VoIP Billing System

ومن يؤت الحكمة
فقد أوتي خيرا كثيرا

QUAID-I-AZAM UNIVERSITY

Muhammad Ali

A report submitted to the
Department of Computer Science, Quaid-I-Azam University Islamabad.
As a partial fulfillment of the requirement for the award of the
Master degree in Computer Sciences

(February 2004)

QUAID-I-AZAM UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Dated: March 22, 2004

FINAL APPROVAL

This is to certify that we have read the project report submitted by Mr. Muhammad Ali and it is our judgment that this report is of sufficient standard to warrant its acceptance by the Quaid-i-Azam University, Islamabad for the degree of Master of Science in Computer Science.

COMMITTEE:

1. External Examiner

Dr. A. Sattar ____________________
Consultant
Multimedia System
Computer Science Department
Allama Iqbal Open University
Islamabad

2. Supervisor

Mr. Ghazanfar Farooq ____________________
Lecturer
Department of Computer Science
Quaid-i-Azam University
Islamabad

3. Chairperson

Professor Dr. Farhana Shah ____________________
Department of Computer Science
Quaid-i-Azam University
Islamabad

In The Name of Allah, the Most Beneficent, the Most Merciful.

This thesis is dedicated to my wonderful parents, brothers and sisters.
Thank you for all the years of love and support.

Acknowledgement

All praises to Almighty **ALLAH** Who Blessed me with the knowledge for which I cannot deserve. Infinite thanks to Almighty **GOD** Whose Blessings has made me able to complete this task and finally blessed me to complete my M.Sc.

I salute the greatness, patience and love of my great ***mother and father***, for whom, I have no words to say thanks. I cannot forget the love and patient of my whole family during my education. I especially want to thank my brother **Ahmed and Shahid** and my beloved sisters whose love always encourage me to go for best.

Thanks to my all those teachers whose love and trust made me able to reach at this stage. I feel great pleasure and honor to express my sincere gratitude to ***Mr.Ghanzafar Farooq*** who gave me the opportunity to prove myself. He remained very kind to me during my project development.

This task was not possible without the unforgettable and great co-operation of my external supervisors in Advanced Communications. I am especially thankful to my great seniors ***Mannan*** Bhai, ***Naveed*** Bhai, ***Javad*** Bhai, and ***Mazhar*** Bhai in Advanced Communications who remained very kind and helpful to me in my project period. I am also truly grateful to ***Haris Bhai*** and ***Ammar Bhai*** whose support able me to achieve my goal. I will never forget the time I spend like a brother with ***Abdullah*** Bhai, ***Usman*** Bhai, ***Nazim*** Bhai, ***Shahbaz*** Bhai, ***Saqib*** Bhai and all seniors. I can never forget my loving and sincere friends ***Ghazanfar***, ***Akbar***, ***Mohzzam***, ***Kashif***, ***Faizi***, ***Tayyab***, ***Sajid*** and all my loving class fellows.

Finally, I owe all my academic success and progress in life to my loving ***mother and father***; whose affection, endless prayers, good wishes and inspiration remained with me for higher ideals of life.

Thanks To Every One, With Love,

Muhammad Ali

Contents

Project Brief

Project Title	***VoIP Billing System***
Undertaken By	***Muhammad Ali***
Supervised By	**Mr. Ghanzafar Farooq** Lecturer Quaid-I-Azam University Islamabad.
Offered By	**Mr. Abdul Mannan** Project Manager Advanced Communications F-10/2 Islamabad, Pakistan.
Date Of Commencement	April 2003
Date Of Completion	February 2004
Source Language and tools	JAVA SQL Server 7.0
Operating System used	Windows 2000 Professional
System used	Intel Pentium II, 333 MHz RAM 128 MB

Abstract

VoIP Billing System is software that Authorizes and Accounts the user communications. The Software is capable of providing REAL-TIME flexible Call Billing and Call Time measurement of all client types. The System keeps the track of all Clients whenever their Call is established. Clients may be PINLess-Users (Registered) or PIN-Users (Card Users). Using Destination Number, the system will get the prefix of city/country and matching that prefix in database, its rate will be found out. In case of cards, the system will pre-calculate the balance of card and time on the basis of its destination. But this is not the case of Registered Users; their bill will be calculated after call establishment. Administrator can configure the rates by its own. Administrator can register the clients and generate the series of cards. Also it can view the client records and generate reports.

Preface

This report is concerned with the development of VoIP Billing Software, software that will facilitate the billing of the client communication. This report describes the analysis, design and development of the system. This thesis contains five chapters, appendix, references, webliography and bibliography.

Chapter 1 briefly describes the company introduction, system overview, and system scope and system objectives.

Chapter 2 describes process of Requirements Engineering; includes Requirements Analysis and Requirements Definition including Functional and non-functional requirements.

Chapter 3 describes System Analysis and System Design including, use case identification, description, use case diagram, classes identification, description, class diagram, sequence diagrams, table design and entity relationship diagram.

Chapter 4 describes the implementation details of the system.

Chapter 5 briefly describes the testing of the system, test cases, evaluates the system and describes the future enhancements and suggestions.

Appendix Contains Feasibility Study, Resources detail, Process Model Selection.

Chapter 1

Introduction

This chapter describes the brief introduction of the Organization, Overview of the system, scope of the system and objectives of the system.

1.1 Organization Overview

Advanced ISP Billing System is a product of Advanced Communications. They are largest ISP Billing solution providers in Pakistan. Advanced ISP Billing is a comprehensive billing package for Internet Service Providers from all over the world. It offers ease of day-to-day operations, superb client management, a whole suite of useful managerial reports, seamless system administration and a lot more. Advanced Radius is also a product of Advanced Communications.

Currently they are providing billing for dial-ups, DSL and wireless service providers in International market.

1.2 Overview - VoIP Billing Software

Voice over IP is a term used in IP telephony for a set of facilities for managing the delivery of voice information using the Internet Protocol (IP). In general, this means sending voice information in digital form in discrete packets rather than in the traditional circuit-committed protocols of the public switched telephone network (PSTN). A major advantage of VoIP and Internet telephony is that it avoids the tolls charged by ordinary telephone service.

VoIP Billing...

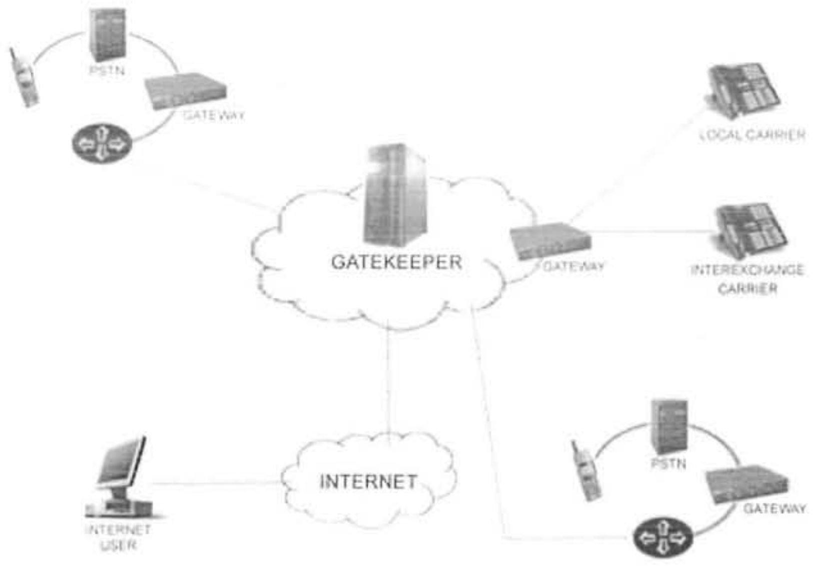
The VoIP Billing System allows Network Service Providers (NSPs) to expand their business via duration-based billing. The Postpaid billing option allows NSPs to bill customers after they make phone calls in real-time, using detailed per-minute call detail records in monthly reports. Also the system supports Prepaid billing allowing customers to prepay for minutes of use - a typical application is a debit card that cancels service when the credit is exhausted.

How it Works?

Post Paid Scenario

Using VoIP, an enterprise positions a "VoIP device" (such as Cisco's AS5300 access server with the VoIP feature) at a gateway. The gateway receives packetized voice transmissions from the calling party using an IP phone. The information is then passes to Billing Software which in turn matches the call number and rate to its internal rate table and passes the information back through a Radius Server to the AS5300. Once this process is complete the calling party is connected to his/her destination.

Upon disconnecting, Radius then sends the final attributes to Billing Software who logs the call duration and rate charged which is used to bill the client in real time.



Pre Paid Scenario

Using VoIP, an enterprise positions a "VoIP device" (such as Cisco's AS5300 access server with the VoIP feature) at a gateway. The gateway receives packetized voice transmissions from the calling party. The AS5300 then prompts the caller for their PIN number, the entered value is then passed onto the Radius server which performs the authentication once confirmed, sends a request back to the AS5300 requesting that the caller now enter the destination number. The information is then passed to Billing System which in turn matches the call number, rate and prepaid time left for the customer, to it's internal rate table and passes the information back through Radius to the AS5300. Finally, the initial process is complete and calling party is told their current prepaid balance and rate of the requested call then connected to his/her destination.

While on a prepaid call the AS5300 will use the attributes that were passed on from the Radius Server to count down the call usage time, and will alert the caller based on pre-defined intervals of time remaining. If the call timer reaches a value of "0" the caller will then be disconnected. However, if the caller disconnects prior to their time running out, the AS5300 will send the final attributes to Billing Software who logs the call duration and rate. These values are then used to adjust the balance of the pre-paid account.

1.3 Scope Of System

Software includes the following key features in the scope

- Can manage the users for both Pin system and Pin-less system.
- Keeps the records of all clients whenever they called according to time and date.
- Provide the facility of configuring rates by the administrator.
- Password protected to prevent unauthorized access.
- Keeps the starting and ending time of each call.
- Provide the facility of viewing database statistics as a report in form of web page.
- Can generate and sale cards having PIN Numbers by the administrator.
- Clients can change their Passwords, PIN Numbers and view their own account records.
- Provide the facility of On-line Client registration
- Provide the facility of assigning Free-Bonuses.
- Provide the facility of On-line Client payment as well as through checks.
- 100% Web-based management.

1.4 Objectives Of System

Software has the following objectives.

- To make a billing software according to organization requirements and customization.
- To provide the facility of flexible REAL-TIME Call Billing and Calling Time measurement.
- To provide a facility of efficient and reliable billing.
- Administrative Interface via Web for user and call rate management
- To provide a facility of an easy monitoring of client records
- Online billing and automated user interaction.
- Easy setup and maintenance

Chapter 2

Requirements Engineering

This chapter describes the requirement analysis, including the detailed requirements of the system and requirement definition describing functional and non-functional requirements.

2.1 Requirement Analysis

"Process of deriving the system requirements through observation of existing system, discussion with potential users and procurers, task analysis and so on."
[Ian Sommerville]

Requirement Analysis is the process of estimating the functionality, through observations, which the system should provide. It is a formal description from the software development team for customer. It is the base to understand the system in terms of function it should provide and their implementation. In order to get a better understanding of the requirements, following requirements are identified by Requirements Analysis.

- Client Management
  - System must provide a support for both PIN and PINless client management.
  - Every client should be registered against some plan.
  - Client can have only one plan at a time.
  - Clients can be of two types Wholesalers and Retailers. Wholesalers must have pinless billing plan. Wholesalers can have IP-based authentication.
  - Corporative accounts can be created.

- Card Series Management
  - System must provide facility of card series management for PIN customers according to different rates, like PTCL Calling Cards have cards of Rs.250, Rs.500 and Rs.1000.
  - System administrator can generate card series of fixed amount and can sell cards.
  - Cards also can have taxes.
  - Each card must have unique PIN Number.

- Call Rates
  - Call rates to same city and other cities and countries should be defined.
  - These rates should be variable and easily editable by billing administrator.
  - Billing system will use these records for accounting purposes, when user dials his/her destination number.

- Call Accounting/Billing
  - When the call is authenticated, radius server receives a session start. Billing System has to get this record from radius server and show on Currently Active Call Status web interface.

  - When user hang-up or CISCO router terminates the call, an accounting record is sent to radius server. Billing system has to get this record and deduct the amount of money as per call time used by the caller and deactivate the user account if his/her balance turned to 0.

- Multiple Reports

  Administrator can generate
  - Call Reports
  - Card Reports
  - Client Reports
  - Payment Reports

- Bill Generation
  - System must be able to generate bills on specific dates having total amount, total call duration and total call charges.

- Receive Payments
  - Clients can pay their payments On-line.
  - Administrator can also receive client payments.

- Free Bonuses
  - Billing Software provides the facility of free bonuses to their clients.

- Security
  - Clients must have their own billing passwords and pin numbers.
  - Administrator also should have login ids and passwords.
  - System should provide the facility of changing passwords and pin numbers.
  - No client can view the details of another client.
  - No client can neither delete anyother or itself nor change its balance.
  - From administrator point of view, there will be a limit for an administrator to create new client accounts up to fixed number.
  - There will be a value of maximum amount of which any administrator can receive in one transaction.

- Administrators can Lock/Unlock clients.
- Application will provide GUI [illegible] administrator for viewing and deletion of records from database.

2.2 Requirements Definition

"Activity of translating the information gathered during the analysis activity into a document that defines a set of requirements."
[Ian Sommerville]

Software requirements definition is an abstract description of the services that the system should provide and constraints under which the system must operate and specify the external behavior of the system.

Software requirements can be broadly categorized into two parts.

- Functional Requirements
- Non-Functional Requirements

2.3.1 Functional Requirements

These are the statements of services the system should provide, these requirements might also explicitly state what the system should not do. Functional requirements definition of the system should be both complete and consistent.

- Billing System will provide flexible and accurate calling time measurement.
- System will provide flexible call billing by storing the balance against each call on server database.
- System will provide the password protected GUI for the easy customization settings to Administrator.
- System will provide the facility of card series generation and selling cards.
- System will give the facility of On-line client registration; receive payments and giving free bonuses.
- System will give the facility to administrator for configuring rates.
- Administrator can register clients; delete client records and lock/unlock clients.
- System will generate the reports, on query by administrator with different options, based on the database statistics.

2.3.2 Non-Functional Requirements

These are the constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, standard and so on. Non-Functional Requirements defines system property and constraints i.e. these are the user requirements which have no concern with the functionality of software. They can be categorized as following.

- Product Requirements
- Process Requirements
- External Requirements

Product Requirements

These are the requirements, related with the need for the delivered product to behave in a particular way.

Usability, it should provide some easy to use GUI to interact with the system and system must practically be implemented on any ISP center and must be integrated with any router and radius server.

Reliability, it must be reliable enough to keep all client information, having their accurate balances and calling time measurement without informing any unauthorized person and fulfill the objectives for the development of the system.

Process Requirements

These are requirements, which are a consequence of organizational policies and procedures.

Delivery, it must be developed and delivered with in 4 to 5 months.

Implementation, JAVA should be selected as system development language.

External Requirements

These requirements arise from factors external to the system and its development process.

Ethical, software code should not be delivered to any one, as organization wants to launch this software after further enhancement as product so code is very much important.

Chapter 3

System Analysis & Design

This chapter will provide detailed system analysis and design in three phases. The first phase describes Use Case Analysis including system use cases, their description and use case diagram. Second phase tells about Object Oriented Design including Classes, their description and Class Relationship Diagram. Third phase describes about Database Design.

3.1 System Analysis

For the analysis and design of system object oriented methodology was established, because it provides a natural way to analyze and design of a real world entities. One of the most fascinating features of this methodology is that it does not try to separate the analysis and design phase, which was implicit in structured approach; rather it provides a smooth transformation from analysis phase. It is easy to understand the system with the help of identifying objects in real life, so, system is modeled using object-oriented methodologies for simplicity.

To represent object oriented analysis and design phase diagram, Unified Modeling Language (UML) used. UML is more precisely specified than any previous object oriented notation.

3.2 Use Case Analysis

A scenario-based description of how actors (Clients and Administrators) interact with the product to be built. Use Cases are defined from an actor's point of view. An actor is a role that interacts with the software. It provides a graphical description of who will use the system and what kind of interaction they can expect to have with the system.

Use-Case analysis phase consists of following stages

- Actors
- Actors description
- Use-Case Identification
- Use-Case diagram

Actors

An actor is someone or something that interacts with the system it's who or what uses the system. By "interacts with the system", we mean that the actor sends or receives the messages to and from the system and exchanges information with the system .In short, actors carry out use cases. Use-Case is always a functionality that actor can use. An actor, who sends or receive message to it, always initiates a use case.

Following are the actors identified for the system

- Administrator
- Client

Actors' description

Administrator:

Administrator will be a controller of the Billing Software who will after login can change the monitoring settings of system, view reports, generate & sell cards, delete client records, receive payments and can also change passwords.

Client:

Billing is done for clients. Clients of the system can view their balance, can pay their bills online through Web Interface. Clients can change their Billing Passwords and Pin Numbers.

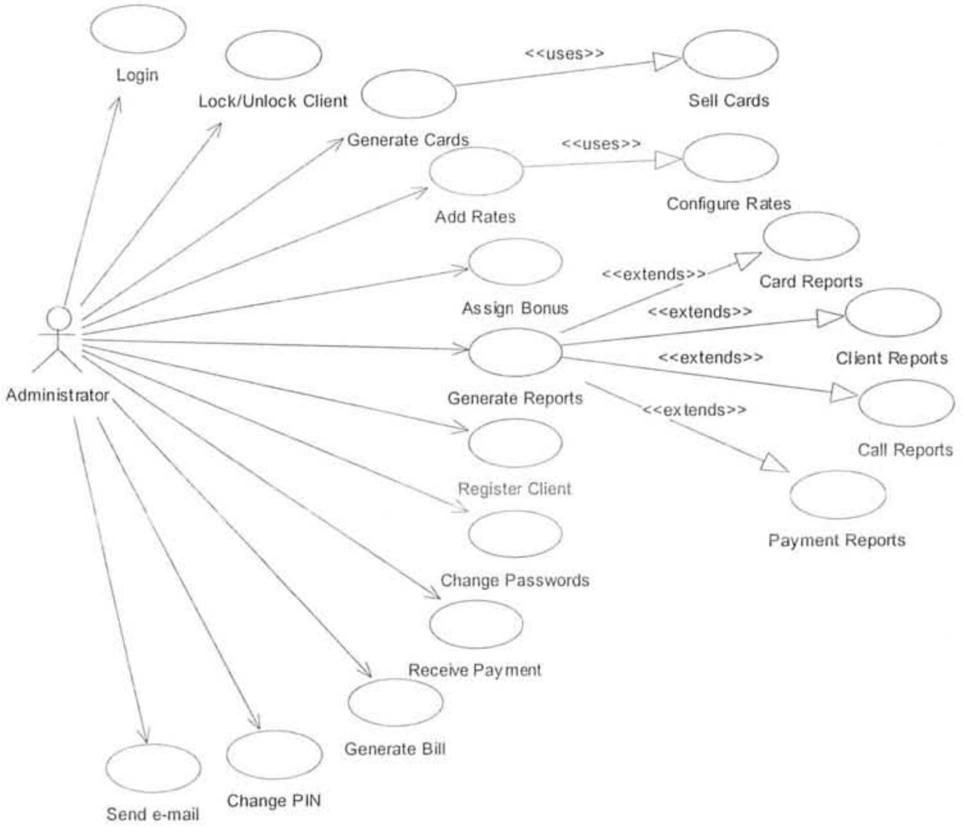
3.3 Use-Case Identification

Use-Cases explain the major functionality performed by the system. Following use-cases are identified in order to explain the abstract functionality of the software.

- Log-in
- Change Password
- Generate Report
- View Report
- Client Registration
- Receive Payment
- Configure Rates
- Generate Cards
- Sell Cards
- Add new Rates
- Lock/Unlock Clients
- Delete Client Records
- View Records
- Generate Bills
- Generate Free Bonuses

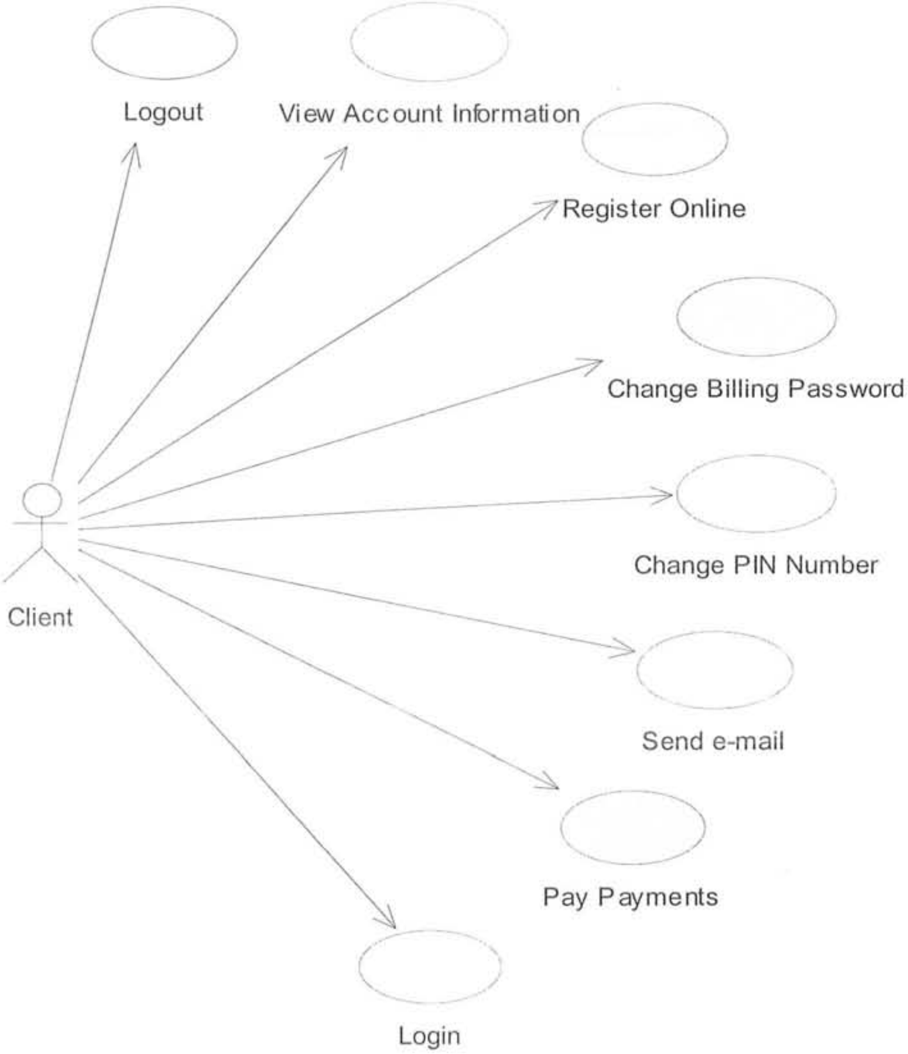
3.4 Use Case Diagram

Use-Case Relationship Diagram of Administrator



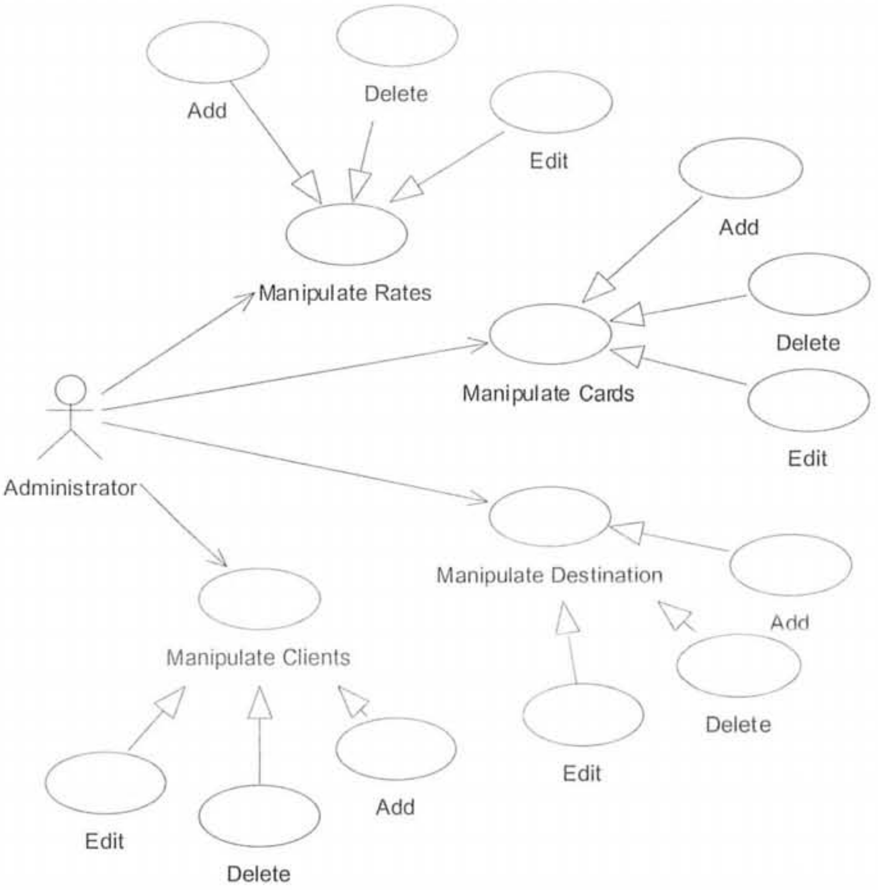
Use-case Diagram of Administrator

Use-Case Relationship Diagram of Client



Use-case Diagram of Client

BASIC DATA MANIPULATION TASKS FOR ADMINISTRATOR



Manipulation Tasks Of Administrator

3.5 Use-Case description

Log-in:

Administrator will login with its personal password to interact with the software.

Change Password:

Administrator can change its password.

Generate Report:

Provides the facility for generating report of monitoring result, according to administrator specifications.

View Report:

Provides the facility for administrator to view the client call result, billing detail in form of report. For the easy viewing this report will be in form of web page.

Generate Cards:

Administrator can generate card series having unique Pin Numbers.He/she can generate card series of fixed amount.

Issue Cards:

Provides the facility of issuing cards after generation. The administrator can only issue cards.

Add New Rates:

Administrator can add new rates of cities and countries according to different prefixes.

Configuring Rates:

System provides the facility of configuring the rates by administrator. He/she can change the city or country rates.

Payment:

In order to Pay the bill, System provide the facility of on-line payment or payment received by the administrator.

Lock/Unlock Clients:

Administrator of the system can lock or unlock the client due to different reasons. For example, if some clients don't pay its bill on specific time then the administrator can lock its status.

View Records:

Administrator and client both can view their records. Client can only view its own record but administrator can view records of registered clients.

Send E-mail:

Administrator can send mails to the clients to remind about payments, any new plan or package.

Generate Free-Bonuses:

Billing software has the facility of generating free bonuses that cab be assigned to clients by the billing administrator.

Generate Bill:

For payment purposes, the billing administrator will generate bills on specified dates. In bill report, there will be the total client amount, total calls duration and total calls charges against its Account Number.

Change Pin Number:

Clients can change their Pin Numbers for simplicity.

3.6 System Design

Design is the process of determining effective and efficient implementation to carry out the function and store the data to fulfill the requirements. During the design the appropriate computers, devices, software services, and the data storage strategies required for system development is selected. The designer's goal is to produce a model or representation of any entity that will later be built. A successful design is the one that can translate the system requirements into the finished product. A good design means that the implementation time complexities are reduced. The design must contain the following characteristics

- The design ensures the accurate translation of customer requirements.
- It should be readable and understandable.
- It should provide complete picture of the software.
- Design forms basis for programming and maintenance.

3.7 Classes Identification

Based on the use cases and the requirements of the system, following classes are recognized

- Login
- Client
- Administrator
- Card
- CallHandler
- Rate
- Payment
- Bonus
- StatisticsReport

3.8 Classes Description

Login

This class will be used for administrator and client authorization.

Client

This class will be used for client registration and information relating clients like client status (Lock/Unlock), client category and more.

Administrator

This class will contain information about administrator registration.

Card

Cards class will be used for holding cards information like card amount, card pincode. This class will also tell us about card status either is valid, expired, using or available.

CallHandler

Call class will be used for call calculation. This class will measure the calling time and also billing the call.

Rate

Rates class will be used to find out the rates of the destinations (city/country).

Payment

Payment class will be used for receive payments either online or through administrator. Client can also pay its bill through checks.

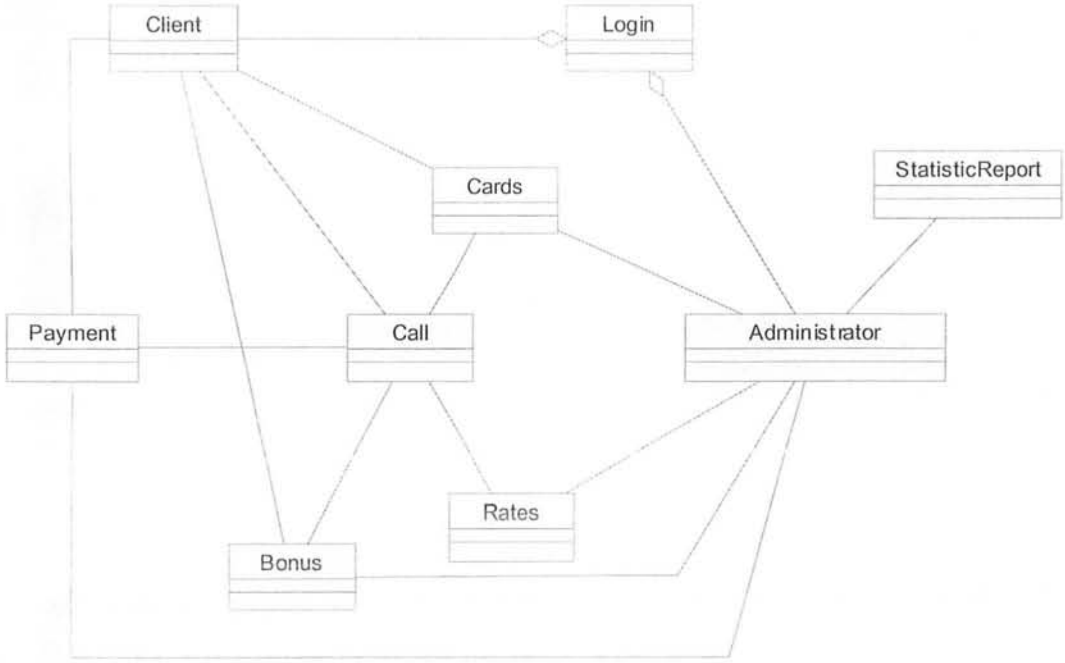
Bonus

This class will be used to generate free bonuses by the administrator and assigned to the clients.

StatisticsReport

This class will use for report generation and report viewing.

3.9 Class Relationship Diagram



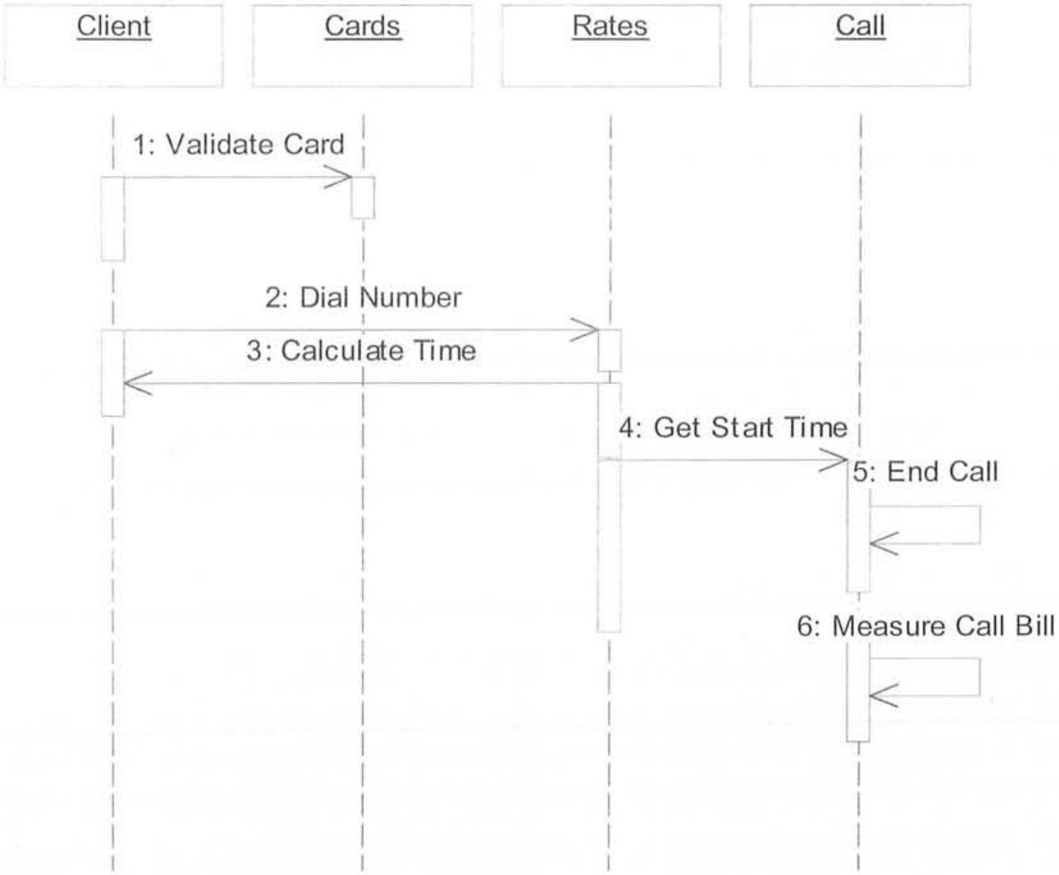
Class Relationship Diagram

3.10 Sequence Diagrams

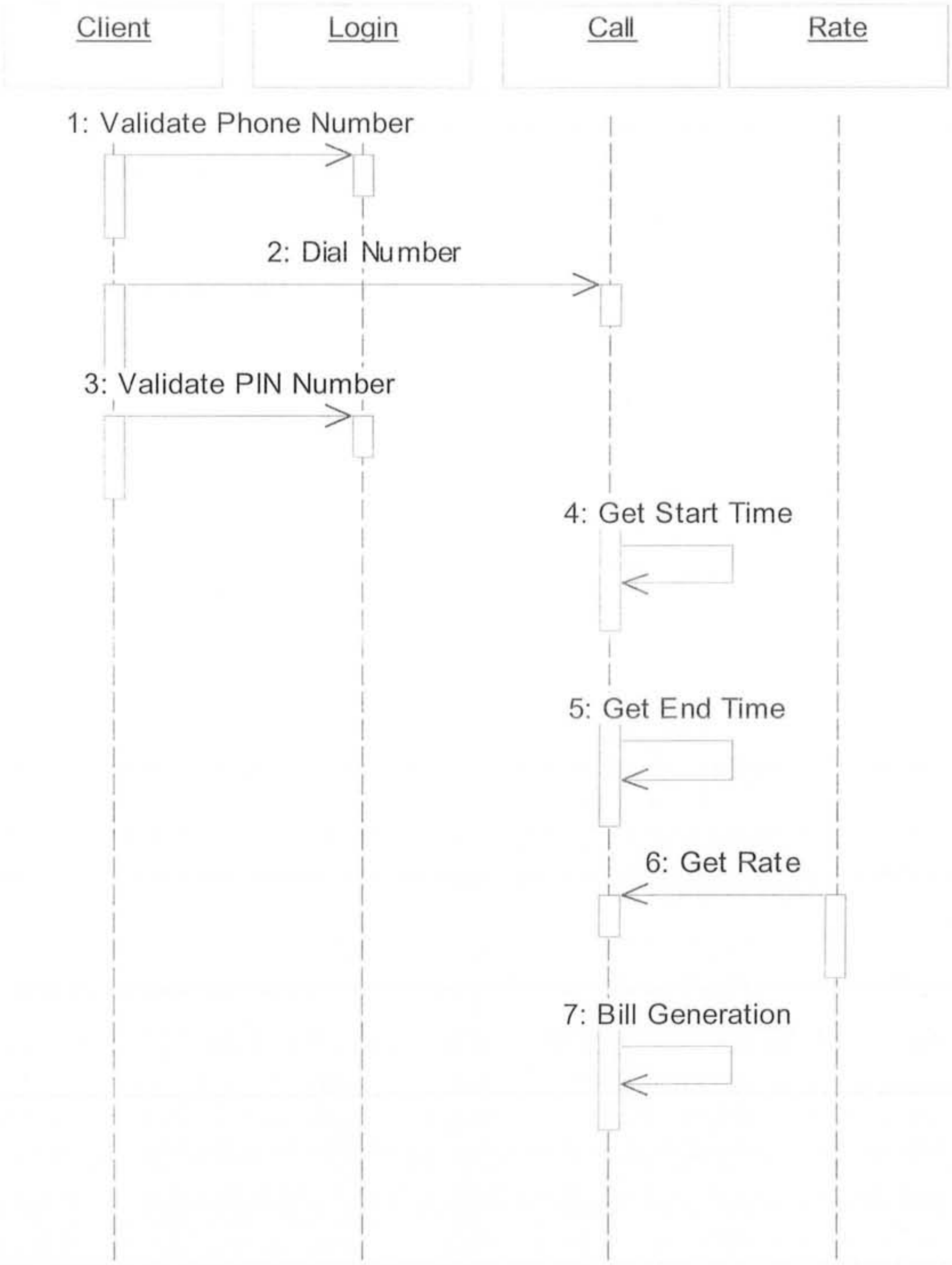
Sequence diagrams and collaboration diagrams are alternate representations of an interaction. A sequence diagram traces the execution of an interaction in time. After Identifying classes and making a class diagram for watching out co-ordination among all classes, lets look at some sequence diagrams for some major functions.

Bill Generation of PIN Client

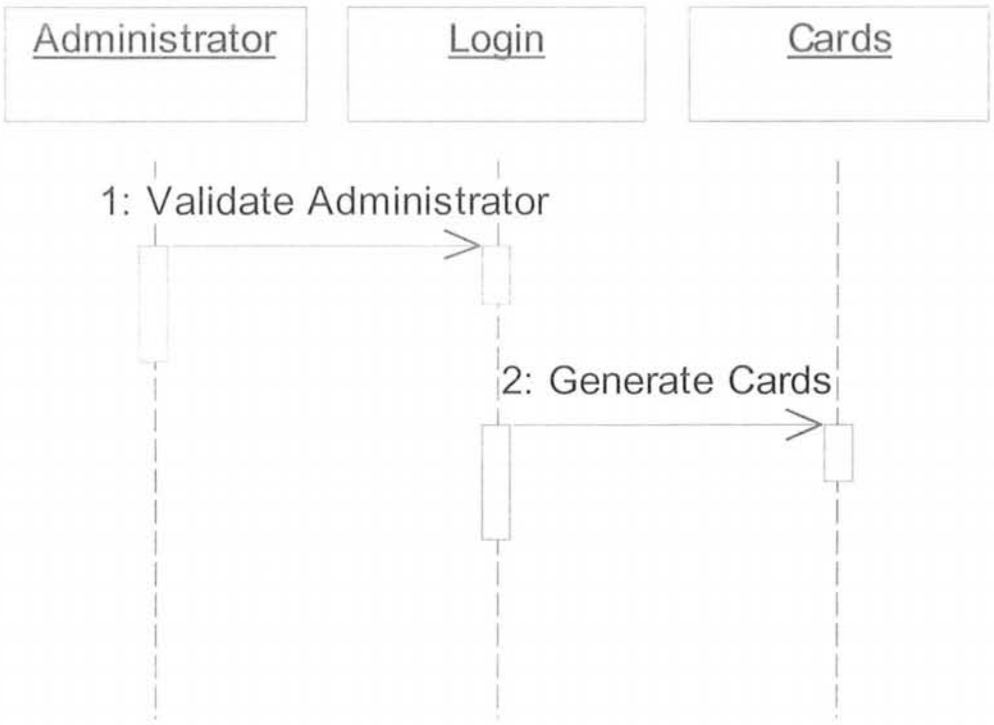
Following is the sequence diagram for the PIN Client Call Establishment.



Bill Generation of Registered Client:



Sequence Diagram Of Card Generation:

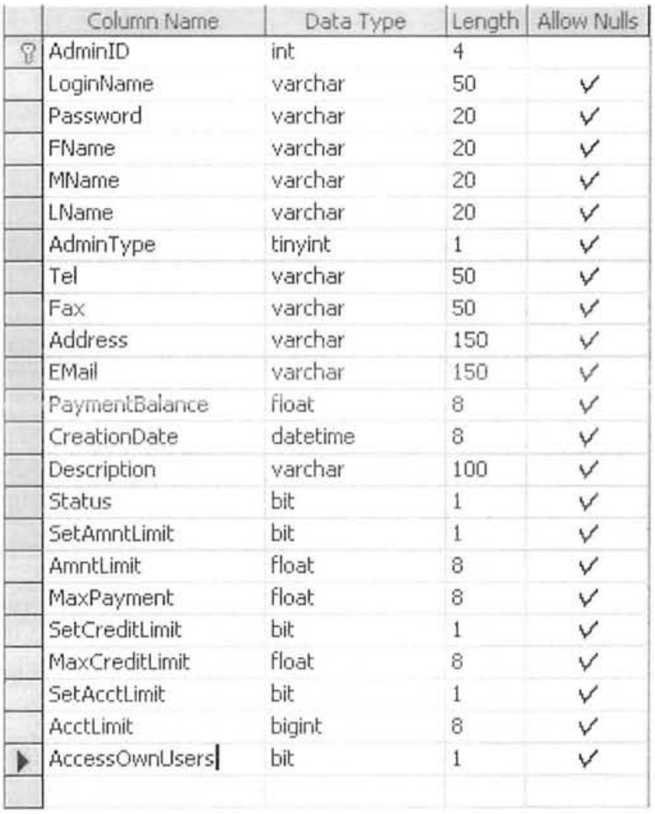


3.11 Database Design

Tables Design

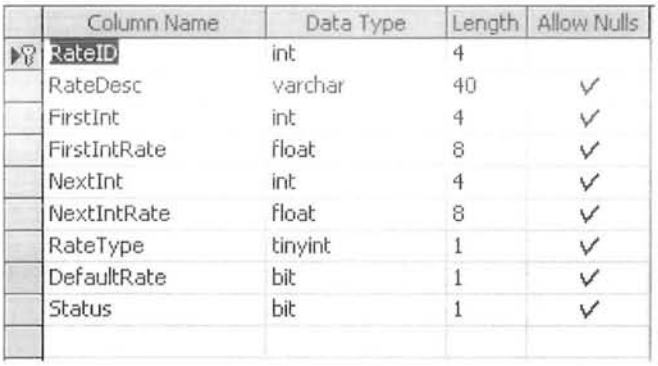
Following are the tables using in the VoIP Billing System:

1.Administrator



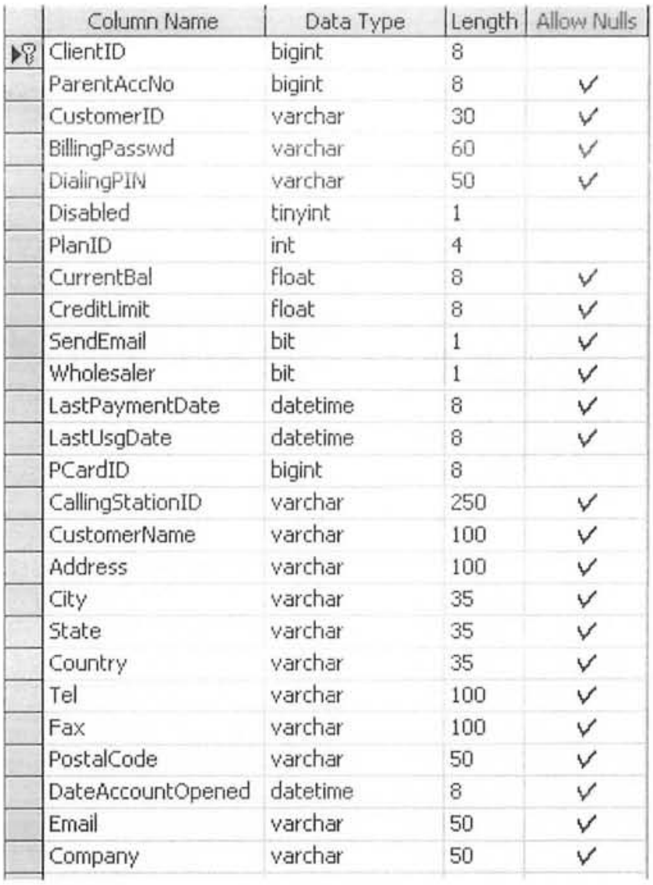
Column Name	Data Type	Length	Allow Nulls
AdminID	int	4	
LoginName	varchar	50	✓
Password	varchar	20	✓
FName	varchar	20	✓
MName	varchar	20	✓
LName	varchar	20	✓
AdminType	tinyint	1	✓
Tel	varchar	50	✓
Fax	varchar	50	✓
Address	varchar	150	✓
EMail	varchar	150	✓
PaymentBalance	float	8	✓
CreationDate	datetime	8	✓
Description	varchar	100	✓
Status	bit	1	✓
SetAmntLimit	bit	1	✓
AmntLimit	float	8	✓
MaxPayment	float	8	✓
SetCreditLimit	bit	1	✓
MaxCreditLimit	float	8	✓
SetAcctLimit	bit	1	✓
AcctLimit	bigint	8	✓
AccessOwnUsers	bit	1	✓

2. Rate



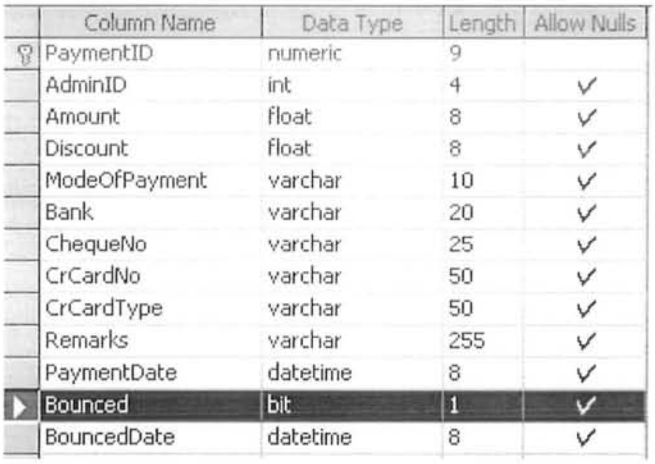
Column Name	Data Type	Length	Allow Nulls
RateID	int	4	
RateDesc	varchar	40	✓
FirstInt	int	4	✓
FirstIntRate	float	8	✓
NextInt	int	4	✓
NextIntRate	float	8	✓
RateType	tinyint	1	✓
DefaultRate	bit	1	✓
Status	bit	1	✓

3. Client



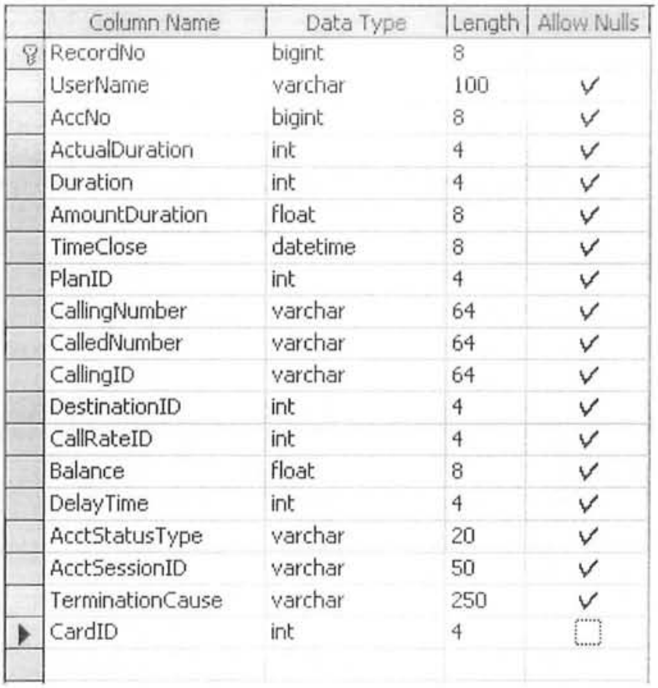
Column Name	Data Type	Length	Allow Nulls
ClientID	bigint	8	
ParentAccNo	bigint	8	✓
CustomerID	varchar	30	✓
BillingPasswd	varchar	60	✓
DialingPIN	varchar	50	✓
Disabled	tinyint	1	
PlanID	int	4	
CurrentBal	float	8	✓
CreditLimit	float	8	✓
SendEmail	bit	1	✓
Wholesaler	bit	1	✓
LastPaymentDate	datetime	8	✓
LastUsgDate	datetime	8	✓
PCardID	bigint	8	
CallingStationID	varchar	250	✓
CustomerName	varchar	100	✓
Address	varchar	100	✓
City	varchar	35	✓
State	varchar	35	✓
Country	varchar	35	✓
Tel	varchar	100	✓
Fax	varchar	100	✓
PostalCode	varchar	50	✓
DateAccountOpened	datetime	8	✓
Email	varchar	50	✓
Company	varchar	50	✓

4. Payment



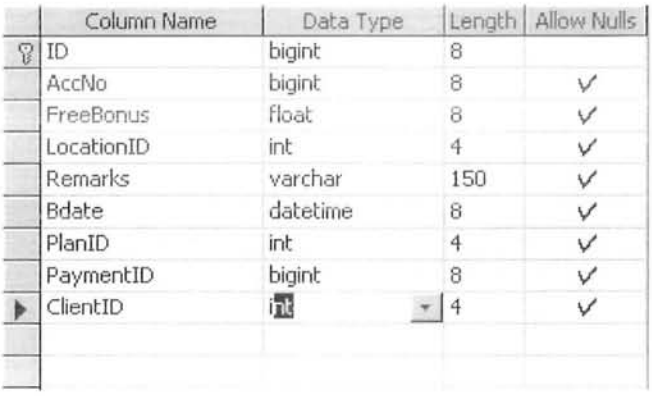
	Column Name	Data Type	Length	Allow Nulls
🔑	PaymentID	numeric	9	
	AdminID	int	4	✓
	Amount	float	8	✓
	Discount	float	8	✓
	ModeOfPayment	varchar	10	✓
	Bank	varchar	20	✓
	ChequeNo	varchar	25	✓
	CrCardNo	varchar	50	✓
	CrCardType	varchar	50	✓
	Remarks	varchar	255	✓
	PaymentDate	datetime	8	✓
▶	Bounced	bit	1	✓
	BouncedDate	datetime	8	✓

6. Call



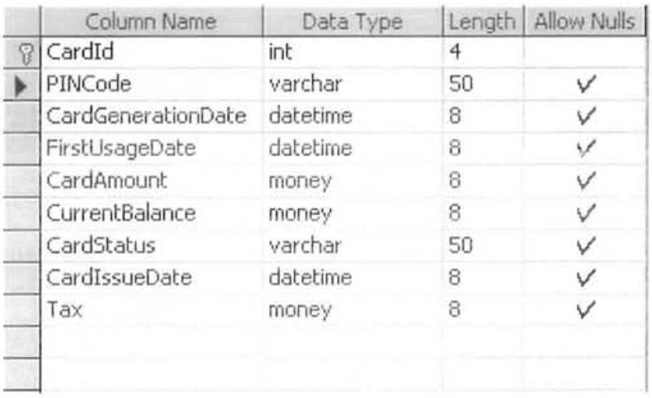
	Column Name	Data Type	Length	Allow Nulls
🔑	RecordNo	bigint	8	
	UserName	varchar	100	✓
	AccNo	bigint	8	✓
	ActualDuration	int	4	✓
	Duration	int	4	✓
	AmountDuration	float	8	✓
	TimeClose	datetime	8	✓
	PlanID	int	4	✓
	CallingNumber	varchar	64	✓
	CalledNumber	varchar	64	✓
	CallingID	varchar	64	✓
	DestinationID	int	4	✓
	CallRateID	int	4	✓
	Balance	float	8	✓
	DelayTime	int	4	✓
	AcctStatusType	varchar	20	✓
	AcctSessionID	varchar	50	✓
	TerminationCause	varchar	250	✓
▶	CardID	int	4	□

7. Bonus



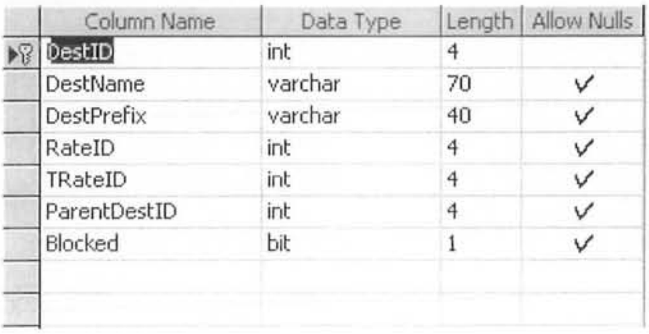
Column Name	Data Type	Length	Allow Nulls
ID	bigint	8	
AccNo	bigint	8	✓
FreeBonus	float	8	✓
LocationID	int	4	✓
Remarks	varchar	150	✓
Bdate	datetime	8	✓
PlanID	int	4	✓
PaymentID	bigint	8	✓
ClientID	int	4	✓

8. Cards



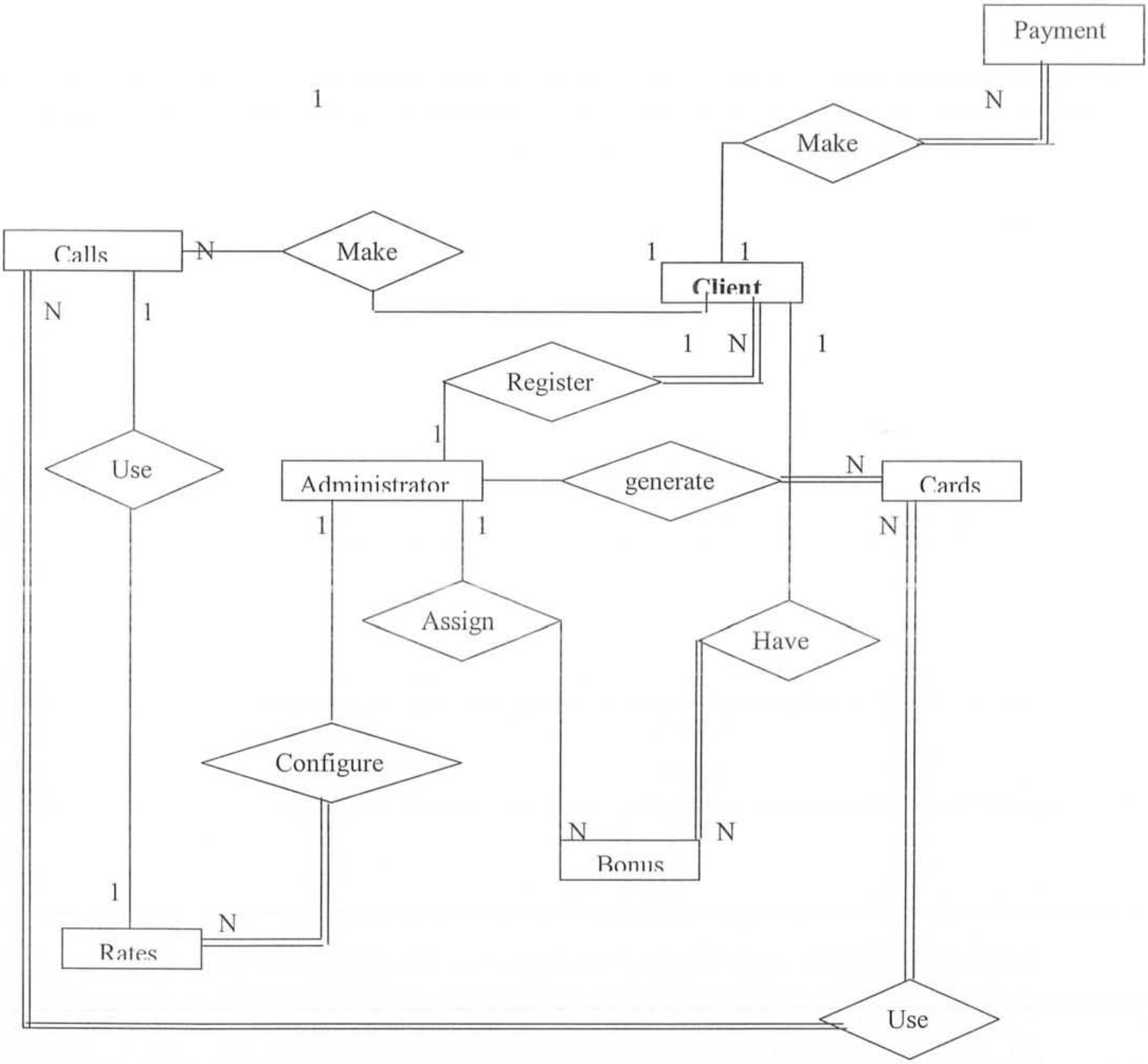
Column Name	Data Type	Length	Allow Nulls
CardId	int	4	
PINCode	varchar	50	✓
CardGenerationDate	datetime	8	✓
FirstUsageDate	datetime	8	✓
CardAmount	money	8	✓
CurrentBalance	money	8	✓
CardStatus	varchar	50	✓
CardIssueDate	datetime	8	✓
Tax	money	8	✓

9.Destination



Column Name	Data Type	Length	Allow Nulls
DestID	int	4	
DestName	varchar	70	✓
DestPrefix	varchar	40	✓
RateID	int	4	✓
TRateID	int	4	✓
ParentDestID	int	4	✓
Blocked	bit	1	✓

3.12 Entity Relationship Diagram



Chapter 4

System Implementation

This chapter gives brief description about the system implementation. What operating system, which language and what technology used to develop that particular system, all detail is in this chapter.

4.1 Introduction

The implementation phase of a project starts after the design. During this phase a design is converted into working software. The software is developed in such a way so that it can meet the requirements and specifications of the users. The implementation of any system is concerned with the tool used in the development work and the component used to implement the system.

4.2 Tool Selection

When the system is implemented, the first thing is to choose appropriate tools for the development of software. The decision of tool selection is very important and crucial. The decision should be wise enough to avoid difficulties in later stages of development. So, the System implementation involves following key features

- Operating System Selection
- Programming Language Selection
- Database Selection

Operating System Selection

An operating system should be chosen that fulfills all the requirements of the software (to be developed) and is easily available to the user. Windows 2000 Professional is selected as development platform as it is organization's standard and later the system will be implemented on the same operating system. So, it is better to use it at development time for easy enhancement and later compatibility.

The system can also be run on Unix/Linux Platform due to java implementation. But as the system is providing web interface to clients for their billing, through that interface they are registered. Therefore, Windows operating system is selected because it provide consistent Graphical User Interface (GUI).

Windows provides *threads* based concurrent execution. All applications that are based on Windows consist of a *process* and one or more *threads*. A process is a single instance of a running application. A thread is the basic unit to which the Windows operating system allocates processor time. A thread can execute any part of the process code, including parts that are currently being executed by another thread.

Windows supports *preemptive multitasking*, which creates the effect of simultaneously executing multiple threads. When a process has more than one thread running, the OS rapidly switches from one thread to another, so that the threads appear to run simultaneously.

Programming Language Selection

The selection of an appropriate language for system development is of vital important. A language should be selected keeping in view both the requirements and the nature of system to be developed. Whatever language is selected it should support the desired programming tasks. JAVA is used to implement the VoIP Billing Software due to the following reasons.

- Java is pure object oriented language.
- It provides programming productively and lets the programmer to focus on his task.
- Its APIs supports the creation of attractive and advanced Graphical User Interface (GUI).
- It provides APIs that supports monitoring of link status and email.
- Because the system is real-time, so for required efficiency Java Servlets (Java Programs) are used.
- Java is Platform Independent and execute on multi platforms.
- It is easily available and is a free ware.
- It supports multi-threading.
- Java is Network oriented.

Database Selection

SQL Server provides a robust development platform for building multi client network applications. Connection from nearly any Microsoft development tool and interface with the underlying database is very easy both to maintain and manage. Another reason for selecting this database is its availability. SQL Server is selected as DBMS to be used on server side to keep the whole record of clients call establishment, clients personal information, rates of cities and countries and card information. In order to get efficiency, Stored Procedures are written.

MySQL don't supports procedures and triggers and due to that reason, there can be delay in billing. That's why, MySQL is not selected.

4.3 HTTP Overview

Most Web clients use the HTTP protocol to communicate with a Web server. HTTP defines the requests that a client can send to a server and responses that the server can send in reply. Each request contains a URL, which is a string that identifies a Web component or a static object such as an HTML page or image file.

The Web server converts an HTTP request to an HTTP request object and delivers it to the Web component identified by the request URL. The Web component fills in an HTTP response object, which the server converts to an HTTP response and sends to the client.

4.4 Java Servlets

Servlets are Java technology's answer to CGI programming. They are programs that run on a Web server and build Web pages. A *servlet* is a Java programming language class used to extend the capabilities of servers that host applications accessed via a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by Web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes.

Chapter 5

System Testing, Evaluation & Future Enhancements

This chapter will describe the brief introduction about system testing, testing strategies including unit testing, integration testing, validation testing and system, test cases, system evaluation including merits and de-merits of the system and future enhancements.

5.1 Introduction

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and coding. The goal of the testing is to increase confidence that the software meets its specifications. It is the process of finding errors, not removing them. Testing a software system typically occupies a large percentage of the time spent developing the complete system. Yet, the amount of scientific effort devoted to this stage of software production is tiny by comparison with those of analysis, specification, design and coding.

A good test case is the one, which has high probability of finding the errors. A successful test is one that uncovers all the errors in the system. Testing is the process of finding errors, it does not ensure the absence of errors.
Testing is often presented as if it is a separate and separable process in the software lifecycle. In fact, testing can (and should) be carried out at each stage in the production and maintenance of a piece of software. It involves two main processes.

- **Validation**

The process involves showing the system to the user and checking whether it fulfills his expectations.

- **Verification**

This process involves testing the system according to requirement specifications. It is checked that whether implementation is according to the specifications.

5.2 Testing Strategy

The classical strategy for testing computer software begins with "testing in the small" and progresses towards "testing in the large".

An overall strategy for system testing is as follow:

- **Unit Testing:** In unit testing different modules of developed system were tested independently. The purpose of testing was to determine that each individual module is functioning properly and to locate logical and coding errors.

- **Integration Testing:** Integration testing is performed after integrating all the modules of the system. Integration testing is performed in order to find, if the integrated modules work correctly or not. After unit testing, integration testing was conducted. Several errors were reported in the integration testing and corrected.
- **Validation Testing:** Validation testing phase comes after all the modules are integrated together. Validation testing is performed to check user variable input to get recognizable output of the system. It is performed to find if the software conforms to the system requirements. Errors encountered while validation testing and were removed.
- **System Testing:** Requirements established as part of software requirements analysis are validated against the software that has been constructed. The software and other system elements are tested as a whole.

5.3 Functional Testing

It is testing of the externally visible functional behavior of the application. It relies on the specification of the system or component. The functionality of the software has been tested according to the following test cases.

5.4 Test Cases

Test case 1:

Purpose: To test whether the GUI of the application views properly or not.

Input: Run the application.

Output: The module was compiled and executed and main page of GUI viewed properly.

Test case 2:

Purpose: To test whether the Client Billing Password of client is valid or not.

Input1: Client ID 4

Password alimalik

Output: Error message shown.

Result: Billing password was not correct.

Input2: Client ID 6

Password aliali

Output: GUI displays the desired page.

Result: Billing password was correct.

Test case 3:

Purpose: *The numbers of generated cards are correct.*

Input: I enters the number of cards and card series and give command to view cards.

Output: A web page of all cards with card series and number of cards was shown.

Result: The number of generated cards was correct.

Test case 4:

Purpose: To test that the rate of city or country is added correctly or not.

Input: I enters the city/country code, name and give command to view the list of rates.

Output: A web page having a list of all rates with their Id's and names was shown.

Result: The rate entered was not correct.

Test case 5:

Purpose: To test that billing application tells correct balance.

Input: I enter the pin code of card in required field.

Output: A massage box was shown with current balance.

Result: System tells the correct balance.

Test case 6:

Purpose: To test that billing application disconnects the call accurately.

Input: I generate an event of disconnect call.

Output: System shows a massage of call disconnected and stops the money conduction.

Confirmation: I checked the balance at call connection time and call disconnected time and calculated the money of call against rate and finally compared that amount into database.

Result: Call was not disconnected properly.

Test case 7:

Purpose: Administrator can register limited client accounts.

Input: I registered the client and before registration checked its limit.

Output: A message was shown that Client registered successfully.

Result: Client registered successfully.

Test case 8:

Purpose: To test that client is locked or not.

Input: I changed the client status as locked with its ID.

Output: A message was shown that Client is locked.

Result: Client is locked successfully.

5.5 Evaluation

Evaluation of the system means to examine what goals are achieved by the system and what are the weaknesses and the deficiencies left behind. Evaluation is the final step in the development of any system. This phase is for the purpose of reviewing whether objectives and functional requirements of the user are fulfilled or not.

Achievements

- Software provides password protected GUI to Administrator to avoid unauthorized access.
- System provides the facility of well-formatted report generation and viewing it.
- System is measuring accurate call time.
- System is very reliable because it is performing real time billing.
- System conducts the user balance from their cards very accurately.
- System is providing billing passwords to registered clients so that they can view only their account information.
- System is providing the facility of card generation and selling cards.
- System is managing both PIN and PINLess users efficiently.
- System provides the facility of changing passwords and pin numbers.
- There will be a limit for an administrator to create new client accounts up to fixed number.
- Clients can be registered On-line as well as registered through Administrator.

Deficiencies

- More efficiency is required, because it is a real time application.

System is satisfying almost all of its major specifications and functionality and it can be concluded that system is satisfying its more then 85% scope.

5.6 Future Enhancements

Organization wants to convert the software into product and wants to launch it in market as one of the best VoIP billing software.

So, future enhancements in brief are

- Multi-lingual support can be implemented to make billing software more useful.
- It will keep the different type of Plans, which will be assigned to the clients.
- It will be integrated with any router and radius server for authentication and authorization purposes.

Other requirements and enhancements with the passage of time will be included in it to prove and launch it as a perfect and the best VoIP billing software for most of windows versions. It will be ensured that software is enough intelligent in almost all conditions.

Appendix

Feasibility Study

Feasibility study is necessary to know that either the development of this software is possible in the current situation with available resources and to identify if any extra resources are required.

- Economical feasibility
- Technical feasibility
- Operational feasibility

Economical feasibility

As it is individual project so no cost involve in searching and hiring the personnel. Organization has already Windows 2000 Professional as its standard operating system for which the system will develop and implement and they already have licensed JAVA and SQL Server. This software is required router for testing which is very expensive, also company has not its own.

Technical feasibility

There exists a powerful concept of web technology in Java language. Java Servlets will be used to develop the system which are java programs runs on the web server. Enough help is available to complete the project successfully plus the technology expected to choose for the development is available and its integration within organizational environment is of no problem because the tool is commonly used over there by the developers.

Operational feasibility

No extra coaching is required to use the software, as it will continue working automatically. This software is operationally feasible because it would provide GUI to interact with the software and user-friendly format to view the database statistics. GUI will provide an easy way to customize the software settings.

Resources

Software Resources required for System development

Windows 2000 Professional
Java (j2sdk 1_4_0)
Microsoft SQL Server 7.0
Rational Rose 98

Hard Resources Used

Pentium II 333 MHz (Compaq)
128 MB RAM
IBM Monitor

Process Model Selection

"Too often, software work follows the first law of bicycling: No matter where are you going, it is uphill and against the wind."

Process Model selection is the one of the important basic decisions for the software development. It is always a difficult task to choose an appropriate Process Model for software development. However a Process Model provides stability, control, and organization to each activity and a timely high quality result. A Process Model is based on the nature of the project and application, the method and tools to be used, and the control and deliverables that are required.

Component Assembly Model is chosen as the major *Process Model* to be followed during the whole process of software development for the under discussion software. As it is not easy to follow a single process model during the whole life cycle of the software development process rather to cope up with the different risks involved at different stages different approaches are needed to be adopted.

This process model very powerfully supports ideas, expected to be followed during software development, and the tool used for development. As mainly the software will be developed in parts, i.e. in modules, so *Spiral Model* provides its support, in each spiral a search for the built-in component or supported APIs in the selected tool, required for coding of a module, if components are not found then these components will be coded and will be made available for later use. Same process, as described above, will be repeated and if some information or material related to the coming modules found, it would be stored. When the scope become clear and the requirements are separately stated they will be not changed until the completion of the project from where we will again change or update it as an enhancement. We are assuming that as project is going to be supervised externally by technical experts. So the requirements will be clear enough before starting design phase and will not change during the development hence it will be a *Waterfall Model.*

In this regard the *Component Assembly Model* is better to adopt, as it is an evolutionary software process model demanding an iterative approach to the creation of software, having the characteristics of both above described models. The object-oriented paradigm emphasizes the creation of classes that encapsulates both the data and the algorithms that are used to manipulate the data. If properly designed and implemented, object oriented classes are reusable across different applications. The use of such model reduces the time involved in the development cycle, reduces the project cost. So, thinking in term of our project the most appropriate and suitable model according to nature of the project is *Component Assembly Model* and will be followed during the whole process of development.