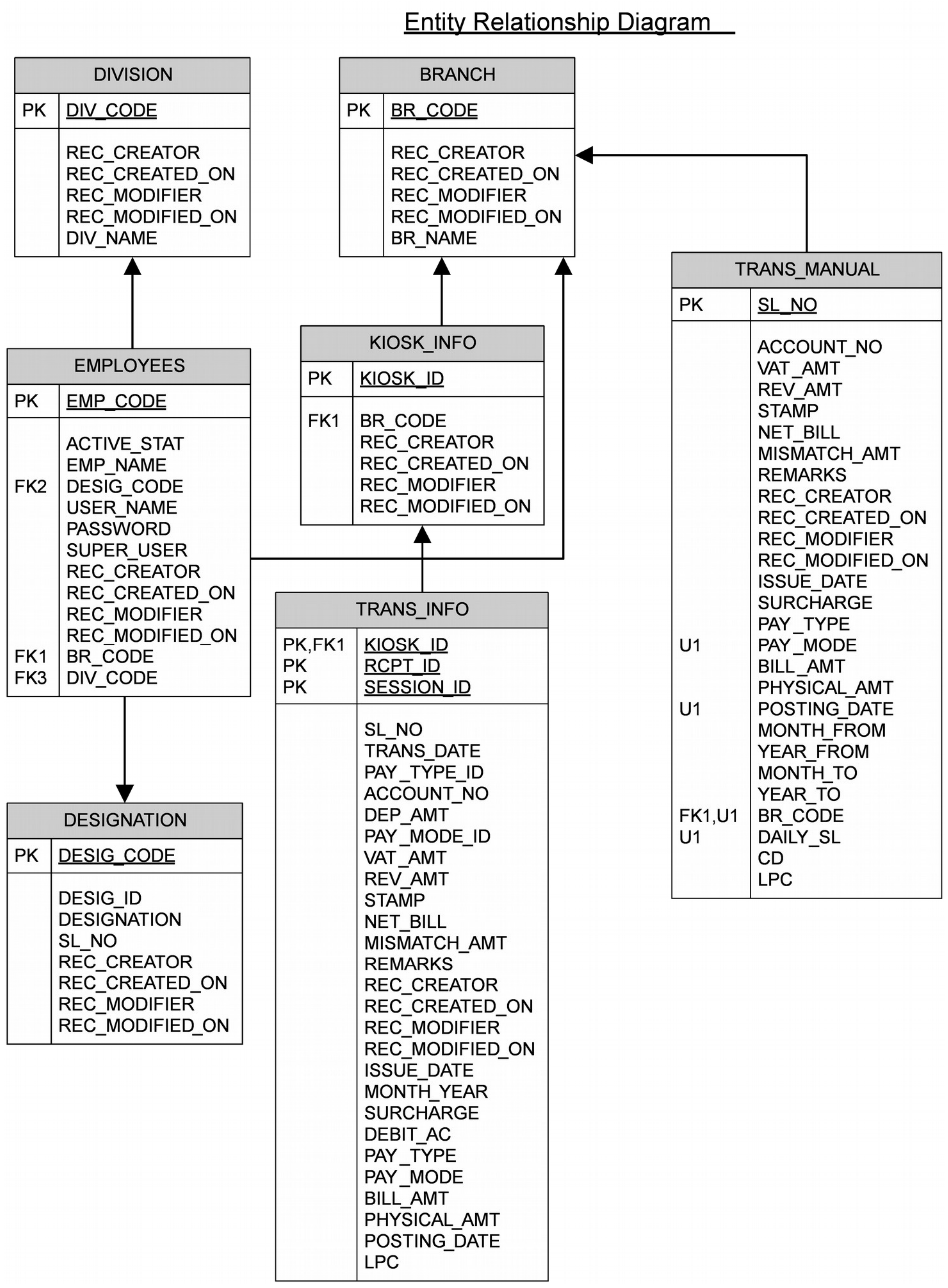


Figure: 3.3 Entity Relationship Diagram

## 3.13 System Requirements

**Minimum Hardware requirements: (Computer)**

| | | |
|---|---|---|
| Processor | : | Intel Core i3, 3.5 GHz or above |
| RAM | : | 2 GB RAM or above |
| Cache Memory | : | 3 MB L3 |
| Hard Disk | : | 500 GB (minimum 3 GB free space) |
| Printer | : | LaserJet Printer |

**SOFTWARE**

| | | |
|---|---|---|
| Operating System | : | Windows XP, Windows 7 or 8 |
| Font-End Tool | : | Oracle Forms developer (10g) |
| Back-End | : | Oracle 10g |

**FRONT END**

I have used Oracle Forms developer (10g) in the Font-End Tool. It is web based, event driven, platform independent.

**BACK END**

I have used Oracle 10g. Oracle 10g provides efficient/effective solution for major database tech.

- Large database and space management.
- Many concurrent database users.
- High transaction processing requirement
- High Availability
- Industry accepted standards
- Manageable security
- Portability

# User Manual

## 4.1 Introduction

User manual will give step by step workflow of Kiosk Transaction Management and Utility Bills Collection System to users. This Software is web based application. Below are the steps to manage of this application.

## 4.2User Manual of web based Application

### 4.2.1 Login Form

When application is launched, then login form is opened. Here user will enter his/her username and password. This form will authenticate user's credential and will forward to main form after successful login.

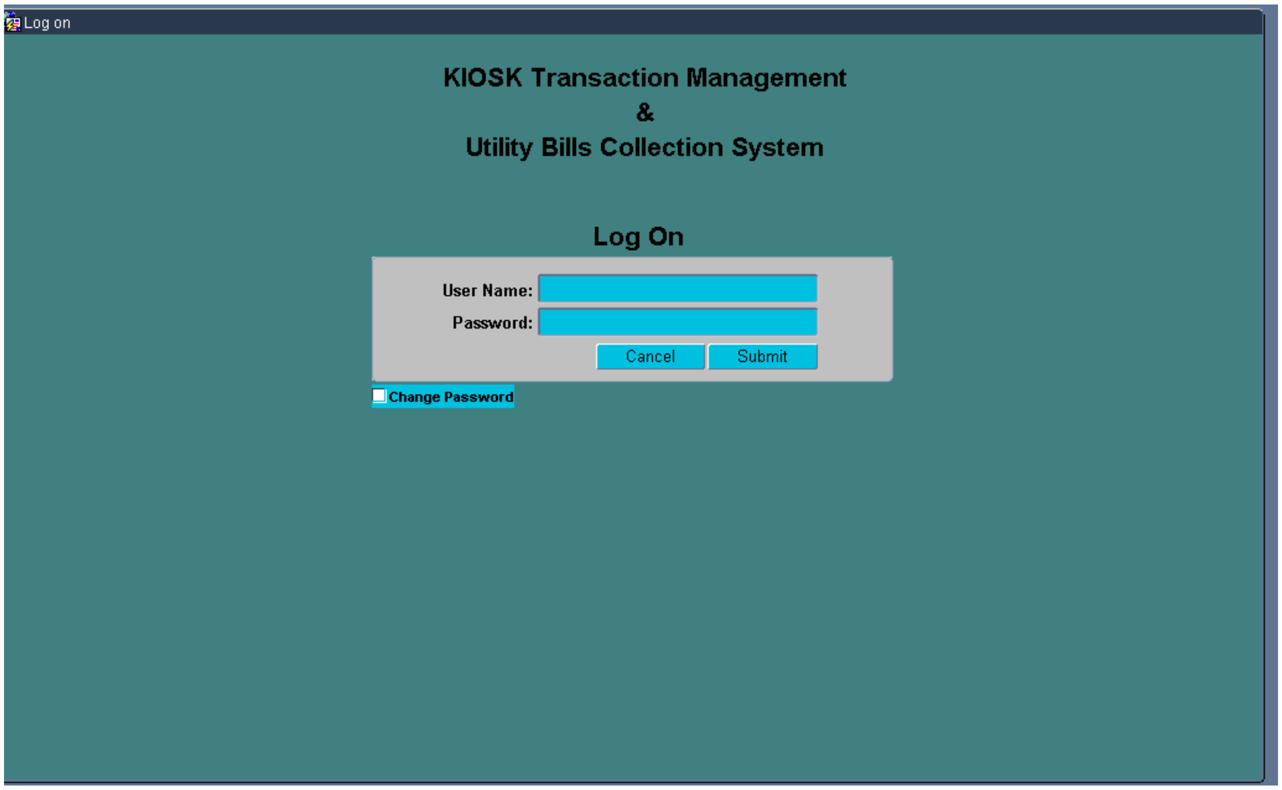
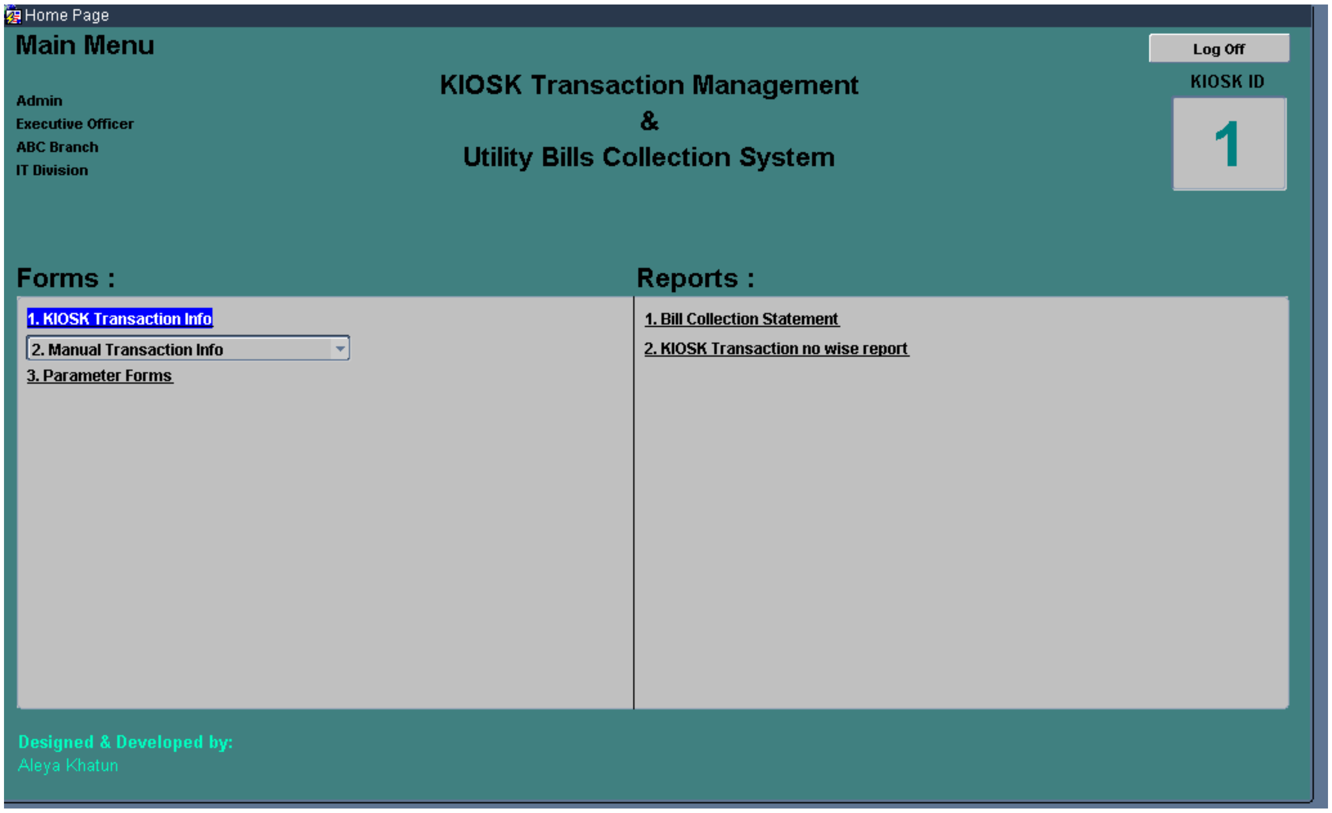


Figure: 4.2.1 Login Form

### 4.2.2 Main Menu

After success full login, main form will be opened. This main form is menu based and shows Kiosk ID. User can navigate into different forms from this main form. Main form shows logged in user's full information. User can log out by this form.



Home Page

Main Menu

Log Off

KIOSK ID

1

Admin
Executive Officer
ABC Branch
IT Division

KIOSK Transaction Management
&
Utility Bills Collection System

Forms :

1. KIOSK Transaction Info
2. Manual Transaction Info
3. Parameter Forms

Reports :

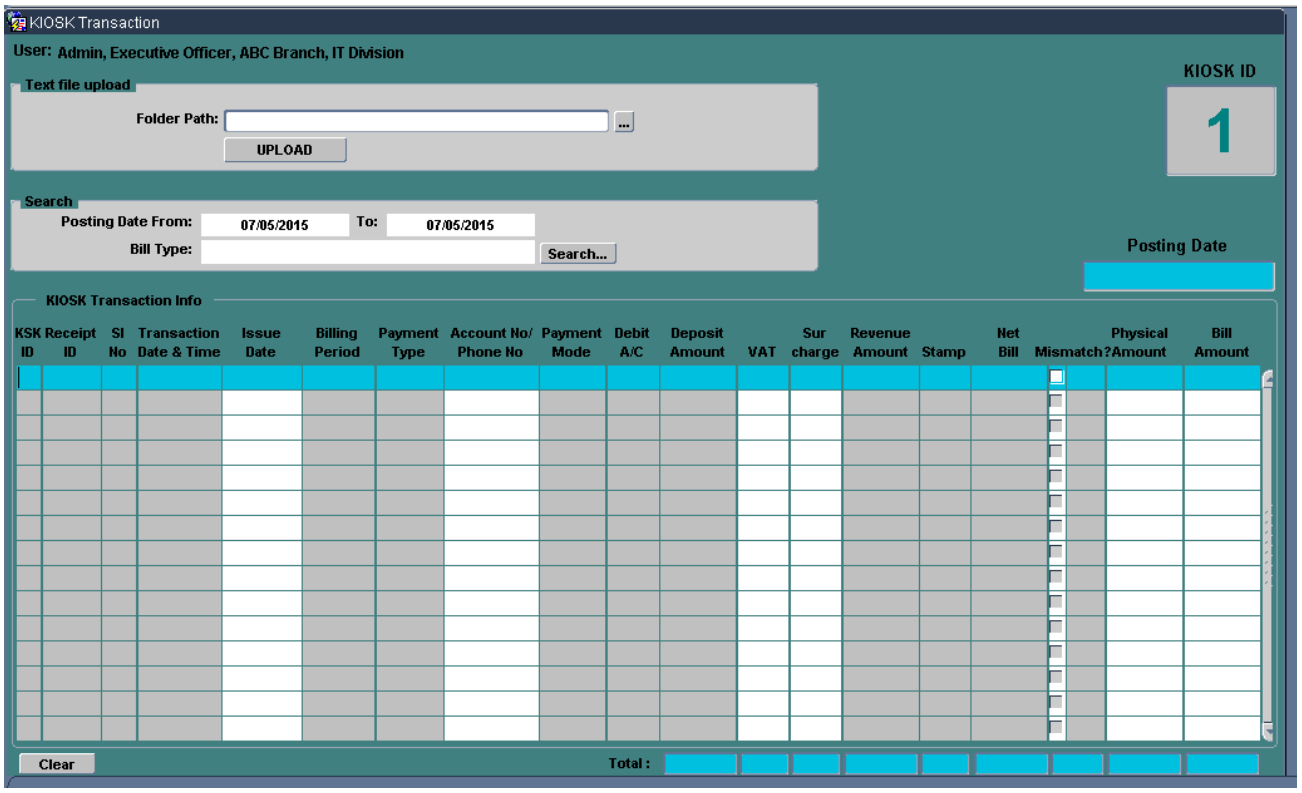
1. Bill Collection Statement
2. KIOSK Transaction no wise report

Designed & Developed by:
Aleya Khatun

Figure: 4.2.2 Main Menu Form

### 4.2.3 KIOSK Transaction Info

All Kiosk Transaction is managed by this screen. First browse text file by using browse button and upload data in database by clicking upload button. After press save button data is uploaded in database. After counting money if physical amount and deposit amount is not same then Input physical amount field value. If physical amount and Deposit amount are not same then tick mark are posted automatically in the in the mismatch field. If physical amount value is less than deposit amount, transaction cannot be posted and account cannot be deposited on that posted day.  After the payment of mismatch amount, transactions are completed. We can also search transaction by posting date or bill type wise. Using clear button we can clear screen.


KIOSK Transaction

User: Admin, Executive Officer, ABC Branch, IT Division

Text file upload

Folder Path: ...

UPLOAD

KIOSK ID

1

Search

Posting Date From: 07/05/2015 To: 07/05/2015

Bill Type: Search...

Posting Date

KIOSK Transaction Info

| KSK ID | Receipt ID | Sl No | Transaction Date & Time | Issue Date | Billing Period | Payment Type | Account No/ Phone No | Payment Mode | Debit A/C | Deposit Amount | VAT | Sur charge | Revenue Amount | Stamp | Net Bill | Mismatch? | Physical Amount | Bill Amount |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Clear

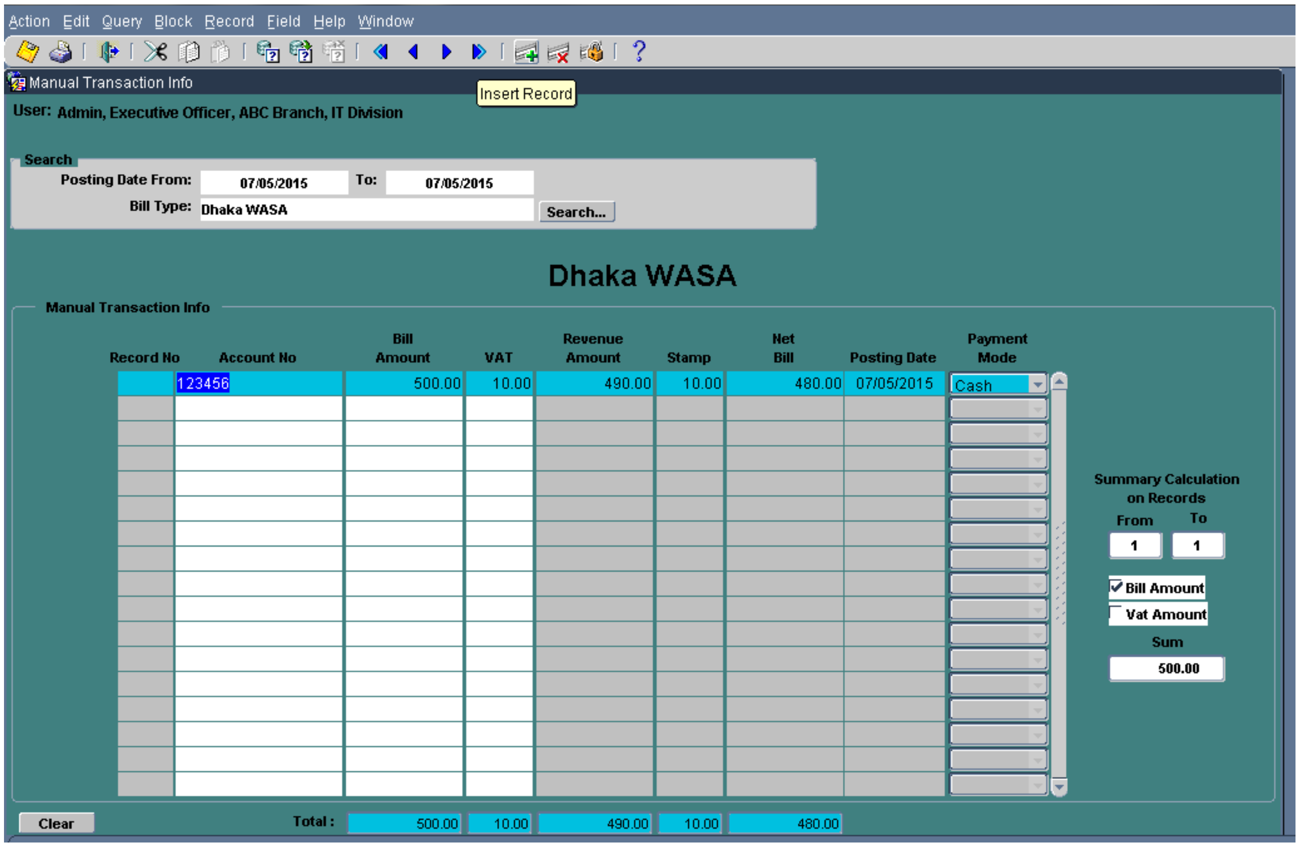
Total :

Figure: 4.2.3 Kiosk Transaction Information Form

### 4.2.4 Manual Transaction Information:

#### 4.2.4.1 Dhaka WASA bill Payment Form

WASA bill is collected by this screen. Using insert option record is inserted and using save button record is saving in database. We can also calculate summery on record wise by click on bill amount option and insert number of record in the **From and To** Field in the right side of the form. Total Bill amount, VAT amount, Revenue amount, Stamp amount, Net Bill amount are shown below in the form. We can also search transaction by posting date or bill type wise. Using clear button we can clear screen.



User: Admin, Executive Officer, ABC Branch, IT Division

Search

Posting Date From: 07/05/2015 To: 07/05/2015

Bill Type: Dhaka WASA

Search...

Dhaka WASA

Manual Transaction Info

| Record No | Account No | Bill Amount | VAT | Revenue Amount | Stamp | Net Bill | Posting Date | Payment Mode |
|---|---|---|---|---|---|---|---|---|
| | 123456 | 500.00 | 10.00 | 490.00 | 10.00 | 480.00 | 07/05/2015 | Cash |
| Total : | | 500.00 | 10.00 | 490.00 | 10.00 | 480.00 | | |

Summary Calculation on Records

From 1 To 1

[✓ Bill Amount]

Vat Amount

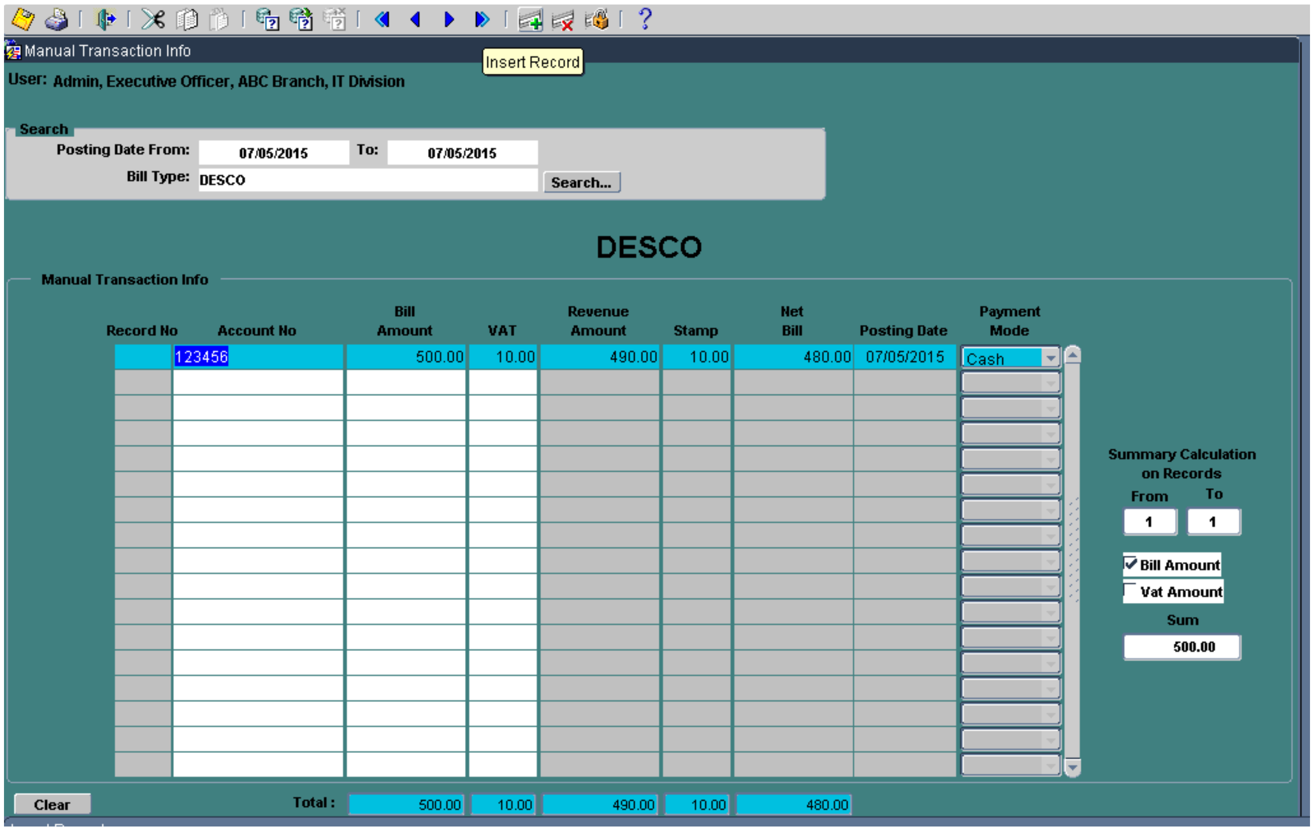
Sum 500.00

Clear

Figure: 4.2.4.1 WASA bill payment Form

### 4.2.4.2 DESCO bill Payment Form

DESCO bill is collected by this screen. Using insert option record is inserted and using save button record is saving in database. We can also calculate summery on record wise by click on bill amount option and insert number of record in the **From and To** Field in the right side of the form. Total Bill amount, VAT amount, Revenue amount, Stamp amount, Net Bill amount are shown below in the form. We can also search transaction by posting date or bill type wise. Using clear button we can clear screen.


Manual Transaction Info

Insert Record

User: Admin, Executive Officer, ABC Branch, IT Division

Search

Posting Date From: 07/05/2015 To: 07/05/2015

Bill Type: DESCO

**DESCO**

Manual Transaction Info

| Record No | Account No | Bill Amount | VAT | Revenue Amount | Stamp | Net Bill | Posting Date | Payment Mode |
|---|---|---|---|---|---|---|---|---|
| | 123456 | 500.00 | 10.00 | 490.00 | 10.00 | 480.00 | 07/05/2015 | Cash |

| | | | | | |
|---|---|---|---|---|---|
| Total : | 500.00 | 10.00 | 490.00 | 10.00 | 480.00 |

Summary Calculation on Records

From: 1 To: 1

[✓ Bill Amount]

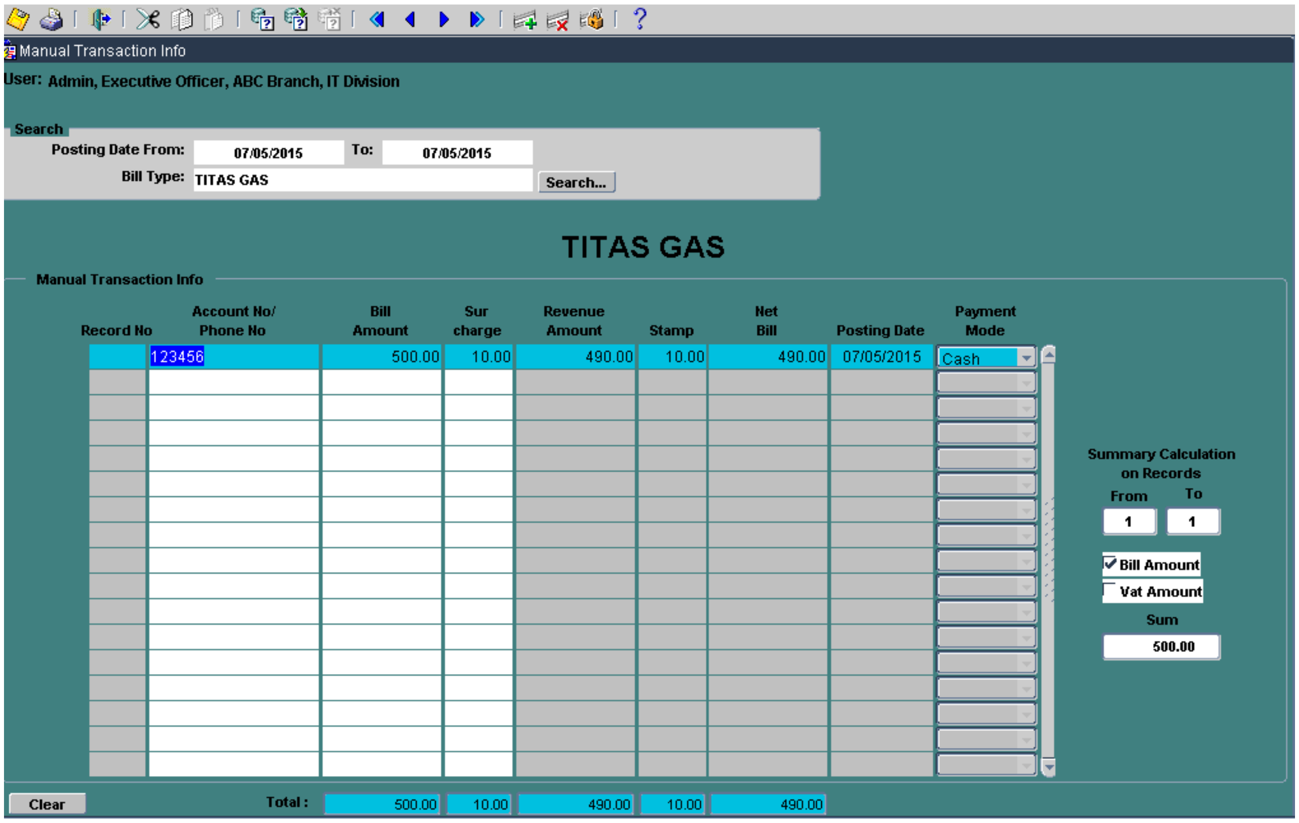
Vat Amount

Sum: 500.00

Figure: 4.2.4.2 DESCO bill Payment Form

### 4.2.4.3 TITAS GAS bill Payment Form

TITAS GAS bill is collected by this screen. Using insert option record is inserted and using save button record is saving in database. We can also calculate summery on record wise by click on bill amount option and insert number of record in the **From and To** Field in the right side of the form. Total Bill amount, VAT amount, Revenue amount, Stamp amount, Net Bill amount are shown below in the form. We can also search transaction by posting date or bill type wise. Using clear button we can clear screen.



Manual Transaction Info

User: Admin, Executive Officer, ABC Branch, IT Division

Search

Posting Date From: 07/05/2015 To: 07/05/2015

Bill Type: TITAS GAS

TITAS GAS

Manual Transaction Info

| Record No | Account No/ Phone No | Bill Amount | Sur charge | Revenue Amount | Stamp | Net Bill | Posting Date | Payment Mode |
|---|---|---|---|---|---|---|---|---|
|  | 123456 | 500.00 | 10.00 | 490.00 | 10.00 | 490.00 | 07/05/2015 | Cash |
| Total : |  | 500.00 | 10.00 | 490.00 | 10.00 | 490.00 |  |  |

Summary Calculation on Records

From: 1 To: 1

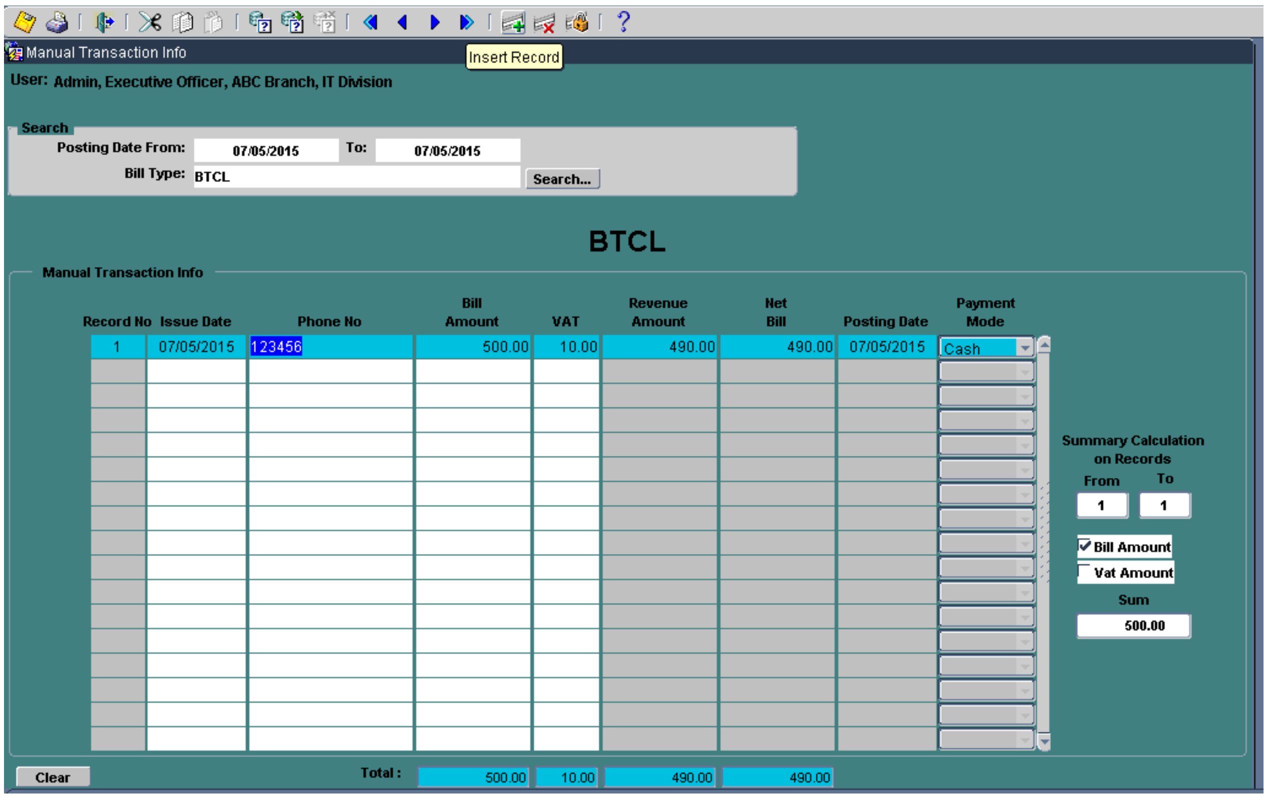
[✓ Bill Amount] Vat Amount

Sum: 500.00

Figure: 4.2.4.3 TITAS GAS bill Payment Form

### 4.2.4.4 BTCL bill Payment Form

BTCL bill is collected by this screen. Using insert option record is inserted and using save button record is saving in database. We can also calculate summery on record wise by click on bill amount option and insert number of record in the **From and To** Field in the right side of the form. Total Bill amount, VAT amount, Revenue amount, Stamp amount, Net Bill amount are shown below in the form. We can also search transaction by posting date or bill type wise. Using clear button we can clear screen.


Manual Transaction Info

Insert Record

User: Admin, Executive Officer, ABC Branch, IT Division

Search

Posting Date From: 07/05/2015 To: 07/05/2015

Bill Type: BTCL Search...

BTCL

Manual Transaction Info

| Record No | Issue Date | Phone No | Bill Amount | VAT | Revenue Amount | Net Bill | Posting Date | Payment Mode |
|---|---|---|---|---|---|---|---|---|
| 1 | 07/05/2015 | 123456 | 500.00 | 10.00 | 490.00 | 490.00 | 07/05/2015 | Cash |

Summary Calculation on Records

From 1 To 1

✓ Bill Amount

Vat Amount

Sum 500.00

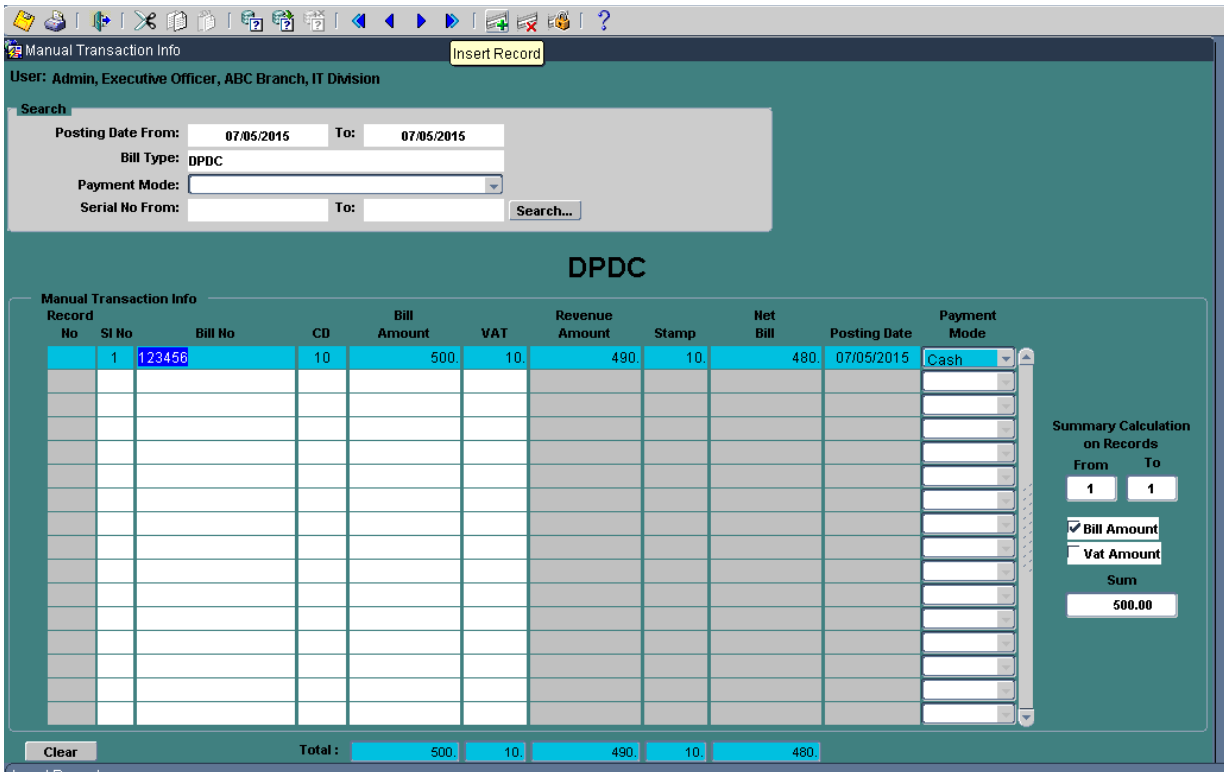
Clear

| Total : | 500.00 | 10.00 | 490.00 | 490.00 |
|---|---|---|---|---|

4.2.4.4 BTCL bill Payment Form

### 4.2.4.5 DPDC bill Payment Form

DPDC bill is collected by this screen. Using insert option record is inserted and using save button record is saving in database. We can also calculate summery on record wise by click on bill amount option and insert number of record in the **From and To** Field in the right side of the form. Total Bill amount, VAT amount, Revenue amount, Stamp amount, Net Bill amount are shown below in the form. We can also search transaction by posting date or bill type wise. Using clear button we can clear screen.


Insert Record

Manual Transaction Info

User: Admin, Executive Officer, ABC Branch, IT Division

Search

Posting Date From: 07/05/2015 To: 07/05/2015

Bill Type: DPDC

Payment Mode:

Serial No From: To: Search...

DPDC

Manual Transaction Info

| Record No | Sl No | Bill No | CD | Bill Amount | VAT | Revenue Amount | Stamp | Net Bill | Posting Date | Payment Mode |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 123456 | 10 | 500. | 10. | 490. | 10. | 480. | 07/05/2015 | Cash |

Summary Calculation on Records

From 1 To 1

[✓ Bill Amount]

Vat Amount

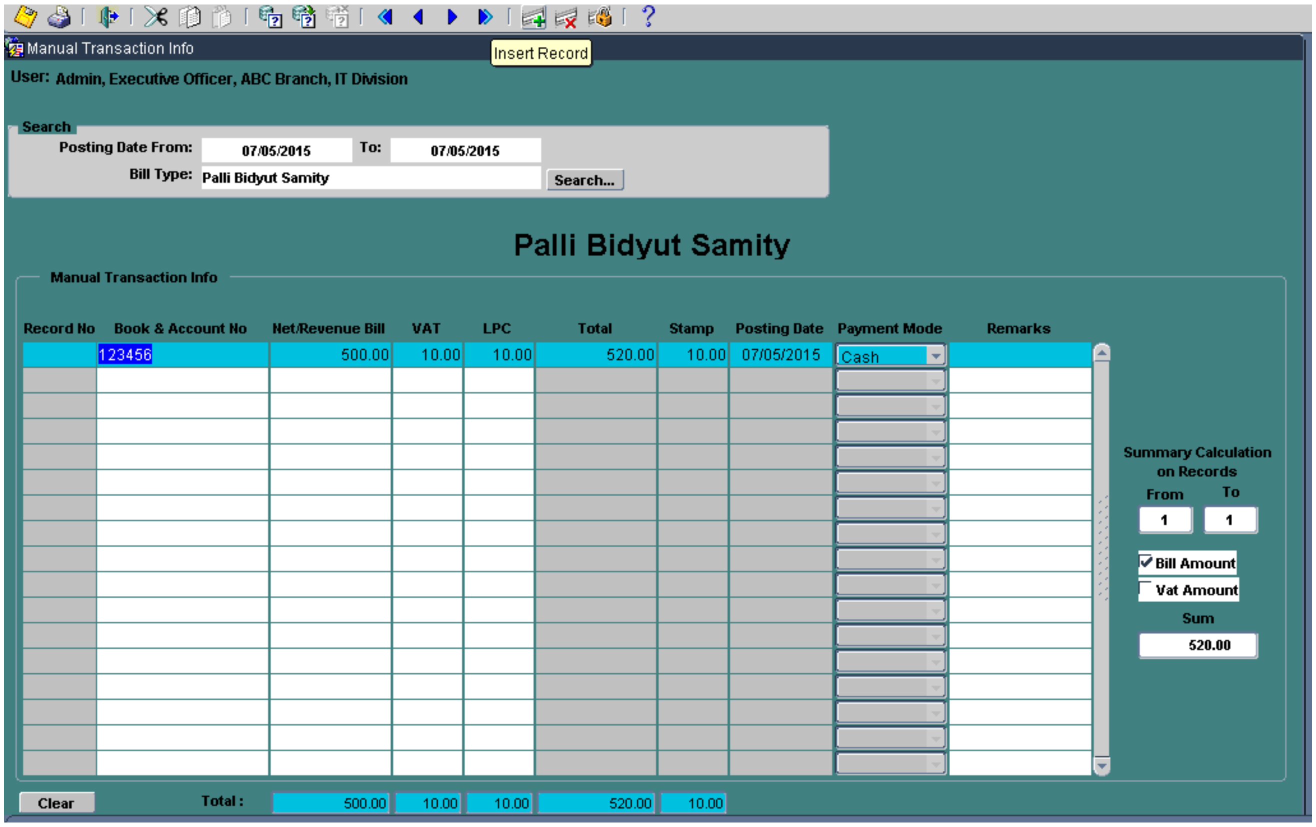
Sum 500.00

Clear Total : 500. 10. 490. 10. 480.

Figure: 4.2.4.5 DPDC bill Payment Form

### 4.2.4.6 Palli Bidyut Samity bill Payment Form

Palli Bidyut Samity bill is collected by this screen. Using insert option record is inserted and using save button record is saving in database. We can also calculate summery on record wise by click on bill amount option and insert number of record in the **From and To** Field in the right side of the form. Total Bill amount, VAT amount, Revenue amount, Stamp amount, Net Bill amount are shown below in the form. We can also search transaction by posting date or bill type wise. Using clear button we can clear screen.


Manual Transaction Info

Insert Record

User: Admin, Executive Officer, ABC Branch, IT Division

Search

Posting Date From: 07/05/2015 To: 07/05/2015

Bill Type: Palli Bidyut Samity

Search...

Palli Bidyut Samity

Manual Transaction Info

| Record No | Book & Account No | Net/Revenue Bill | VAT | LPC | Total | Stamp | Posting Date | Payment Mode | Remarks |
|---|---|---|---|---|---|---|---|---|---|
| | 123456 | 500.00 | 10.00 | 10.00 | 520.00 | 10.00 | 07/05/2015 | Cash | |

Summary Calculation on Records

From: 1 To: 1

[✓ Bill Amount]

Vat Amount

Sum: 520.00

Clear

Total : 500.00 | 10.00 | 10.00 | 520.00 | 10.00

Figure: 4.2.4.6 Palli Bidyut Samity bill Payment Form

### 4.2.4.7 Parameter Forms

Only software administrators are used parameter form. Kiosk information, branch information, division information, designation information, user information etc. is setup by this form.

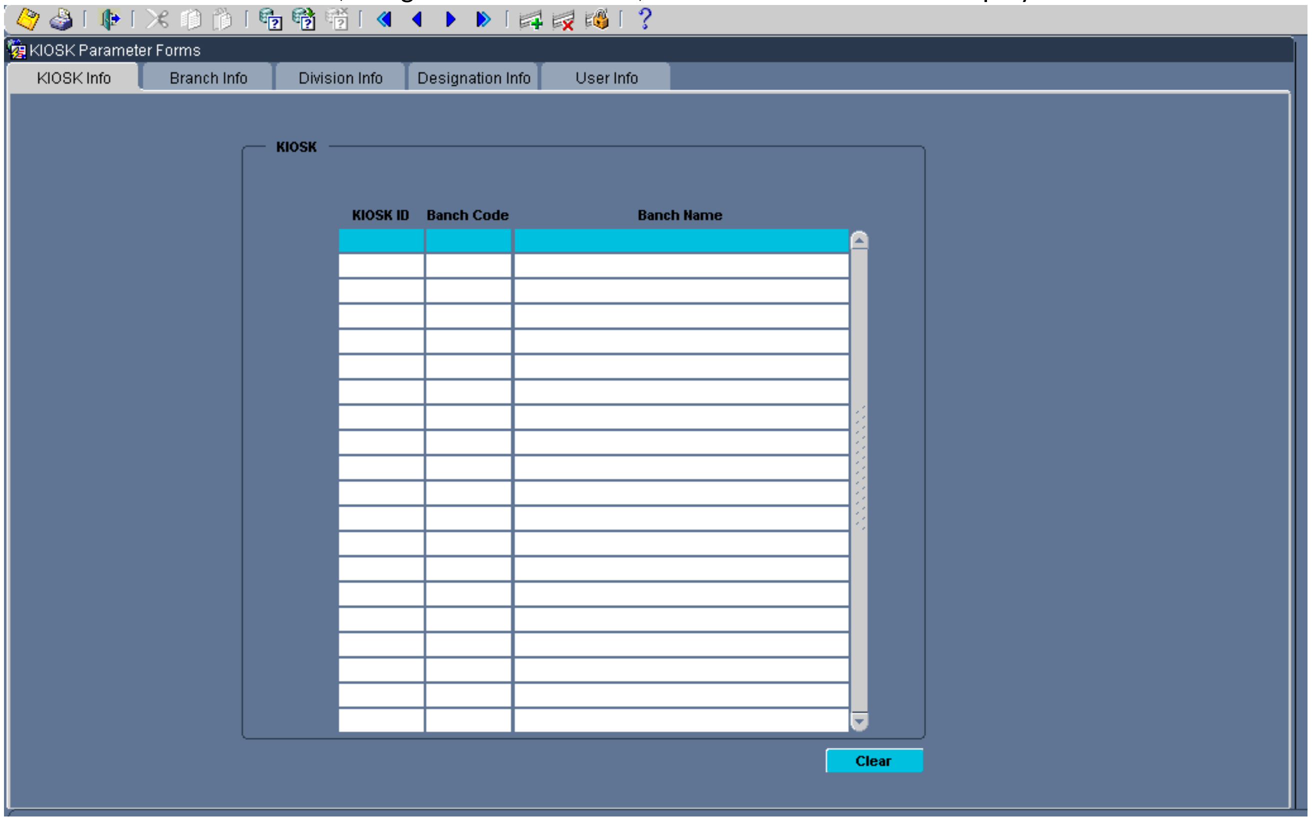
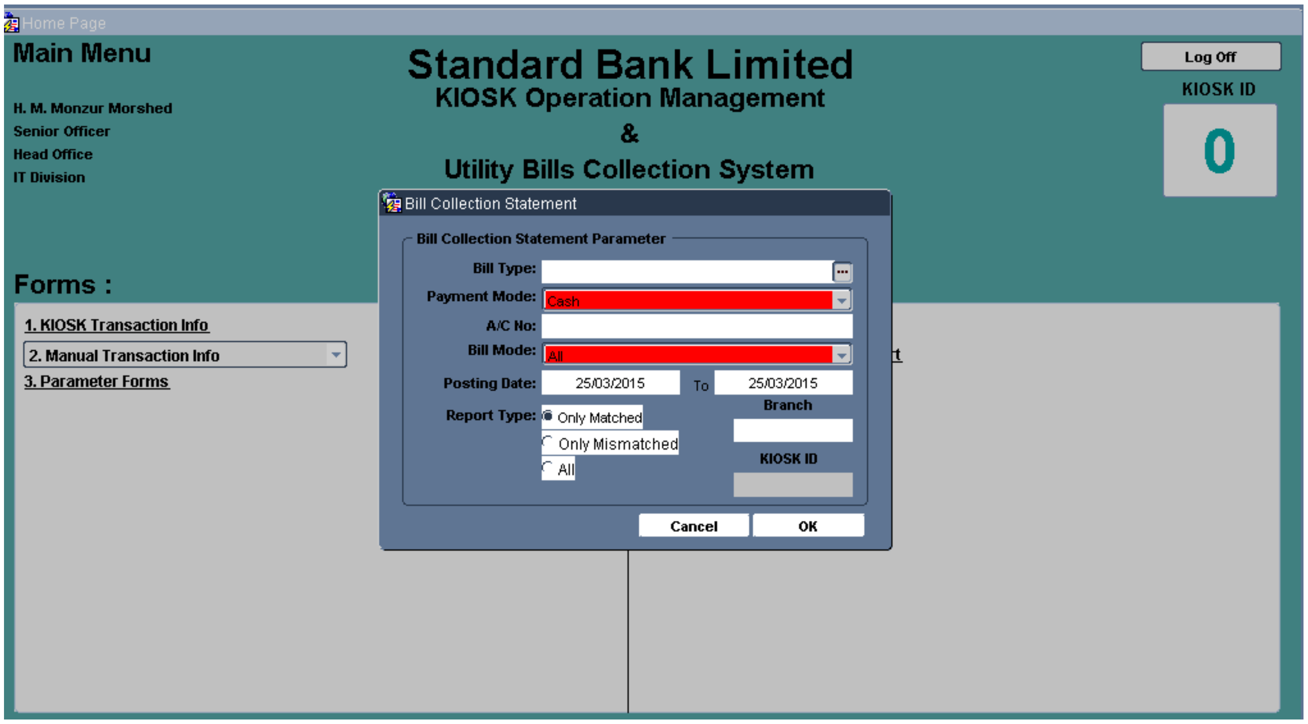


Figure: 4.2.4.7 Parameter Forms

### 4.2.4.8 Reports:

From the 'Reports' part of this software users can take various reports related to kiosk transaction information and manual of transactional information. 'Bill collection statement' shows type wise all bills transaction and 'kiosk transaction no wise report' shows bill type wise report only kiosk transaction.


Home Page
Main Menu
H. M. Monzur Morshed
Senior Officer
Head Office
IT Division
Standard Bank Limited
KIOSK Operation Management
&
Utility Bills Collection System
Log Off
KIOSK ID
0
Forms :
1. KIOSK Transaction Info
2. Manual Transaction Info
3. Parameter Forms
Bill Collection Statement
Bill Collection Statement Parameter
Cancel
OK

Bill Type:

Payment Mode: Cash

A/C No:

Bill Mode: All

Posting Date: 25/03/2015 To 25/03/2015

Branch

Report Type: [✓ Only Matched] Only Mismatched All

KIOSK ID

Figure: 4.2.4.8 Reports

# Conclusion and Future Work

## 5.1 Conclusion

Kiosk Transaction Management & Utility Bills Collection System is ultra-modern software that offers comprehensive solutions to manage transaction in bank. This system is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to banks

This offers various sub-systems and a seamless integration. By being modular, each module can be used as a standalone solution or can be integrated in a phased manner. Modules are also so designed to meet the present as well as future requirements of the organization and process a unique ability with the business growth.

This is a great improvement over the manual system. The computerization of the system has speed up the process. In the current manual system, transaction managing is difficult. This system was thoroughly checked and tested with dummy data and thus is found to be very reliable.

We can get following advantages using this system:

- It is fast, efficient and reliable
- Avoids data redundancy and inconsistency
- Very user-friendly
- Easy accessibility of data
- Number of personnel required is considerably less
- Provides more security and integrity to data

This system not only provides an opportunity to the bank but also can increase the profitability of the organization. Reporting sometimes becomes awfully pathetic and complex. This product will eliminate any such complexity to generate different type wise report. Very important for some, the reduced cost of the manpower would pay for the cost of this product within a short time after its implementation.

## 5.2 Future Work

Considering the availability of mobile phones in the hand of every individual, the bank has already introduced SMS and Alert banking for the convenience of the customer. This is becoming very popular and useful means of enquiring bank account information. With this small device a customer can enquire about his account balance, see on the screen last few transactions, transfer funds, pay utility bills and many more.

We will develop our system in future such that clients will be made sure over SMS after the payment received by our system.

## References

1. http://www.grameenphone.com/personal/vas/mobile-financial-services/bill-pay

2. http://www.dutchbanglabank.com/electronic-banking/sms-alert.html

3. http://www.innova.com.tr/kiosk/payment-kiosk/bill-payment-kiosk.asp

4. http://www.cusi.com/payment-services/payment-kiosks.html

5. http://www.atmexperts.com/bill-payment-kiosk.html

6. http://www.everestadvanced.co.in/advanced-bill-payment-solution.html