Lab Report of

"TELEPHONE BILLING SYSTEM"

Submitted by

GROUP NO 18
131071056-SHAMA KAMAT
131071041-RUPALI GAWALI
131071045-CHETANA PATEL
141071966-KAVERI KOTHE

T. Y. B. Tech. (COMP) 2015-16

Under the Guidance of

Mrs. P. M. Chawan

Department of Computer Technology

Veermata Jijabai Technological Institute

(An Autonomous institute affiliated to University of Mumbai) Mumbai - 400019.

CERTIFICATE

This is to certify that **SHAMA KAMAT [131071056], RUPALI GAWALI [131071041], CHETANA PATEL [131071045], KAVERI KOTHE [141071966]** student of T.Y.B.Tech (Computer Engineering), Veermata Jijabai Technological Institute (VJTI), Mumbai have successfully completed all assignments of Software Engineering as part of the term work under the guidance of Mrs**. P. M. Chawan.**

Prof. P. M. Chawan

Dept. of Computer Engineering VJTI,

STATEMENT BY CANDIDATE

I wish to state that the work embodied in this special topic titled "Software Project Management" form our own contribution to the work carried out under the guidance of Prof. P. M. Chawan at the Veermata Jijabai Technological Institute (VJTI), Matunga, Mumbai – 19. This work has not been submitted for any other Degree or Diploma of any University / Institute. Wherever references have been made to previous works of others, it has been clearly indicated.

SHAMA KAMAT

RUPALI GAWALI

CHETANA PATEL

KAVERI KOTHE

Table of Contents

1. PROBLEM STATEMENT 8

1.1 PROBLEM ANALYSIS 8
1.2 PROPOSED SYSTEM 9
1.3 SOFTWARE PROCESS MODEL 10
1.3.1 LINEAR SEQUENTIAL MODEL 10

2. PROJECT SCOPE 11

2.1 PRODUCT CHARACTERISTICS 11
2.1.1 FEATURES 11
2.1.2 TOOLS 11
2.1.3 ACCESSIBILITY 11
2.1.4 ACCOUNT 12
2.2 PRODUCT-RELATED DELIVERABLES 12
2.2.1 USER VIEW 12
2.2.2 ADMINISTRATOR VIEW 12

3. RESOURCES 13

3.1 PROJECT RESOURCES 13
3.2 HUMAN RESOURCES 13
3.2.1 PROJECT MANAGER: 13
3.2.2 DEVELOPERS: 13
3.3 SYSTEM REQUIREMENTS 14
3.3.1 DEVELOPMENT ENVIRONMENT: 14
3.3.2 HARDWARE REQUIREMENTS: 15
3.3.3 SOFTWARE REQUIREMENTS: 16

4. SOFTWARE ESTIMATION 17

4.1 ESTIMATION TECHNIQUES
4.1.1 PROBLEM BASED ESTIMATION 17
4.1.2 LOC BASED ESTIMATION: 17
4.1.3 FP BASED ESTIMATION: 18
4.2 ESTIMATION 18
4.2.1 FP BASED ESTIMATION 18
4.2.2 LOC BASED ESTIMATION 20
4.3 COCOMO BASED ESTIMATION 21

5. RISK MANAGEMENT 25

5.1 RISK IDENTIFICATION: 25
5.1.1 GENERIC RISKS: 25
5.1.2 PRODUCT SPECIFIC RISKS: 25
5.2 RISK PROJECTION OR RISK ESTIMATION: 26
5.3 RISK MITIGATION, MONITORING AND MANAGEMENT: 27

6. PROJECT SCHEDULING AND TRACKING
28

6.1 TASK SET FOR THE PROJECT 28
6.1.1 TYPICAL PROJECT TYPES: 28
6.1.2 DEGREE OF RIGOR: 28
6.1.3 TASK SELECTOR VALUE COMPUTATION: 29
6.2 TASK NETWORK: 30
6.2.1 ACTIVITY DIAGRAM: 31

6.3 TRACKING THE SCHEDULE FLOW OF PROJECT SCHEDULE: 31
6.3.1 TIME FLOWCHART: 32
6.3.2 PLANNING TABLE: 32
6.3.3 TRACKING AND SCHEDULING: 33

7. PROJECT PLAN
34
7.1 DEFINITION AND INTRODUCTION OF PROBLEM STATEMENT: 36
7.2 PROJECT SCOPE AND OBJECTIVE: 37
7.3 WORK BREAKDOWN STRUCTURE 37
7.4 BUDGET AND SCHEDULE 37
7.5 RISK ANALYSIS AND MANAGEMENT: 37
7.6 MONITORING AND REPORTING MECHANISMS 37
7.7 PROJECT RESOURCES: 38
7.8 KNOWLEDGE AND SKILLS 38
7.9 STAKEHOLDER INVOLVEMENT 38
7.10 PROJECT ESTIMATION 38
7.10.1 ESTIMATION TECHNIQUES 39
7.10.2 PROBLEM BASED ESTIMATION 39
7.10.3 LOC BASED ESTIMATION 39
7.10,4 FP BASED ESTIMATION 39
7.10.5 PROCESS BASED ESTIMATION 39
7.11 SOFTWARE QUALITY ASSURANCE: 40
7.11.1 QUALITY CONCEPT: 40
7.11.2 SQA: 40
7.12 SOFTWARE CONFIGURATION MANAGEMENT: 40
7.12.1 SCM: 41
7.13 REQUIREMENT ANALYSIS MODELING: 41
7.13.1 DATA MODELING: 41
7.13.2 FUNCTIONAL MODELING: 41
7.13.3 DATA DICTIONARY: 42
7.14 DESIGN AND SOFTWARE TESTING 42
7.15 PROJECT MAINTENANCE AND REUSABILITY: 42

8. SOFTWARE QUALITY ASSURANCE
42
8.1 QUALITY CONCEPTS: 42
8.2 QUALITY CONTROL: 43
8.3 COST OF QUALITY: 43
8.4 SQA SOFTWARE QUALITY: 44
8.5 SQA PLAN 44
8.5.1 INTRODUCTION: 44
8.5.2 MANAGEMENT: 45
8.5.3 DOCUMENTATION 45
8.5.4 STANDARDS, PRACTICES, CONVENTIONS, AND METRICS: 45
8.5.5 REVIEWS AND AUDITS: 45
8.5.6 TEST: 45
8.5.7 PROBLEM REPORTING AND CORRECTIVE ACTION; 46
8.5.8 TOOLS, TECHNIQUES, AND METHODOLOGIES: 46
8.5.9 CODE CONTROL: 46
8.5.10 RECORDS COLLECTION, MAINTENANCE, AND RETENTION: 47
8.5.11 TRAINING: 47
8.5.12 RISK MANAGEMENT: 48

9. SOFTWARE CONFIGURATION MANAGEMENT
49
9.1 CONFIGURATION MANAGEMENT: 50
9.2 SOFTWARE CONFIGURATION MANAGEMENT (SCM) PROCESS: 50

10. REQUIREMENT ANALYSIS MODELING
53
10.1 DATA MODELING: ENTITY RELATIONSHIP DIAGRAM: 53
10.2 FUNCTIONAL MODELING: DATA FLOW DIAGRAM 54

10.3 BEHAViORIAL MODEL 55
10.4 DATA DICTIONARY: 56
10.5 ALGORITHM FOR MEMBER 56

11. DESIGN
58
11.1 TABLES: 62
11.2 ARCHITECTURAL DIAGRAM: 62
11.3 PSPEC 62
11.4 USER INTERFACE 63

12. SOFTWARE TESTING **69**

12.1UNIT TESTING 70
12.2 INTEGRATED TESTING 71
12.3 VALIDATION TESTING 72

13. CONCLUSION
73
13.1 FINAL THOUGHT 73

FUTURE TELEPHONE BILLING SYSTEM

ASSIGNMENT NO 1

PROBLEM STATEMENT

"Telephone Billing System " is developed as per seeing the increasing requirement to speed up the work and incorporate a new work culture. Thus a new software has been proposed to reduce manual work, improving work efficiency, saving time and to provide greater flexibility and user-friendliness as the system previously followed was totally manual one with lots of errors. Telephone Billing System is a command-driven application that helps manage records of the customers and generate their telephone bills.To Develop a software to implement telephone billing system.The software keeps a record of details of customers and calculates their bill amount and generates the bill. Employee(identified by his name and password) has the access to the database.

The employee can:

1.**INSERT** the name of a customer by specifying the various details about him such as name, Id etc.

2.**DELETE** the name of a customer by specifying the customer id.

3.**CALCULATE** the bill .

4.**MODIFY** customer's profile for modifying details.

5.**RETRIEVE** customer's use and plans subscribed by the customer.

6.**EXIT** the database. With every transaction the database is updated and maintained. The software is completely safe.

Existing System:

The main problem within the existing systems is that the work done in them are performed by the individual which takes more time to enter and store the details in which there is more possibility for human made errors. The data retrieval takes more time and sometimes the data may be lost and there are even no chances to find the appropriate statistical calculations of the bills.

The existing System has following drawbacks:

In the existing system all the office works was done manually. The manual work processes was time consuming and hence slow. Following are the main drawbacks of the existing system:

- The existing system is totally manual thus there are chances of error in processing.
- The basic and major drawbacks in the existing system are the speed of retrieval of data from files, which leads to delay.
- Maintenance of voluminous data is very cumbersome and laborious job.
- The manual jobs such as calculation are more error prone.
- There are plenty of chances of duplicity of data and information.
- Updating is very tedious job.
- There is no central database from where one can get different statistical data at one place.

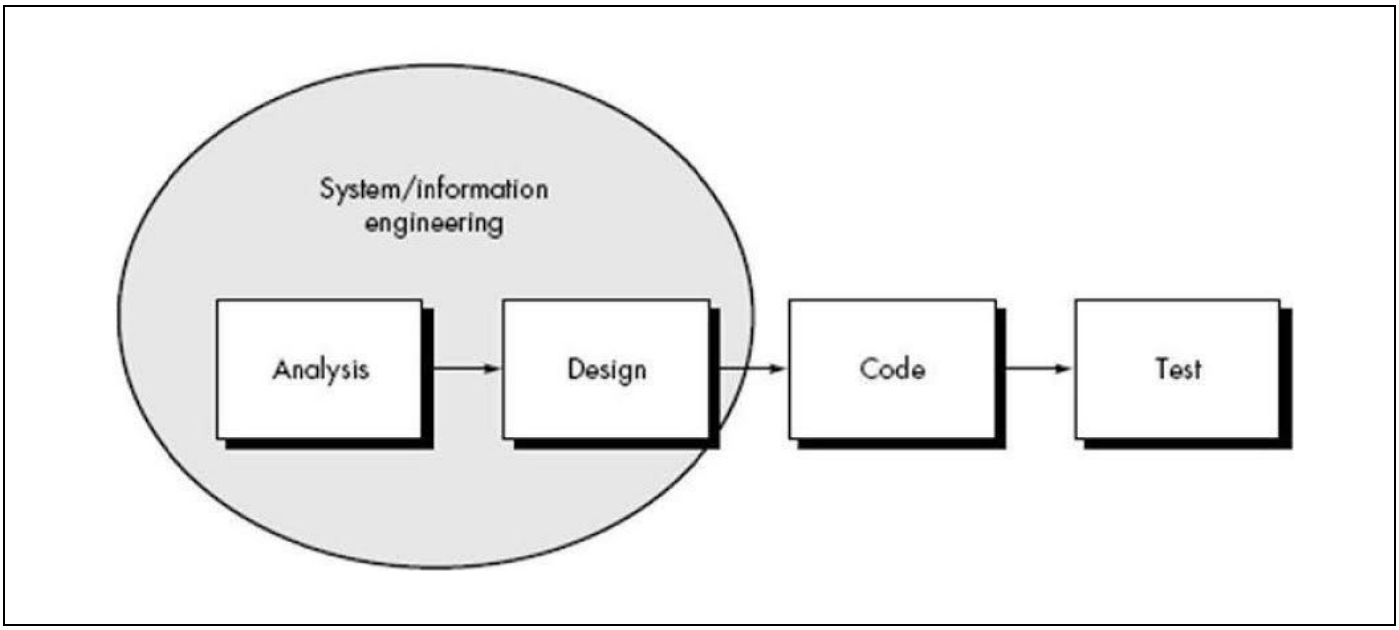
The above facts, figures and drawbacks clearly indicate that there is need for computerization and thus decided to computerize the "TELEPHONE BILLING SYSTEM". Since the existing system was totally manual which has lots of complexities, shortcomings in itself and all the data was being stored in registers, files etc. thus to overcome the limitation of the existing system, the new computerized system was needed, so that information can be provided to the user more quickly, easily and more accurately.

Proposed System:

The proposed system is having many added advantages which is having higher number of economic features for user interface. The system modular will allow the system to pair with its other system for easy data transfer with high control levels in user hands. The billing calculations will be more appropriate and genuine without any errors and by using this application the user bills can be generated easily. One application will support many user connections through which user can access billing process in an easiest way than ever.

The proposed Telephone Billing system software has the following objectives:

- **Enhancement**: The main objective of Telephone Billing System is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with the computer-based system.
- **Automation**: The Telephone Billing System automates each and every activity of the manual system and increases its throughput. Thus the response time of the system is very less and it works very fast.
- **Accuracy**: The Telephone Billing System provides the uses a quick response with very accurate information regarding the users etc. Any details or system in an accurate manner, as and when required.
- **User-Friendly**: The software Telephone Billing System has a very user-friendly interface. Thus the users will feel very easy to work on it. Make the present manual system more interactive, speedy and user friendly.
- **Availability**: The transaction reports of the system can be retried as and when required. Thus, there is no delay in the availability of any information, whatever needed, can be captured very quickly and easily.
- **Maintenance Cos**t: Reduce the cost of maintenance.
- **Security**: Safety is the prime concern for the system as its deals with crucial issues like customer's personal information and tracing.



Project Advantages:

- Simple application with easy to understand and print bill details.
- We can add more features by adding more modules to software.
- Manual paper work process will be reduced and data is secured.

Disadvantages:

- Console based application where data is entered in console as input and results is shown in file and command prompt.

System analysis is the performance management and documentation of activities related to the four life cycle phases of any software namely:

- · The Study Phase
- · The Design Phase
- · The Development Phase
- · The Operators Phase

System analysis is a vast field of study through which system analyst puts his thoughts and searches for the solution of problem. He has to get a clear idea of what he has in hand and what he has to produce. He has to extract the essence of expectations. He has to satisfy the user in the very possible way. System analysis needs and should include the following steps of study:

- Study of current methods, the basic inputs available and output desired.
- The splitting of a variable inputs so as to reduce redundancy and increase consistency.
- Give the idea of key – field (if any) .
- Ideas regarding code generation.

Software Analysis starts with a preliminary analysis and later switches on to a detailed one. During the preliminary analysis the Analyst takes a quick look at what is needed and whether the cost benefits. Detailed analysis studies in depth all the cornered factors, which builds and strengthens the software.

ASSIGNMENT NO 2

SCOPE OF PROJECT

The main objective while implementing the project Telephone Billing System were to minimize the work and at the same time increase the speed of the work done. The present project has been developed to meet the aspirations indicated in the modern age. An attempt has been made through this project to do all work ease & fast. It provide current add, Update, MoveNext, MovePrevious, MoveLast, Find & Delete all facilities to accomplish the desired objectives.

Features:

- Consistent user interface with high economic features built into it.
- User has complete control as it provides and accept only appropriate and valid data.
- User - friendly error messages are provided wherever necessary.
- Addition, deletion, modification of records as when needed.
- Providing connections to new customers.
- Bill generation for customers.
- System design in modular and structured way so as to make the integration with other subsystem easier.

Tools:

The proposed system requires the following tools and technologies for implementation

- Web server - Xampp
- Browser -Google Chrome
- Client Side Coding - HTML
- Client Side Scripting - JavaScript
- Database Server - MySQL
- Server Side Scripting - PHP

View:

- Employee: Employee can interface to perform necessary functions on clients database.
- Customer: Customer can register for any system's plan.

Accounts:

- Employee: Employee can insert, update, bill or delete customer.
- Customer: Customer can register himself or ask for any service to the system.

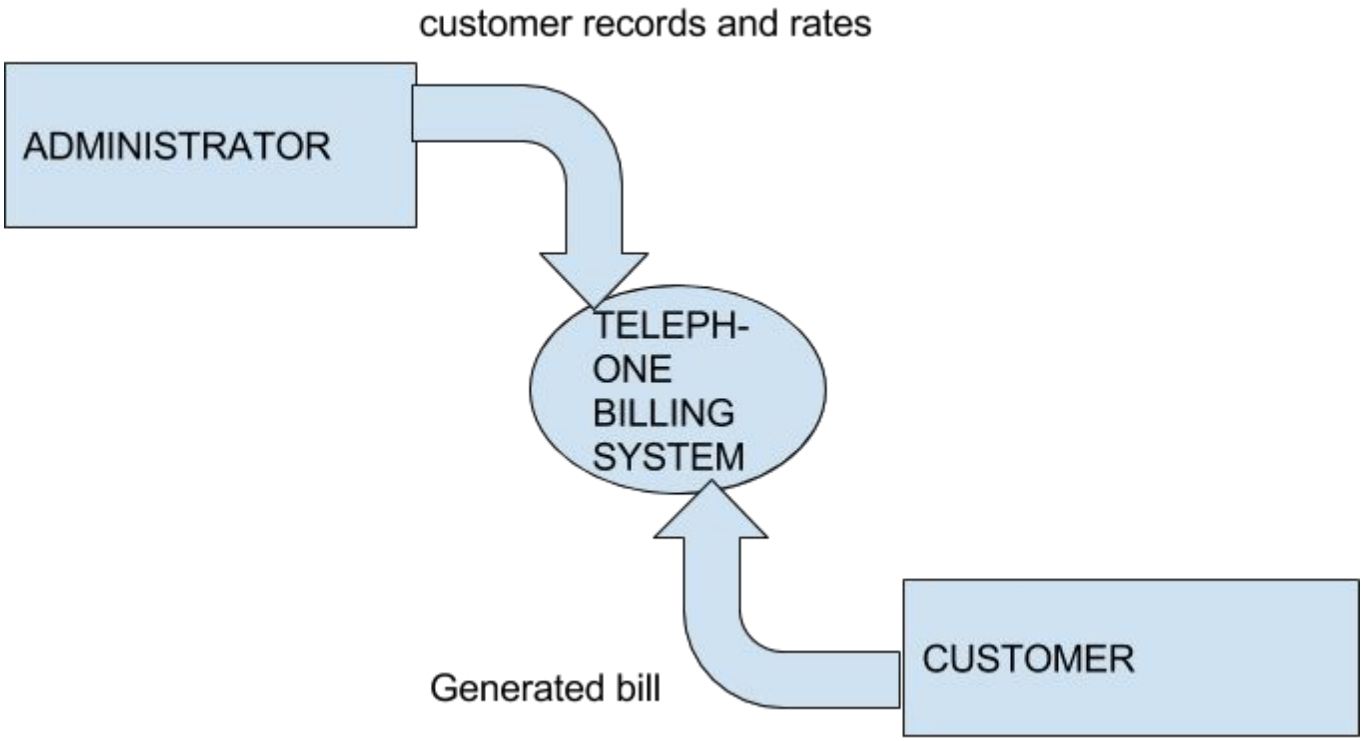
This new system is built with the following **objective**:

- Information retrieval will become easy.
- Maintenance of database as well as overall project will become easy.
- Security measure will be adopted, by maintaining the login of username and the password.
- Data redundancy will be greatly reduced

Main goal of the system:

- To reduce workload of staff.
- To reduce the delay in processing time.
- To reduce the delay in bill generation.
- To provide the user-friendliness in all possible ways.
- To provide greater flexibility.
- Make maintenance changes easy.
- To store data in a centralized location to reduce redundancy and increase consistency.

Data Flow Diagram:
Context Diagram



ASSIGNMENT NO 3

RESOURCES

For information system projects, more specific resources may include system developers, project managers, system analysts, stakeholders, development environments and information.

3.1 Project Resources

These include labour equipment (e.g. workstations)

1. Materials
2. Space
3. Services.
4. Elapsed time can often be reduced by adding more staff
5. Money: used to buy the other resources

RESOURCE ALLOCATION

- Identify the resources needed for each activity
- Identify resource types / individuals are interchangeable within the group.
- Allocate resource types to activities and examine the resource histogram.

3.2 Human Resources

Project Manager:

- A project manager is the person responsible for accomplishing the stated project objectives. They concentrate on resources where there is a possibility that, without planning, they might not be sufficiently available when required.

Developers:

- Their work includes researching, designing, developing, and testing software. A software developer may take part in design, computer programming or software project management..
- In this project, a group of 4 members under the guidance of a project manager will have to complete the task in 3 months. Team members need to complete different task together assigned by the project manager

3.3 System Requirements

Development Environment:

The success of the project also depends upon the quality and availability of the tools which are being used to develop the software. The development requirements are of two types:

Hardware requirements:

- Computer - The minimum requirements to run this system are
    i. CPU (Processor)- 2.0 GHz
    ii. CPU (RAM) - 2GB
    iii. At least 6GB free storage

- Internet connection -At least 1Mbps broadband
- Reliable firewall to secure data in the system

Software requirements:

- Operating system -This system is compatible with Windows XP, Windows Vista & Windows
- Client side Coding - HTML
- Client Side Scripting - JavaScript
- Database Server - MySQL ·
- Server Side Scripting - PHP
- Web Server - Xampp
- Browser - Mozilla Firefox/Google Chrome

Reusable software components

Component –based software emphasizes on the idea of reusability i.e. creation and reuse of software building blocks. They are been classified as follwos:

- Off-the-shelf components
    - Components are from a third party or were developed for a previous project
    - Ready to use; fully validated and documented; virtually no risk

- Full-experience components
  - Components are similar to the software that needs to be built
  - Software team has full experience in the application area of these components
  - Modification of components will incur relatively low risk.
- Partial-experience components
  - Components are related somehow to the software that needs to be built but will require substantial modification
  - Software team has only limited experience in the application area of these components
  - Modifications that are required have a fair degree of risk.
- New components
  - Components must be built from scratch by the software team specifically for the needs of the current project
  - Software team has no practical experience in the application area Software development of components has a high degree of risk

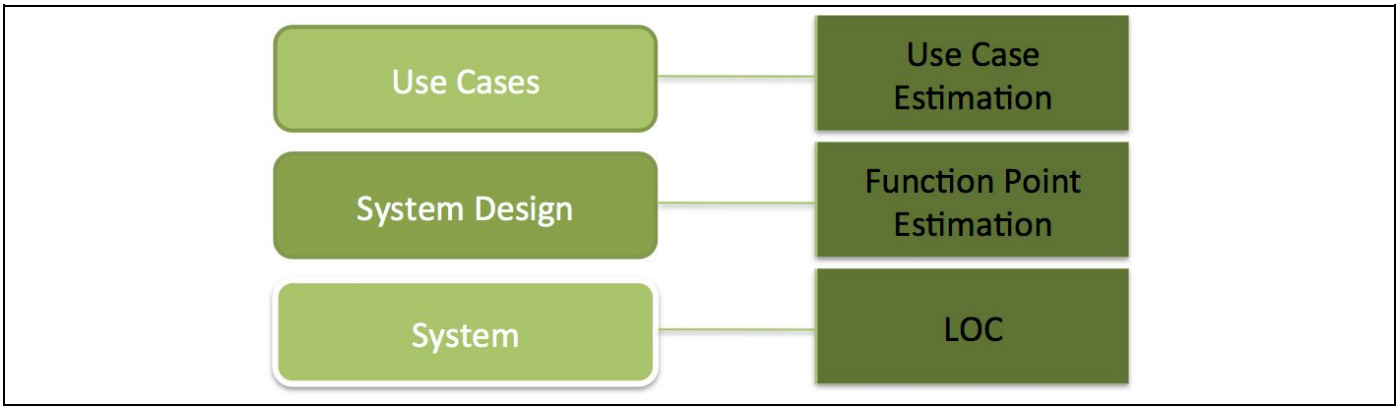
Resource Requirement Specification:
SRS (Software Requirement Specification) is a document that completely describes what the proposed should do, without describing how the software does it.
The purpose of the project is to develop a system which is user friendly, easy to use , maintain and satisfies all the requirements of the user.

PERFORMANCE REQUIREMENT

1) The operation time should be small and the throughput should be high.
2) It should produce timely and accurate result.

ASSIGNMENT NO 4

EFFORT ESTIMATION



4.1 Estimation Techniques

Software project estimation is a form of problem solving, and in most cases, the problem to be solved (i.e., developing a cost and effort estimate for a software project) is too complex to be considered in one piece. For this reason, we decompose the problem, re characterizing it as a set of smaller, more manageable problems. But before an estimate can be made, the scope of the software is understood, to be built and generate an estimate of its "size". If a direct approach is taken, size can be measured in LOC. If an indirect approach is chosen, size is represented as FP.

4.1.1 Telephone Billing System Based Estimation

LOC and FP are basic measures from which productivity metrics can be computed. LOC and FP data are used in two ways during software project estimation:

1. As an estimation variable to "size" each element of the software and
2. As baseline metrics collected from past projects and used in conjunction with estimation variables to develop cost and effort projections

4.1.2 LOC Based Estimation:

The Lines of Code (LOC) method measures the software and the process by which it is being developed. Before an estimate for software is made, it is important and necessary to understand software scope and estimate its size.
The Lines of Code (LOC) method is a direct approach method and requires a higher level of detail by means of decomposition and partitioning.

4.1.3 FP Based Estimation:

Function points(FP) is basic data from which productivity metrics could be computed. The approach is to identify and count a number of unique function types:
external inputs (e.g. file names)
external outputs (e.g. reports, messages)
queries (interactive inputs needing a response)

external files or interfaces (files shared with other software systems)
internal files (invisible outside the system)

Each of these is then individually assessed for complexity and given a weighting value which varies from 3 to 15 i.e from simple to complex.

Unadjusted function points (UFP) is calculated as follows :
The sum of all the occurrences is computed by multiplying each function count with a simple/average/complex complexity and then adding up all the values.
So, FPA expresses the functional size of an information system in a number of function points.
Decomposition for FP-based estimation focuses on information domain values rather than software functions, i.e. each of the information domain characteristics — inputs, outputs, data files, inquiries, and external interfaces — as well as the 14 different complexity adjustment values are estimated. The resultant estimates can then be used to derive a FP value that can be tied to past data and used to generate an estimate.

4.2 Estimation

4.2.1 FP Based Estimation Step 1:

Measurement parameter	**Count**	**Simple**	**Average**	**Complex**	**Function Points**
Number of user inputs	2	3	4	5	6
Number of user outputs	3	4	5	7	12
Number of user enquiries	3	3	4	6	9
Number of files	1	7	10	15	7
Number of external interfaces	1	5	7	10	5
Total Count	-	-	-	-	**39**

Step 2:
We must consider 14 "complexity adjustment values" rated on a scale of 0-5

- No influence -0
- Incidental -1
- Moderate -2
- Average -3
- Significant -4
- Essential -5

FP Estimation Factor	Ratings
1. Reliable backup and recovery	5
2. Data communications	5
3. Distributed processing functions	2
4. Performance critical?	4
5. Will the system run on existing heavily utilized operational environment?	5
6. Online data entry?	5
7. Online data entry over multiple screens or operations	3
8. Are the master files updated online	5
9. Are the inputs, outputs or enquiries complex?	3
10. Is the internal processing complex?	3
11. Is the code designed to be reusable?	4
12. Is conversion and installation included in the design	2
13. Is the system designed for multiple installations in different organizations?	3
14. Is the application designed to facilitate change and ease of use by the user?	3

Total Complexity adjustment factors = 52

Step 3:
Total FP = 39
Sum of all Fi = 52
FP = $39 \times (0.65 + 0.01 \times 52) = 45.63$

The average organizational productivity for systems of this type is 6.5 FP/pm. Based on burdened labor rate of Rs.65, 000 per month, cost per FP is approximately Rs.10000

Based on the FP estimate and historical productivity data, the total estimated project cost is Rs.456300, and the estimated effort is 7 person-months.

4.2.2 LOC Based Estimation

The basis of the Measure Lines Of Code is that program length can be used as a predictor of program characteristics such as effort and ease of maintenance.. LOC method is a direct approach measures the software and the process by which it is being developed. Before an estimate for software is made, it is important and necessary to understand software scope and estimate its size.

For the Telephone Billing Software, the following major functions can be identified - Database Management, User Interface & Functionality.

Following the decomposition technique for LOC, the following table is developed. A range of LOC estimates is developed for each function and after applying the equation,

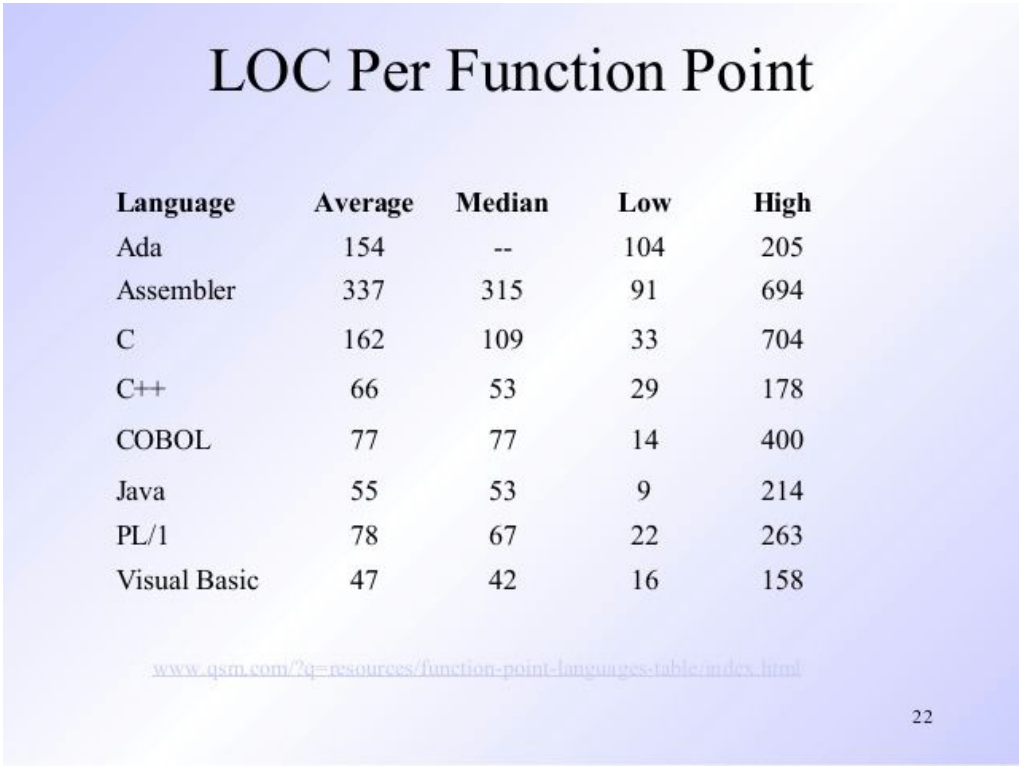
Language	Most-likely	Optimistic	Pessimistic
Java	53	14	134
SQL	21	13	37
HTML	40	14	48

Expected= (optimistic + 4x most-likely +pessimistic)/6 [LOC/FP]

Function	Estimated LOC
Functionality Code	4200
Database Management	1563
User Interface	2590

An estimated 8,353 lines of code are established for the software. A review of historical data indicates that the average organizational average productivity for systems of this type is 600 LOC/pm. Based on a burdened labor rate of Rs.65 000 per month, cost per line of code is approximately Rs.8. Based on LOC

estimate and historical productivity data, total estimated project cost is Rs.8, 12,000 and estimated effort is approximately 13 person-month.



LOC Per Function Point

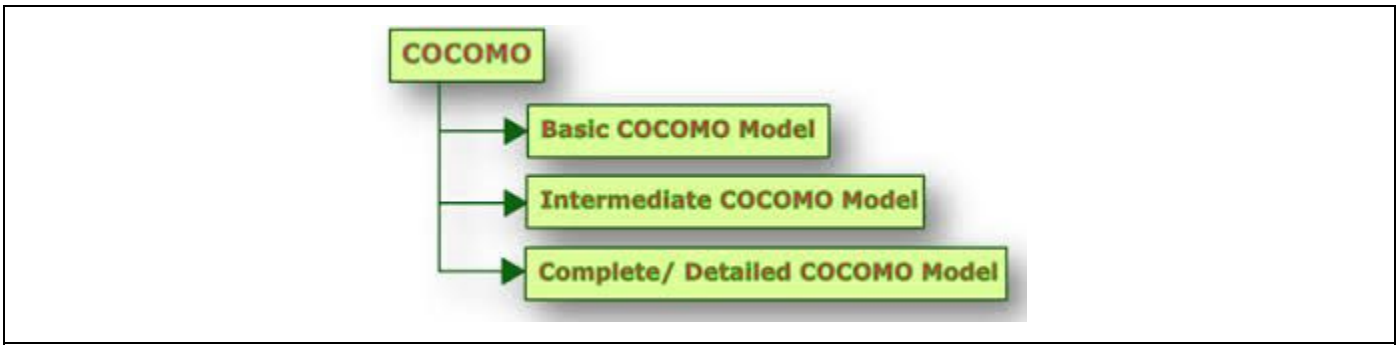
Language	Average	Median	Low	High
Ada	154	--	104	205
Assembler	337	315	91	694
C	162	109	33	704
C++	66	53	29	178
COBOL	77	77	14	400
Java	55	53	9	214
PL/1	78	67	22	263
Visual Basic	47	42	16	158

www.qsm.com/?q=resources/function-point-languages-table/index.html

22

4.3 COCOMO model:

The Constructive Cost **Model** (**COCOMO**) is an algorithmic software cost estimation **model** developed by Barry W. Boehm. The **model** uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics.



Basic COCOMO compute software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC, KLOC).
COCOMO applies to three classes of software projects:

- Organic projects - "small" teams with "good" experience working with "less than rigid" requirements
- Semi-detached projects - "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements
- Embedded projects - developed within a set of "tight" constraints. It is also combination of organic and semi-detached projects.(hardware, software, operational, ...)

The basic COCOMO equations take the form:

Effort Applied (E) = $a_b(KLOC)^{b}_{b}$ **[person-months]**

Development Time (D) = $c_b(\text{Effort Applied})^{d}_{b}$ **[months]**

People required (P) = Effort Applied / Development Time **[count]**

where, **KLOC** is the estimated number of delivered lines (expressed in thousands) of code for project. The coefficients a_b, b_b, c_b and d_b are given in the following table:

Software project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Basic COCOMO is good for quick estimate of software costs. However it does not account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and so on.

Intermediate COCOMO model:

The basic COCOMO model assumes that effort and development time are functions of the product size alone. However, a host of other project parameters besides the product size affect the effort required to develop the product as well as the development time. Therefore, in order to obtain an accurate estimation of the effort and project duration, the effect of all relevant parameters must be taken into account. The intermediate COCOMO model recognizes this fact and refines the initial estimate obtained using the basic COCOMO expressions by using a set of 15 cost drivers (multipliers) based on various attributes of software development. For example, if modern programming practices are used, the initial estimates are scaled downward by multiplication with a cost driver having a value less than 1. If there are stringent reliability requirements on the software product, this initial estimate is scaled upward. Boehm requires the project manager to rate these 15 different parameters for a particular project on a scale of one to three. Then, depending on these ratings, he suggests appropriate cost driver values which should be multiplied with the initial estimate obtained using the basic COCOMO

In general, the cost drivers can be classified as being attributes of the following items:

Product: The characteristics of the product that are considered include the inherent complexity of the product, reliability requirements of the product, etc.

Computer: Characteristics of the computer that are considered include the execution speed required, storage space required etc.

Personnel: The attributes of development personnel that are considered include the experience level of personnel, programming capability, analysis capability, etc.

Development Environment: Development environment attributes capture the development facilities available to the developers. An important parameter that is considered is the sophistication of the automation (CASE) tools used for software development.

Complete COCOMO model:

A major shortcoming of both the basic and intermediate COCOMO models is that they consider a software product as a single homogeneous entity. However, most large systems are made up several smaller sub-systems. These subsystems may have widely different characteristics. For example, some subsystems may be considered as organic type, some semidetached, and some embedded. Not only that the inherent development complexity of the subsystems may be different, but also for some subsystems the reliability requirements may be high, for some the development team might have no previous experience of similar development, and so on. The complete COCOMO model considers these differences in characteristics of the subsystems and estimates the effort and development time as the sum of the estimates for the individual subsystems. The cost of each subsystem is estimated separately. This approach reduces the margin of error in the final estimate. The following development project can be considered as an example application of the complete COCOMO model. A distributed Telephone Billing System (MIS) product for an organization having offices at several places across the country can have the following sub-components:

- Database part
- Graphical User Interface (GUI) part
- Communication part

Of these, the communication part can be considered as embedded software. The database part could be semi-detached software, and the GUI part organic software. The costs for these three components can be estimated separately, and summed up to give the overall cost of the system.

Estimation:

The COCOMO model used is basic and the type of Software Project is Organic Therefore, Telephone Billing Software is purely organic.
Hence the values are:

Effort Applied (E) = a_b (KLOC)^{b_b}= 2.4*(2.0)^1.05 =4.97 person-months

=5 person-months

Development Time (D) = c_b (Effort Applied)^{d_b}= 2.5*(4.97)^0.38 = 4.6 months

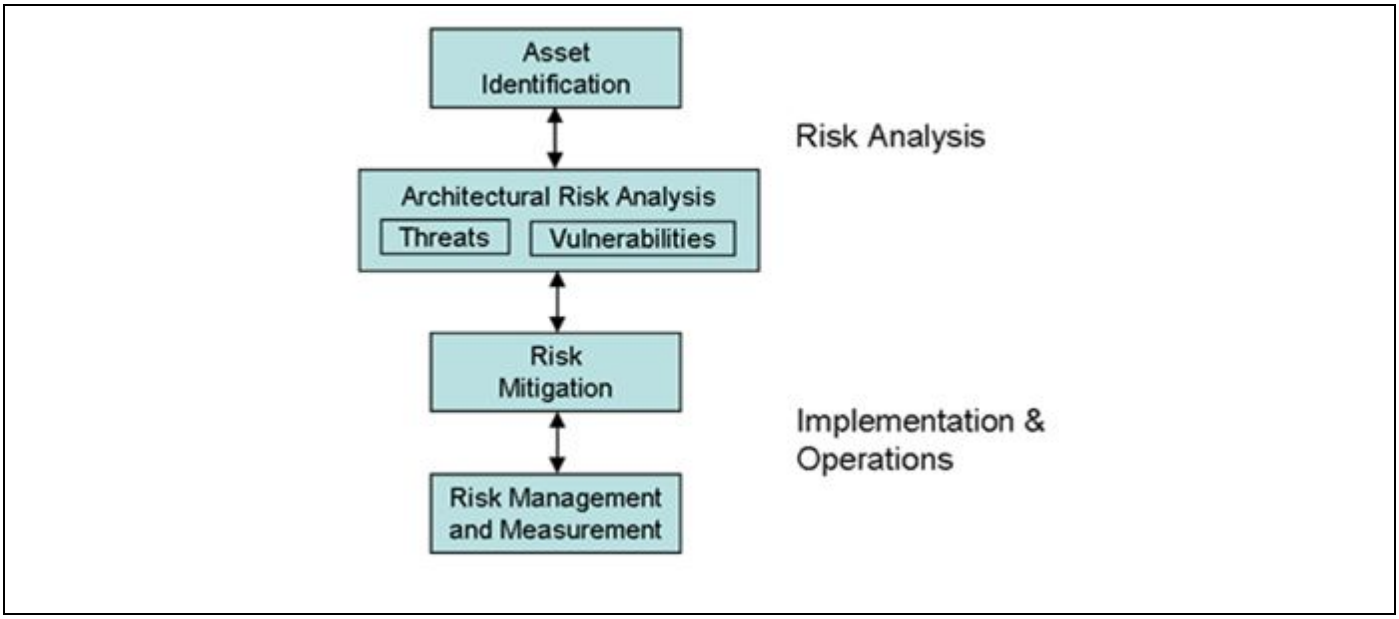
=5 months

People Required (P) = Effort Applied / Development Time= E/D =1.08 count

= 2 count

ASSIGNMENT NO 5

RISK ANALYSIS AND MANAGEMENT



Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. A risk is a potential problem/ it might happen or it may not. But regardless of its outcome we must identify it, assess its probability of occurrence, estimate its impact and establish a contingency plan in order to reduce its impact on the software engineering process. The following are the steps involved in risk analysis:

5.1 Risk Identification:

Risk identification is a systematic attempt to specify threats to the project plan. By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary.

There are two distinct types of risks for each of the categories

1. Generic risks are potential threats to every software product.
2. Product/specific risks – any special characteristics of the product that may threaten the project plan.

One method for identifying risks is to create a risk item checklist. The checklist can be used for risk identification and focuses on some subset of known and predictable risks in the following generic subcategories :Product size, Business Impact, Customer Characteristics, Staff Size and Experience, Process Definition, Development Environment and Technology to be built.

For a “Telephone Billing System” the following are the main risks that could occur:

- Delay in the completion of the project due to changes in the requirements made by the customer in the final stages of the project.
- Failure to identify complex functionalities and time required to develop those functionalities.
- Wrong budget estimation.
- If the server goes down, records could be lost and the users won't be able to access the information.
- Unexpected project scope expansions.
- Too many users could use the system at the same time and the server could be flooded with requests. This traffic will slow down the service tremendously.
- Personnel need extra time to learn unfamiliar software tools, hardware and programming language.
- Unfamiliar areas of the product take more time than expected to design and implement

5.2 Risk Projection or Risk Estimation:

Risk projection, also called risk estimation, attempts to rate each risk in two ways—the likelihood or probability that the risk is real and the consequences of the problems associated with the risk, should it occur. The project planner, along with other managers and technical staff, performs four risk projection activities:

- Establish a scale that reflects the perceived likelihood of a risk.
- Delineate the consequences of the risk,
- Estimate the impact of the risk on the project and the product, and
- Note the overall accuracy of the risk projection so that there will be no misunderstandings .

Generic Risks:

Risks associated with the "newness" of the technology and thus lack of experience. Project budget could get exceeded. Delay in the completion of the project due to changes in the requirements made by the customer in the final stages of project. Project schedule may not be followed. Compromise on the quality of the software due to tightening of deadlines.

Project specific risks:

If the server goes down, records could be lost and the users won't be able to access the information. Too many users could use the system at the same time and the server could be flooded with requests. This traffic will slow down the service tremendously.

RISK ID	RISK	RISK TYPE	PROBABILITY	IMPACT
1	System Failure	Technical risk	50%	1
2	Delivery deadline will be lightened	Business risk	60%	2
3	Funding will be lost	Technical risk	60%	2

Impact Values:

1. Catastrophic
2. Critical
3. Marginal
4. Negligible

5.3 Risk Mitigation, Monitoring and Management:

All of the risk analysis activities presented to this point have a single goal—to assist the project team in developing a strategy for dealing with risk.

An effective strategy must consider three issues:

- Risk avoidance
- Risk monitoring
- Risk management and contingency planning

Risk id:1 (System Failure)
Description :A **system failure** can occur because of a hardware failure or a severe software issue, causing the system to freeze, reboot, or stop functioning altogether.
Mitigation, Monitoring and Management: Ensuring that enough free disk is available for normal operations, and upgrading disk space as required .Backing up of database, either onto external devices or remote servers, to prevent loss of data in case of flooding .Monitoring jobs running on the database.

Risk id:2(Delivery deadline will be lightened)
Description: Deadline allotted for completion o project.
Mitigation, Monitoring and Management: We can avoid delay by following the schedule strictly and the team should consists of experienced professionals .The developer show always communicate with the customer so that last moment changes are avoided.

Risk id:3(Funding will be lost)
Description: The exceeding of budget due to inaccurate estimation of time and money is a potential risk.
Mitigation, Monitoring and Management: Firstly, determine the budget process .regularly review the budget ,both on an informal and formal basis. Whilst the total project budget cost is important, are there any components which look high, low or unusual? What is the budgeted project cost per user (total project budget costs divided by number of users)?

ASSIGNMENT NO 6

PROJECT SCHEDULING AND TRACKING

For the scheduling of project, we take following steps:

(1) Split the project into tasks and estimate time and resources required to complete each task.
(2) Organize tasks concurrently to make optimal use of workforce and time.
(3) Minimize task dependencies to avoid delays caused by one task waiting for the other to complete.
(4) Depends on the manager's intuition and experience.
The process model, framework activities and task set are included here.

6.1 Task set for the project

A task set is the collection of work tasks, milestones, and deliverables. Different set of task proves to be appropriate to different projects. An effective software process is the collection of task sets, each designed to meet the needs of different types of project task sets are designed to accommodate different types of project and different degrees of rigor..

6.1.1 Typical project types:

- Concept development projects
- New application development project
- Application enhancement project
- Application maintenance project
- Reengineering project

6.1.2 Degree of rigor:

- Casual Structured
- Strict
- Quick reaction

The grades are allotted as
0-non incidental
1-minimal
2-low
3-moderate
4-substantial

6.1.3 Task Selector Value Computation:

Adaption Criteria	Grade	Weight	Product
Size of project	3	1.20	3.60
Numbers of potential users	3	1.10	3.30
Mission criticality	3	1.10	3.30
Application longevity	4	1.20	4.80
Stability of requirements	3	1.20	3.60
Ease of communication	4	0.90	3.60
Maturity of applicable technology	3	0.90	2.70
Performance constraints	3	0.80	2.40
Embedded/non embedded characteristics	2	1.20	2.40
Project staffing	3	1.00	3.00
Interoperability	4	1.10	4.40
Reengineering factors	3	1.20	3.60

Task Selector value =3.19

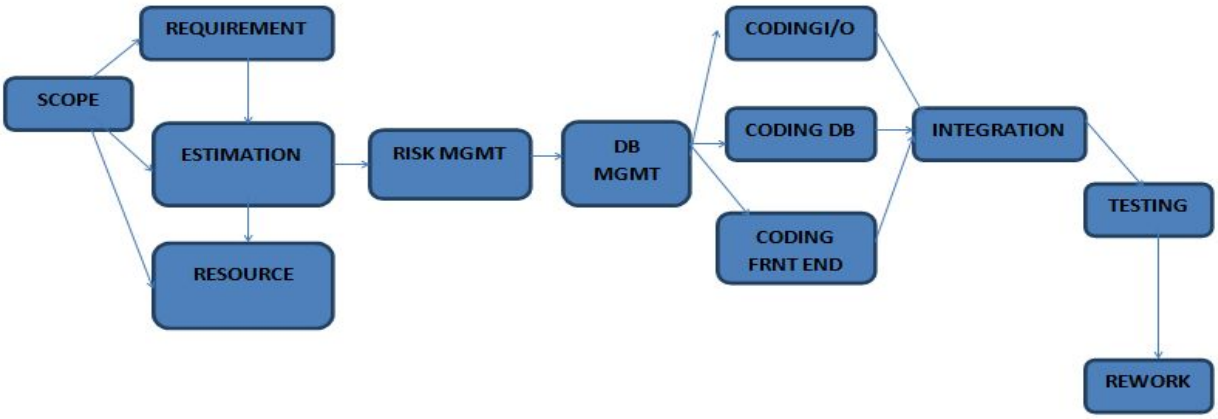
Task set selector value	Degree of rigor
TSS<1.2	Causal
1.0<TSS<3.0	Structured
TSS>>2.4	Strict

6.2 Task Network:

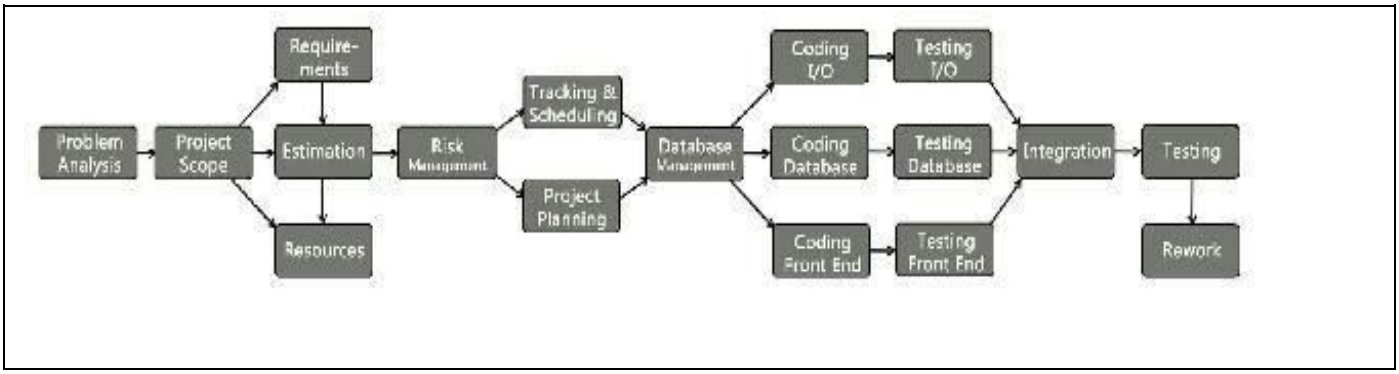
Project tasks and dependencies are noted diagrammatically in task network.
Functional decomposition is as shown.

Task	**Description**
T1	Scoping and overall Design of the software
T2	Database Implementation
T3	Coding database
T4	Designing Interfaces
T5	Coding Interfaces
T6	Test Planning
T7	Integration
T8	Testing
T9	Rework
T10	Final Check

Activity Diagram:



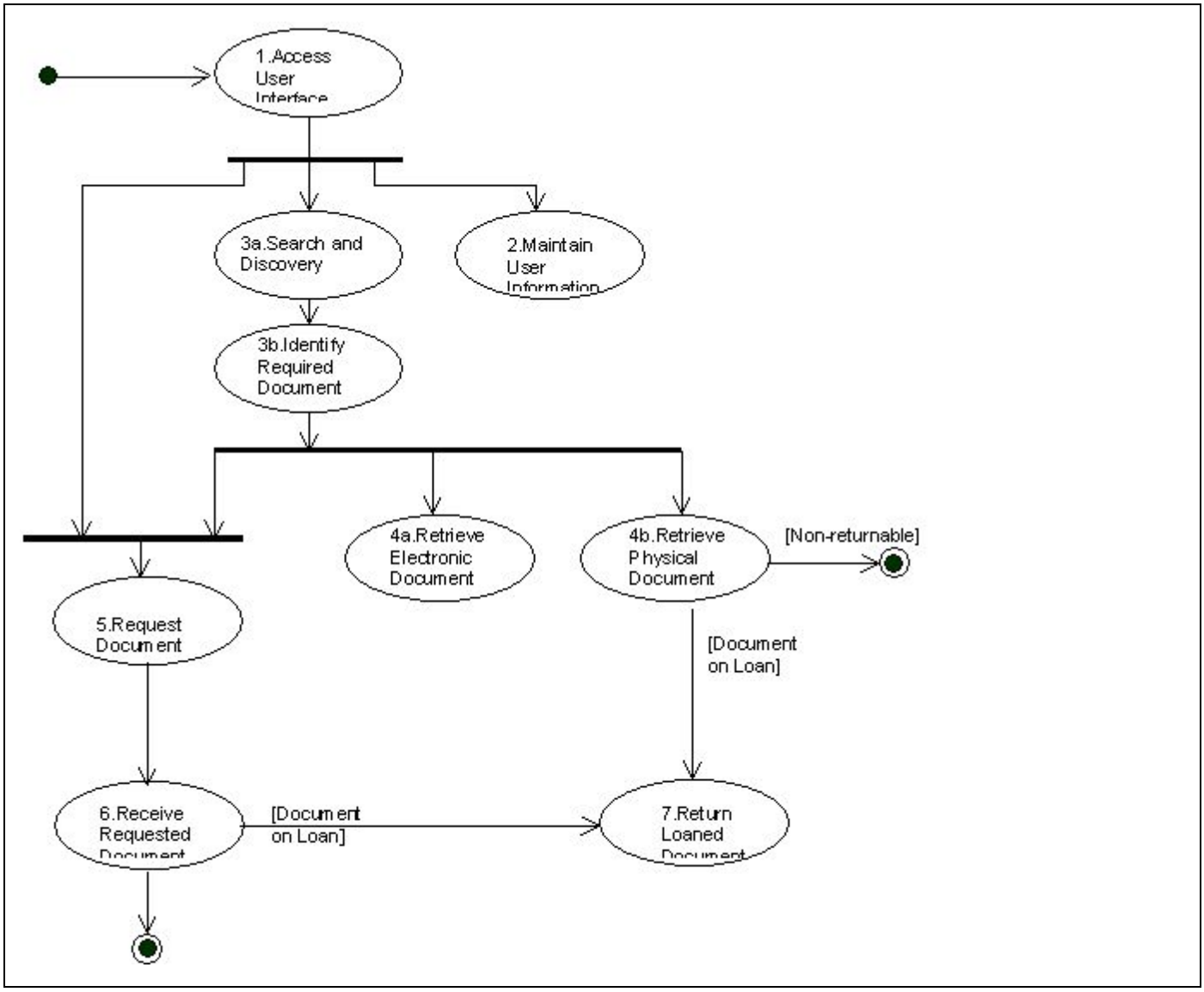
A detailed activity network diagram:



6.3Tracking the Schedule Flow of Project Schedule:

- Establish the project constraint
- Make initial assessments of project parameters
- Define project milestones and deliverables.
- While project has not been completed or cancelled
- Draw up project schedule
- Initiate activities according to schedule
- Wait(for a while)
- Review project
- Progress
- Revise estimates of project parameters
- Update project schedule
- Re/negotiate project constraints and deliverables
- If (problems arise) then- Initialize technical review and revision

Rough flowchart of telephone billing system:



6.3.1 Time Flowchart

	Task NamePlan Start	Planned start	Planned End	Duration
1	cone and Objective, requirement gathering and plannin	17/7/2015	23/7/2015	7
2	Estimation	24/7/2015	30/7/2015	7
3	Resource and arrange necessary tools	31/7/2015	5/8/2015	7
4	Risk analysis and management	6/8/2015	12/8/2015	7
5	Scheduling tasks	13/8/2015	20/8/2015	8
6	Preparing database and coding for it	24/8/2015	6/9/2015	12
7	Code for all modules and front end	7/9/2015	15/9/2015	9
8	Integrating modules	16/9/2015	20/9/2015	5
9	Testing	21/9/2015	25/9/2015	5
10	Reworking	26/9/2015	30/9/2015	5

6.3.2 Planning Table

	Task NamePlan Start	Planned start	Actual Start	Planned End	Actual End
1	Scone and Objective, requirement & plar	17/7/2015	20/7/2015	23/7/2015	23/7/2015
2	Estimation	24/7/2015	25/7/2015	30/7/2015	28/7/2015
3	Resource and arrange necessary tools	31/7/2015	2/7/2015	5/8/2015	5/8/2015
4	Risk analysis and management	6/8/2015	7/8/2015	12/8/2015	10/8/2015
5	Scheduling tasks	13/8/2015	15/8/2015	20/8/2015	19/8/2015
6	Preparing database and coding for it	24/8/2015		6/9/2015	
7	Code for all modules and front end	7/9/2015		15/9/2015	
8	Integrating modules	16/9/2015		20/9/2015	
9	Testing	21/9/2015		25/9/2015	
10	Reworking	26/9/2015		30/9/2015	

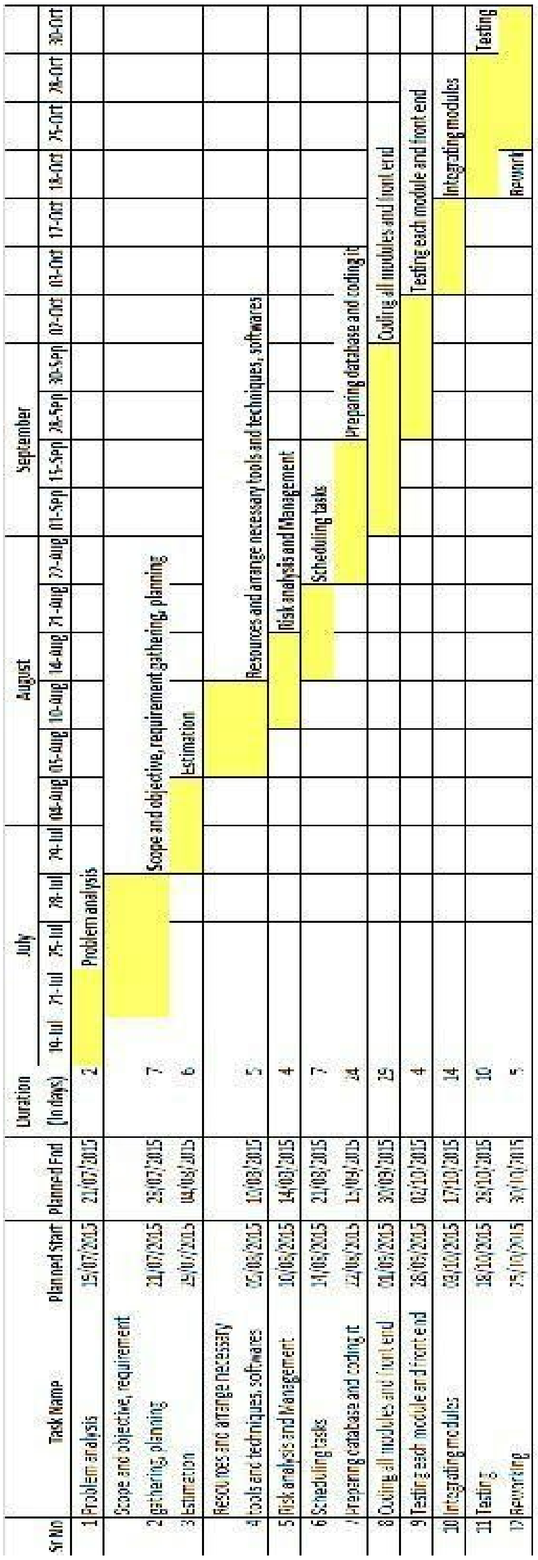
6.3.3 Tracking and Scheduling:

When you are managing a project, one of the most important responsibilities is effectively scheduling tasks and tracking their completion. This helps ensure that your project is completed on time and within budgetary constraints. Different ways of tracking the project are -

1) Conducting different periodic project status meetings where team members report progress and problems.
2) Evaluating results of all reviews throughout the process.

3) Determining whether the formal milestones are accomplished within time as depicted by Gantt chart.
4) Using earned value analysis to assess progress quantitatively.

Timing Diagram:



Sr No	Task Name	Planned Start	Planned End	Duration (In days)
1	Problem analysis	19/07/2015	21/07/2015	2
2	Scope and objective, requirement gathering, planning	21/07/2015	25/07/2015	7
3	Estimation	29/07/2015	04/08/2015	6
4	Resources and arrange necessary tools and techniques, softwares	05/08/2015	10/08/2015	5
5	Risk analysis and Management	10/08/2015	14/08/2015	4
6	Scheduling tasks	14/08/2015	21/08/2015	7
7	Preparing database and coding it	22/08/2015	15/09/2015	24
8	Coding all modules and front end	01/09/2015	30/09/2015	29
9	Testing each module and front end	28/09/2015	02/10/2015	4
10	Integrating modules	03/10/2015	17/10/2015	14
11	Testing	18/10/2015	28/10/2015	10
12	Reworking	25/10/2015	30/10/2015	5

ASSIGNMENT NO 7

PROJECT PLAN

- A project plan should be based on the project requirements and developed estimates, and is a formal, consistent and approved document that provides for the essential project guidelines
- Generally, the purpose of the document is to support management and control of the project execution .
- The plan should cover all phases of the project and should ensure that all involved plans are consistent with each other .

Why do we have a project plan?

There is no single reason for a software engineering project to have a project plan. A project plan usually contains several parts produced in order to help the project team with their project. The main objectives with a project plan are the following :

- - Guide project execution
- - Document project planning assumption
- - Document project planning decisions regarding alternatives chosen
- - Facilitate communication among stakeholders
- - Define key management reviews as to content, extent, and timing
- - Provide a baseline for progress measurement and project control

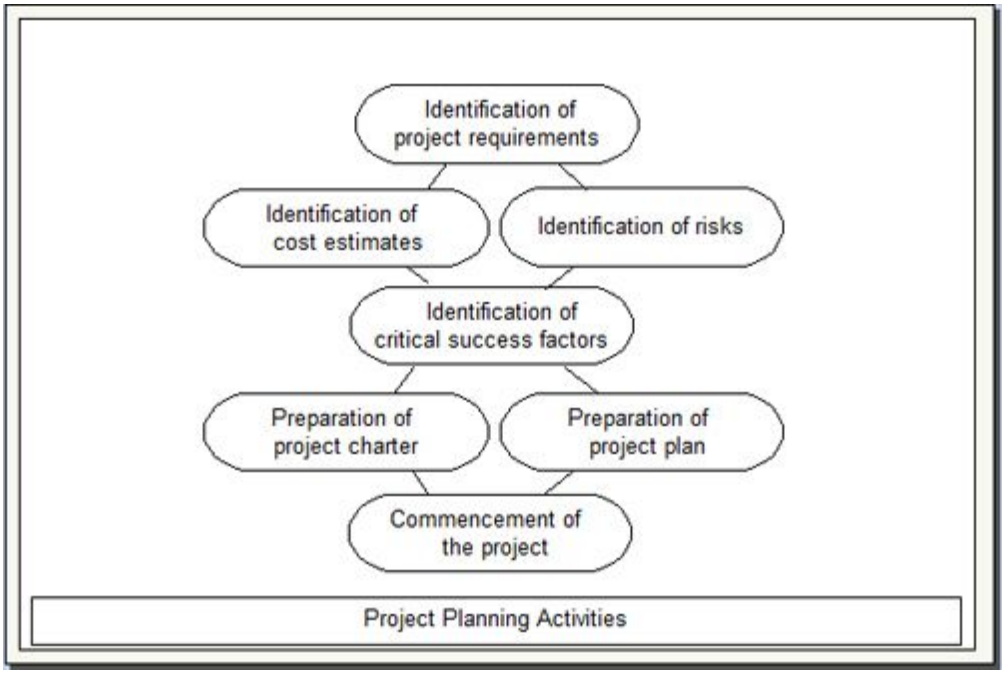
The project plan is generally developed in the initial phase of the project and needs to be reviewed and agreed upon by all concerned persons. However, the plan is expected to change over time and is updated each time the actual progress differs from the plan or when project conditions changes, which require new approaches . A carefully prepared project plan if properly followed and committed to, should lead to a successful project and eliminate many of the pitfalls inherent in the project management process . It provides leadership vision and facilitates for management to utilize available resources efficiently.

Different types of project plans:

In addition to the main software project plan, different types of specific plans may be developed to support the main plan in different areas.

Example of such plans may be the following :

- - Quality plan - Includes the quality procedures and standards that concern the project.
- - Validation plan - Covers approaches, resources and schedule involved in the system validation.
- - Configuration management plan – Consists of the configuration management procedures and structures to support the project.
- - Maintenance plan - Predicts the maintenance requirements, maintenance costs and the effort required.
- - Staff development plan - Includes how available skills and experience will be developed.



An outline of the plan is presented below:

Project planning process:

1. A statement of work (SOW) describes all work that will be produced and a list of all people who will perform that work.

2. A resource list contains the list of all resources that will be needed for the product and their availability.

3. A work breakdown structure and set of estimates.

4. We framed a project schedule and task network to plan the work considering the given deadlines.

5. A risk plan that identifies any risks that might be encountered and indicates how those risks would be handled should they occur.

The outline of the project is as presented below:

7.1 Definition and Introduction of Problem Statement:

A problem statement is a clear concise description of the issue that needs to be addressed by a problem solving team. In project management, the problem statement is part of the project charter and defines what the problem is so that they the project team and stakeholder can focus their attention on solving the problem. It is important to have a good problem statement before starting eliciting requirements for a solution. Problem statement for this project is Telephone Billing System.

7.2 Project Scope and objective:

The scope and objectives of the project are generally set by extracted requirements. The scope is a statement defining the project and its deliverables and should clearly and concisely state project information such as, what it is, what it does, how much it will cost, and when it will be delivered . The project scope has strong relationships to the project schedule and involved resources. Thus, modifications of the project may also affect the project scope.

Scope is defined by answering the following questions:

1) Context
2) Information Objectives
3) Function and Performance

7.3 Work Breakdown Structure (WBS) -

A WBS is a deliverable-oriented grouping of project components that organizes and defines the total project scope . Thus, the WBS divides the total scope into major work packages, which are 12 further subdivided into manageable work items to be accomplished in order to finish the project

7.4 Budget and Schedule -

The budget and schedule of the project are based on established estimates . Schedule development implies to determine start and finish dates for concerned project activities, which for example may be performed through simulations or mathematical analysis. The schedule may be presented by Gantt charts, milestone charts, etc. and may be supplemented by supporting detail documents that include identified assumptions and constraints . Cost budgeting implies to allocate the overall cost estimates to individual work items. The budget should be based on and supported by the WBS, project schedule, and the cost estimates

7.5 Risks -

Planning for project risks should address issues that could jeopardize accomplishment of critical project objectives . The planning process involves identifying project risks, quantifying the risks, and developing risk responses.

7.6 Monitoring and Reporting Mechanisms –

The purpose of monitoring mechanisms are to provide an understanding of the project progress in order to take appropriate corrective actions if project performance deviates from established plans. Monitoring involves monitoring actual values of planning parameters, such as cost, effort, and schedule. These values are compared to the estimates and possible deviations are identified . Reporting mechanisms are concerned with collecting and disseminate performance information in order to provide involved stakeholders with status, progress and information about how resources are used to fulfil established objectives

7.7 Resources - The resources and quantities required in order to carry out the project should be identified and described . Project resources come in various forms such as, personnel, funds, equipment, facilities, material, information etc. and the selection of these resources should be based on the established estimates . Establishing resource requirements allow for several benefits such as, identification of resource shortage, identification of feasibility problems due to resource conflicts, etc.

7.8 Knowledge & Skills - Planning for knowledge and skills involves both training of project team and acquisition of knowledge from external sources. The knowledge and skills required to execute the project should be identified and the currently available knowledge and skills should be assessed . With this information available, the deficiency of knowledge and skills is identified and mechanisms for providing this knowledge and skills are selected .

7.9 Stakeholder Involvement -

Stakeholders involved in the project should be identified and their functions requiring representation in the project should be defined. Furthermore, the level of interaction and the relevance of each involvement should be described . An appropriate technique to handle this effort is to develop a two-dimensional matrix with stakeholders along one axis and project activities along the other axis . Obviously, a project plan may take various forms depending on the needs and purpose. The project plan is obviously developed when the developer has reached an agreement with the customer. Prior to such an agreement, the supplier must be certain that they are capable of undertaking a project. If the supplier doesn't possess the 13 knowledge, skills or time to undertake the project, it's a waste of time to initiate negotiations with the customer. To find out whether this is the case or not, effort and time estimates must be established. When this is settled and negotiations are initiated, the supplier must come up with a reasonable price for the product. The price must be compatible, cover the development costs and furthermore generate a desirable profit. In order to achieve this balance the project costs must be estimated. With all the estimates at hand, it’s possible to establish the project scope, a schedule and budget, resources required, and knowledge and skills required.

7.10 Project Estimation:

Project estimation implies to predict the effort required to successfully execute the project. Lack of project estimates makes the project boundaries quite vague. Estimates serve as a compass, navigating the project team throughout the project lifecycle. Making estimations are not difficult, but to establish accurate and realistic estimates is one of the most challenging and important activity in software development

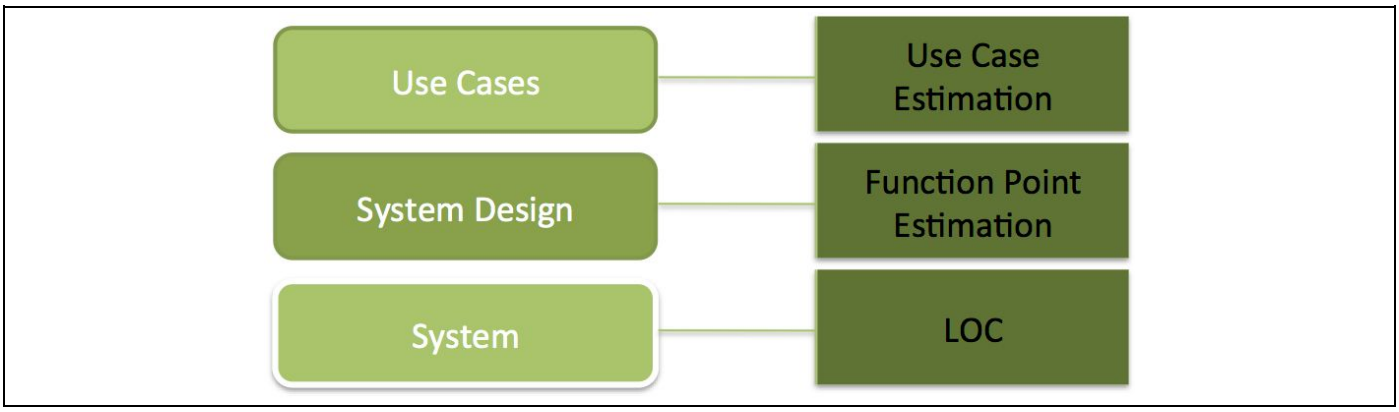
Why do we estimate?

There are several reasons to establish project estimates. Project estimates, generally, provide the project staff, management and other stakeholders with basic guidelines and settles the project scope. As mentioned above, estimates are also an important instrument to verify that the supplier is capable of undertaking the project and moreover provides for a price indication. Thus, accurate estimates are critical to both the software supplier and their customers. Underestimating the costs may lead to that the supplier may be forced to exceed the budget, abandon the time plan and decrease functionality and quality of the software However,

overestimating may result in too many resources allocated to the project or that the supplier loses the contract in the bidding phase, because of a too high price .

Importance:

- Estimates can help to classify and prioritize development projects with respect to an overall business plan.
- Estimates facilitate to allocate resources to the project and how these resources will be used.
- Estimates make it easier to manage and control projects when resources are linked to real needs



7.10.1 Estimation Techniques: Could be direct or indirect**.**

7.10.2 Problem Based Estimation:
The project planner begins with a bounded statement of software scope and from this statement attempts to decompose software into problem functions that can each be estimated individually. LOC or FP (the estimation variable) is then estimated for each function. Alternatively, the planner may choose another component for sizing such as classes or objects, changes, or business processes affected.

7.10.3 LOC Based Estimation:
A formal method to measure size by counting number of lines of Code. When LOC is used as the estimation variable, decomposition is absolutely essential. The greater the degree of partitioning, the more likely reasonably accurate estimates of LOC can be developed.

7.10.4 FP Based Estimation: Function Point Analysis (FPA) FPA is method to measure the functional size of an information
system. The functional size reflects the amount of functionality that is relevant to and recognized by the user in the business. It is independent of the technology used to implement the system.

7.10.5 Process Based Estimation:
The process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated.

7.11 Software Quality Assurance:

Software quality assurance (SQA) is an umbrella activity that is applied throughout the software process.

SQA encompasses:

(1) A quality management approach
(2) Effective software engineering technology (methods and tools),
(3) Formal technical reviews that are applied throughout the software process,
(4) A multi testing strategy,
(5) Control of software documentation and the changes made to it,
(6)A procedure to ensure compliance with software development standards (when applicable), and
(7) Measurement and reporting mechanisms

7.11.1 Quality Concept:
Quality is a characteristic or attribute of something which encompasses requirements and quality of conformance that focuses on implementation.

7.11.2 SQA:
Stands for Software Quality Assurance. Quality assurance consists of the auditing and reporting functions of management.

7.12 Software Configuration Management:
Software configuration management (SCM) is an umbrella activity that is applied throughout the software process. Because change can occur at anytime, SCM activities are developed to (1) identify change, (2) control change, (3) ensure that change is being properly implemented, and (4) report changes to others who may have an interest.

Configuration Management Process

SCM:

Software configuration management (SCM) is a set of activities designed to control change by identifying the work products that are likely to change, establishing relationships among them, defining mechanisms for managing different versions of these work products, controlling the changes imposed, and auditing and reporting on the changes made.

7.13 Requirement Analysis Modeling:

7.13.1 Data Modeling:

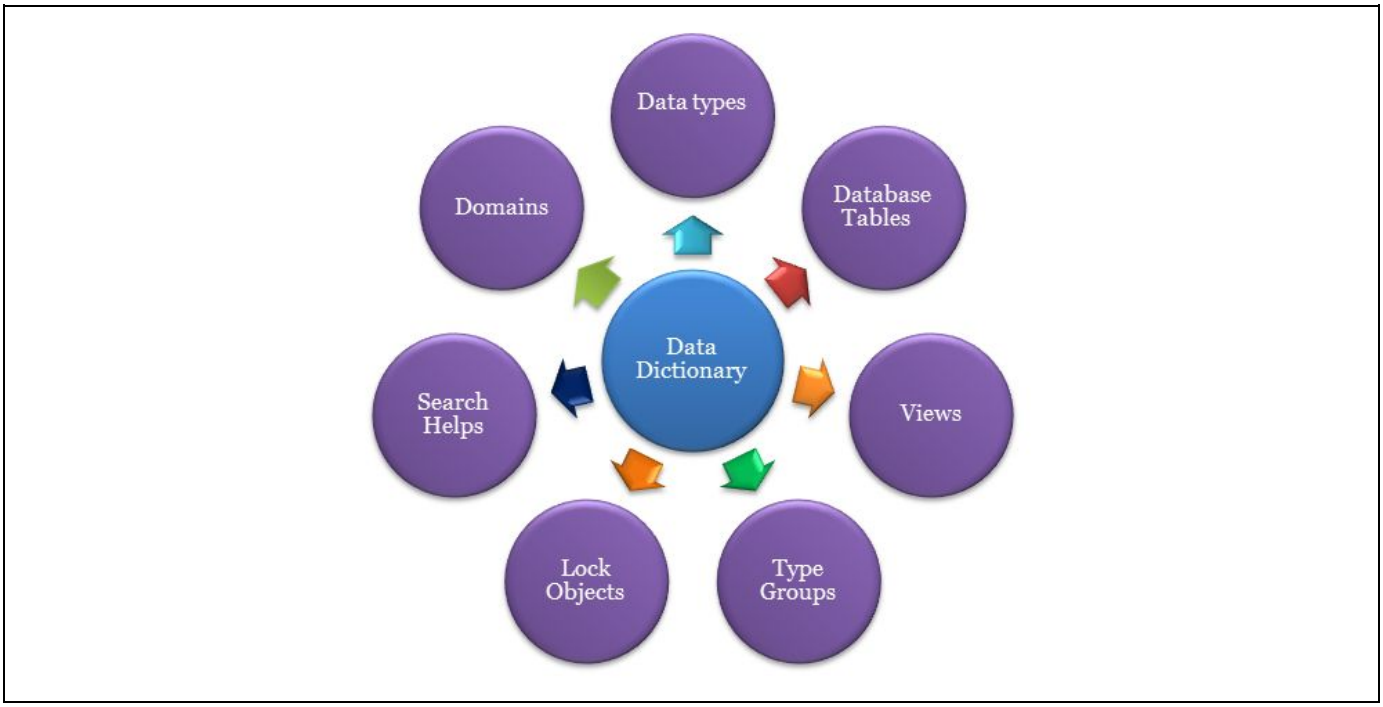
ER Diagram: An entity relationship diagram is an excellent tool for planning and designing a database. The entity relationship model starts with the entities, data normalization starts with the attributes and the two tools tend to verify each other. The entity relationship models entities, attributes and relationships map smoothly to a physical database.

7.13.2 Functional Modeling:

Data Flow Diagram: DFDs are a preliminary step used to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing. A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

7.13.3 Data Dictionary:

A data dictionary is a collection of descriptions of the data objects or items in the data model.



7.14 Design and test:

Designing is a constructive task. Software should be designed in a manner that uses debugging techniques. That is, software should be capable of diagnosing certain classes of errors. In addition, the design should accommodate automated testing and regression testing. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation. In order to find the highest possible number of errors, tests must be conducted systematically and test cases must be designed using disciplined techniques. Test case design focuses on a set of techniques for the creation of test cases that meet overall testing objectives.

7.15 Project maintenance and reusability:

A reusable component may be code, but the bigger benefits of reuse come from a broader and higher level view of what can be reused. Software reuse can cut software development time and costs. The major advantages for software reuse are as following: Increase software productivity. The systematic development of reusable components. Improve software system interoperability. Shorten software development time. Produce better quality software. Develop software with fewer people. Move personnel more easily from project to project. The systematic reuse of these components as building blocks to create new systems. Reduce maintenance costs & software development. Produce more standardized software.

ASSIGNMENT NO 8

SOFTWARE QUALITY ASSURANCE

Software quality assurance (SQA) is a process that ensures that developed software meets and complies with defined or standardized quality specifications. SQA is an ongoing process within the software development life cycle (SDLC) that routinely checks the developed software to ensure it meets desired quality measures.

8.1 Quality Concepts:

Software quality assurance (SQA) is an umbrella activity that is applied throughout the software process. SQA encompasses

(1) A quality management approach,

(2) Effective software engineering technology (methods and tools),

(3) Formal technical reviews that are applied throughout the software process,

(4) A multi-tier testing strategy,

(5) Control of software documentation and the changes made to it,

(6) A procedure to ensure compliance with software development standards (when applicable),

(7) Measurement and reporting mechanisms.

Quality refers to measurable characteristics of software. These items can be compared based on the given standards:

1. Quality of design: Refers to the characteristics that designers specify for an item. The grade of materials, tolerances, and performance specifications all contribute to the quality of design. As higher-grade materials are used, tighter tolerances and greater levels of performance are specified, the design quality of a product increases, if the product is manufactured according to specifications.

2. Quality of conformance: It is the degree to which the design specifications are followed during manufacturing. Again, the greater the degree of conformance, the higher is the level of quality of conformance. Crucial is customer satisfaction (quality is only a part of it):

User Satisfaction = Compliant product + Quality + Delivery within budget and Schedule

8.2 Quality Control:

Quality control involves the series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements placed upon it. Quality control includes a feedback loop to the process that created the work product. A key concept of quality control is that all work products have defined ,measurable specifications to which we may compare the output of each process. The feedback loop is essential to minimize the defects produced.

Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that product quality is meeting its goals.

8.3 Cost of Quality:

The cost of quality includes all costs incurred in the pursuit of quality or in performing quality-related activities. Quality costs may be divided into costs associated with prevention, appraisal, and failure.

Prevention costs include:

1.Quality planning

2. Formal technical reviews
3.Test equipment

4.. Training

8.4 SQA Software Quality:

The goal is to achieve high-quality software. Software quality is defined as conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

The definition serves to emphasize three important points:

1. Software requirements are the foundation from which quality is measured. Lack of conformance to requirements is lack of quality.

2. Specified standards define a set of development criteria that guide the manner in which software is engineered. If the criteria are not followed, lack of quality will almost surely result.

3. A set of implicit requirements often goes unmentioned (e.g., the desire for ease of use and good maintainability).

If software conforms to its explicit requirements but fails to meet implicit requirements, software quality is suspect.

8.5 SQA Plan

8.5.1 Introduction:

Purpose: This document outlines the actions of our team in order to make our system "Telephone Billing System" and other related artifacts conform to the requirements of the customers and the qualitative standards within the specified project resources and constraints following IEEE standards.

Scope: The scope of this document is to outline all procedures, techniques and tools to be used for quality assurance of this project. This plan:

1. Identifies the SQA responsibilities of the project developer and the SQA consultant
2. Lists the activities, processes, and work products that the SQA consultant will review and audit
3. Identifies the SQA work products

8.5.2 Management:

A description of each major element of the organization along with the SQA tasks and their relationships. It consists of the following topics:

2.1 Organization: This project is teamwork where strength of each team is 3 members. The tasks are divided among the members by mutual co-ordination.

2.2 Roles

2.3 Tasks and Responsibilities:

The responsibilities of the developers are as follows:

1. Develop the requirement specification and cost estimation for the project.
2. Develop the design plan and test plan for testing the tool
3. Implement and test the application and deliver the application along with the necessary documentation.
4. Give a formal presentation to the committee on completion of the analysis,design and testing phases .The committee reviews the developer's work and provides feedback/suggestions.
5. Planning, coordinating, testing and assessing all aspects of quality issues.

The responsibilities of the committee members are to:

1. Review the work performed by the developer
2. Provide feedback and advice

8.5.3 Documentation:

In addition to this document, the essential documentation will include:

The Formal Specification Document gives the formal description of the product design specified in Object Constraint Language (OCL).

The Software Design Description (SDD) depicts how the software will be structured.

Software Test Plan describes the test cases that will be employed to test the product.

Software User Manual (SUM) identifies the required data and control inputs, input sequences, options, program limitations or other actions.

8.5.4 Standards, practices, conventions, and metrics:

Metrics - Lines of Code (LOC) and Function Points (FP) are used to measure the size of the software.

8.5.5 Reviews and audits:

The Committee will perform reviews at various stages of the project. This review will determine whether the requirements have been met for the deliverable, check that the product meets the requirements, ensure that

the SQA plan has been adhered to, verify the performance of the software and ensure that acceptance testing is carried out. In addition the developer will conduct a Formal Technical Review after the design phase. A design checklist will be used and the developer will check to see whether his/her design meets the checklist criteria.

8.5.6 Test:

Testing will be carried out in accordance with the Software Testing Plan (STP).Testing documentation will be sufficient to demonstrate that testing objectives and software requirements have been met. Test results will be documented and discussed in the final phase of the project.

8.5.7 Problem reporting and corrective action;

The corrective action process describes the steps for

1. Problem identification and correction occurring during software development to verify early detection of actual or potential problems
2. Reporting of the problem to the proper authority,
3. Analysis of the problem to propose corrective measures,
4. Timely and complete corrective action
5. The recording and follow-up of each problem's status.

8.5.8 Tools, techniques, and methodologies:

The tools are evaluated for adequacy by assessing whether they perform the desired functions and for applicability by assessing whether the tool capabilities are needed for the software development or support. Planned tools are evaluated for feasibility.

8.5.9 Code control: Code control includes the items listed below:

a. Identifying, labelling , and cataloging the software to be controlled
b. Identifying the physical location of the software under control
c. Identifying the location, maintenance, and use of backup copies
d. Distributing copies of the code
e. Identifying the documentation that is affected by a change
f. Establishing a new version
g. Regulating user access to the code.

SQA will conduct ongoing evaluations of the code control process to verify that the process of controlling the code is effective and in compliance with reference.

8.5.10 Records collection, maintenance, and retention:

Records and reports that provide a history of product quality throughout the software life cycle document SQA activities. Measurement data collected will be reviewed for trends and process improvement. All SQA records will be collected and maintained in the SDL or archival storage for the life cycle of the product or a minimum of few years.

8.5.11 Training:

Training never starts from a scratch. All those working for a project have minimal knowledge about the project and working conditions. Following are some necessary points:

1. Everyone involved or concerned with the project should be well versed with the problem statement and specifications of the project and familiar with the platform in which test cases are performed.
2. The working members should coordinate with each other so that everything remains on a track.
3. The concerned authority must dry run the program code and perform actual execution after rectifying the mistakes if any.

Training should be given to all involved in the development of the software to improve the quality of performance of the entire Software team.

Consider a Tester who is a member of the Software Team. He has to be trained to perform the various testing activities of the project. The steps that he will follow are:

1. First **Validation Testing** is performed while "Sign Up" and Unit Testing of Sign Up module is validated when he signs up successfully.Likewise is it done for addition deletion and so on.

2. He has to carry out **Unit Testing** of each of the modules such that the defined tasks are performed and the database is updated accordingly and the modules are as follows,
   - Adding Customer
   - Removing Customer
   - Modifying details of Customer

3. As and when the web pages of the modules are encountered, Validation Testing is performed for each of them.

4. The tester has to perform **Integration Testing** by navigating through the various pages of the system.

5. **Performance Testing** is then performed to check the response time(the time in which output is obtained) or efficiency of the system.

6. Finally, **Regression testing** is performed so that new bugs after changes in the system are encountered.

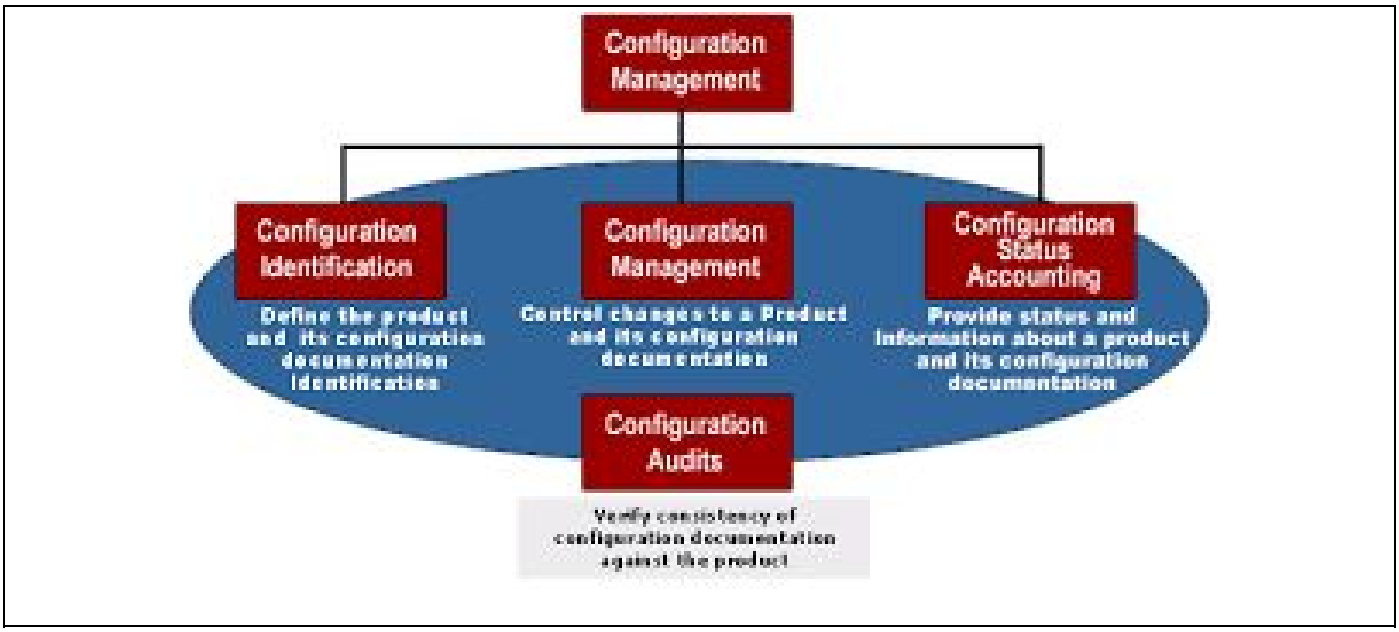
8.5.12 Risk management:

Risk Management is done according to software documentation audit of project.
Following are the primary and secondary roles played by each member of the group:

NAME	Primary Role	Secondary Role
Rupali Gawali	User Interface	Implementing Database, Designing
Chetana Patel	Requirements Gathering	Testing, Designing, Documentation
Shama Kamat	Functional Designing	Estimation, user Interface
Kaveri Kothe	Documentation	Requirements Analysis

ASSIGNMENT NO 9

SOFTWARE CONFIGURATION MANAGEMENT



Software configuration management (SCM) is an umbrella activity that is applied throughout the software process. Because change can occur at anytime, SCM activities are developed to

1.Identify change,
2. Control change,
3. Ensure that change is being properly implemented,
4.Report changes

Software configuration management (SCM) is a set of activities designed to control change by identifying the work products that are likely to change, establishing relationships among them, defining mechanisms for managing different versions of these work products, controlling the changes imposed, and auditing and reporting on the changes made.

Because many work products are produced when software is built, each must be uniquely identified. Once this is accomplished, mechanisms for version and change control can be established. To ensure that quality is maintained as changes are made, the process is audited; and to ensure that those with a need to know are informed about changes, reporting is conducted.

A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

9.1 Configuration Management:

The items that comprise all information produced as part of the software process are collectively called a software configuration for e.g. computer programs, documents and data.

Changes are inevitable and in most of the cases, justified. Customers may have modified requirements. Developers may want to modify the technical approach. Managers want to modify the project strategy. For this matter, changes must be

1. Analyzed in advance
2. Recorded before implementation
3. Reported by the need-to know basis
4. Controlled to improve quality and reduce errors.

The sources of changes can be stated as:

1. New business or market conditions
2. New customer needs demand modification of data produced by information systems,

   functionality delivered by products, or services delivered by a computer based system.
3. Reorganization or business growth/downsizing causes
4. Budgetary or scheduling constraints

9.2 Software Configuration Management (SCM) Process:

SCM Umbrella Activity has following subtasks

- Identifies
- Controls
- Audits
- Reports modifications that invariably occur while software is being developed and after it has been released to a customer. All information produced as part of software engineering becomes part of a software configuration. The configuration is organized in a manner that enables orderly control of change.

SCI: Software Configuration Items:

There are growing number of artifacts for manage sharing the SE process. They are

1. **Programs: Source level and executable forms**
2. **Documents: Technical practitioners and users a**
3. **Data: Internal and external**

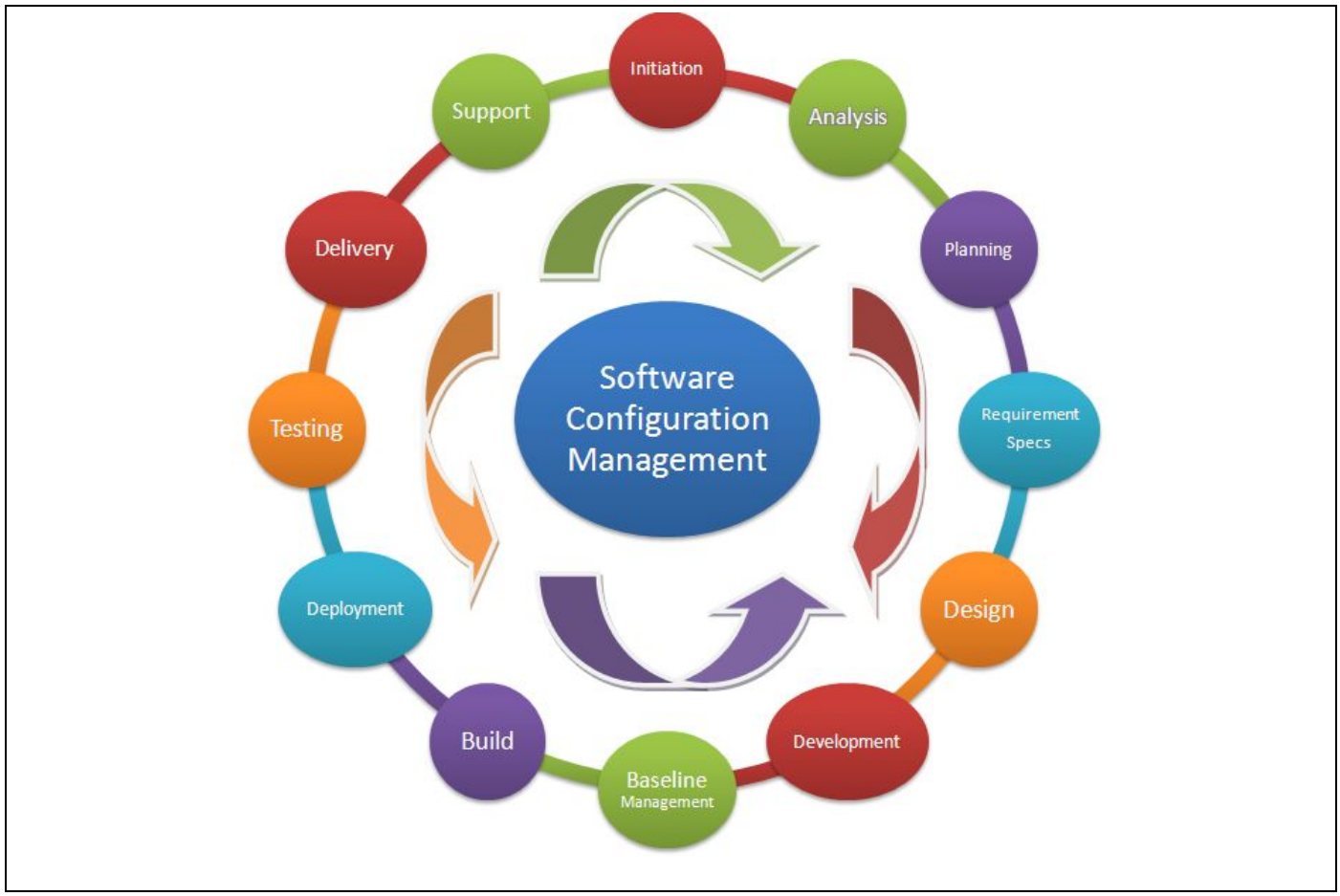
Types Of SCI:

1. Requirements specification
2. Project plan
3. Preliminary user manual
4. Design specification
5. Source code listings
6. Test specifications
7. Installations/operations
8. Executable programs
9. Database description
10. As-built user manual
11. Maintenance documents
12. Standards and procedures

SCM TASKS:

1. Identifying change: The identification scheme for software objects must recognize that objects evolve throughout the software process. Evolution graphs for each SCI are used .For example in Library Management System, the changes can be in terms of number of people accessing site simultaneously may increase.
2. Version control: Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process.

3. Control change: An engineering change order (ECO) is generated for each approved change. The ECO describes the change to be made, the constraints that must be respected, and the criteria for review and audit E.g. For the above change, an extra server may be arranged as control measure.
4. Configurational Auditing: Ensure proper implementation: A software configuration audit complements the formal technical review by assessing a configuration object.
5. Reporting changes to others.



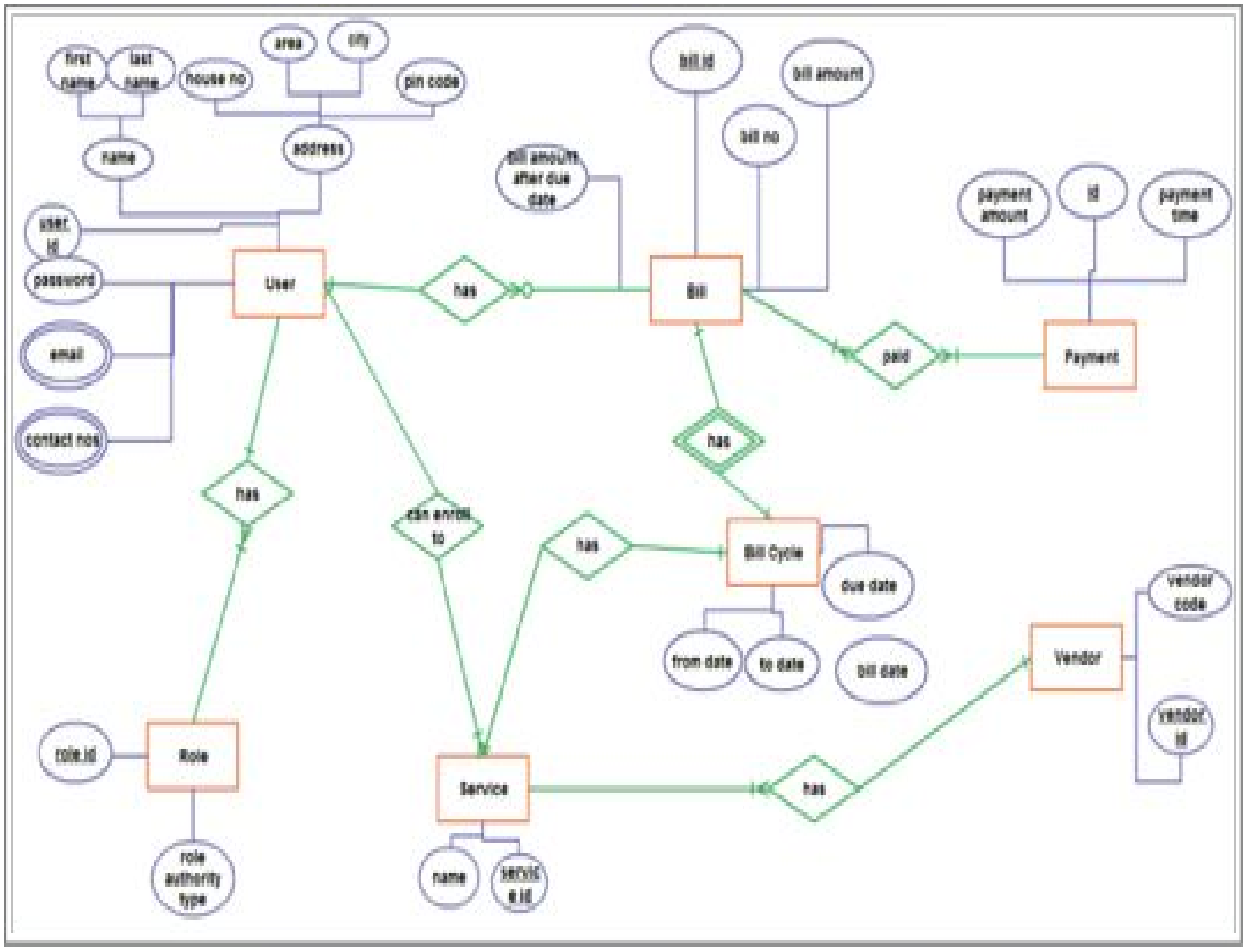
ASSIGNMENT NO 10

Requirement Analysis Modeling

Requirement analysis modeling uses combination of text and diagrammatic form to depict requirements for data,function and behavior in a way that is relatively easy to understand and more straight forward to review for correctness, completeness and convenience.

10.1 Data Modeling: Entity Relationship Diagram:

ERD depicts relationships between data objects. The ERD is the notation that is used to conduct the data modeling activity. The attributes of each data objects noted in the ERD can be described using data object description.



Use Case Analysis

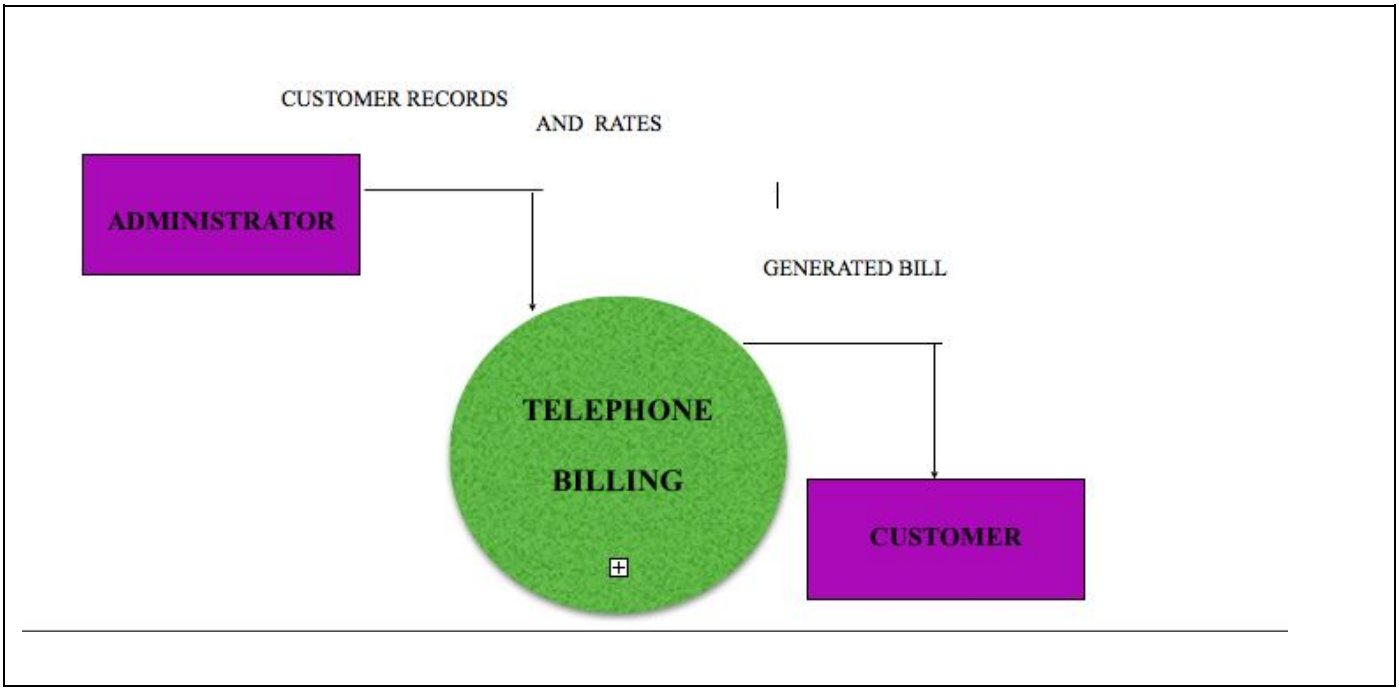
Actor	Role
Vendor	Vendor are the company's to which your telephone line connected and are the one who creates the plan of billing systems
User	Any person can become a user by taking telephone connection from vendor by entering few basic details such as their name ,address, email-id, contact no etc.Users can se their bill updates and their details by simply creating a user id and password
Service	Any person that manage Customer Care services

Use Case for Telephone Billing System

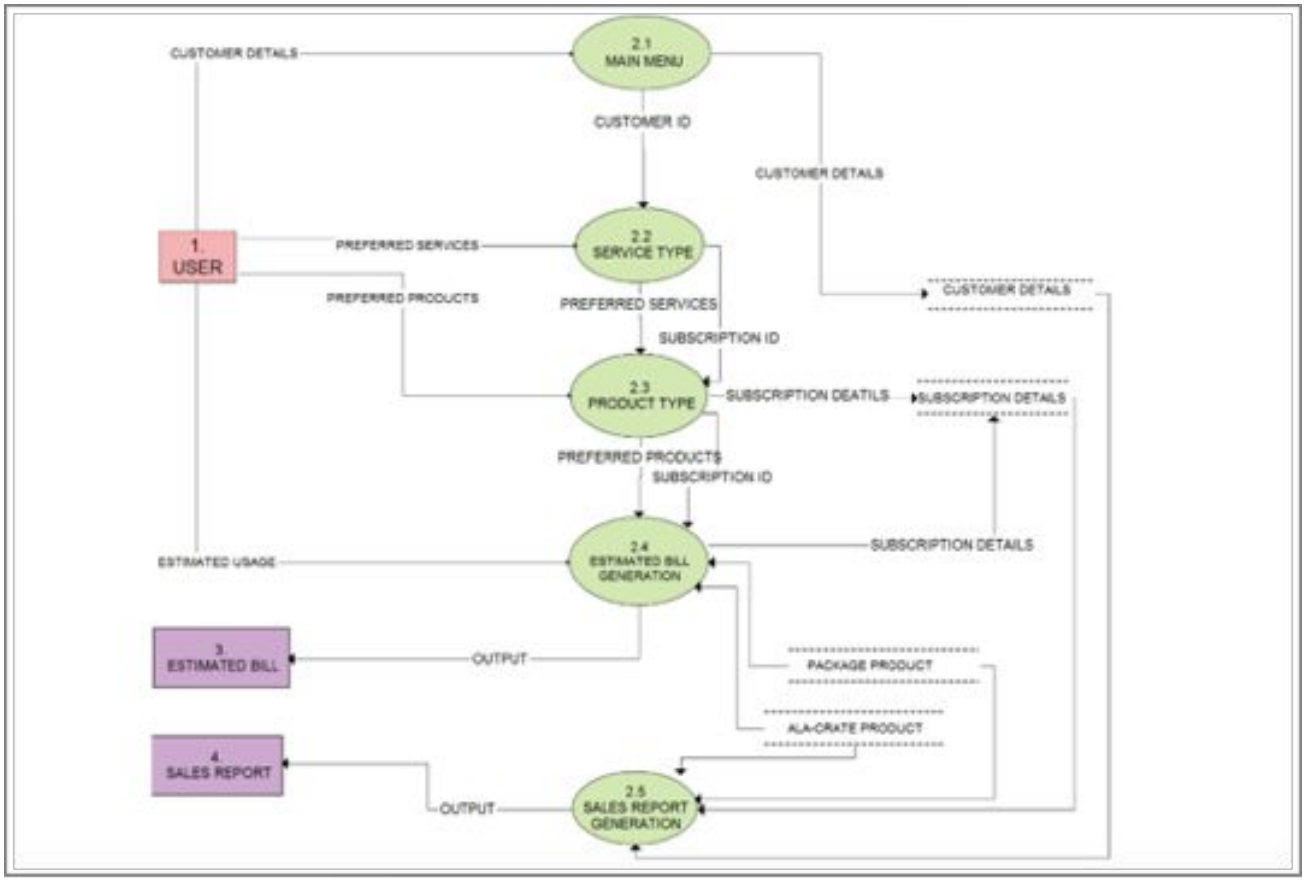
10.2 Functional Modeling: Data Flow Diagram

An activity that translates the data model developed during analysis into implementable data structures. Data models support data and computer systems by providing the definition and format of data .If this is done consistently across systems then, compatibility of data can be achieved. If the same data structures are used to store and access data then different applications can share data. The results of this are indicated below. However, the system and interface often cost more than they should, to build, operate and maintain. They may also constrain the business rather than support it. A major cause is that the quality of the data models implemented in system and interfaces are poor.

Level 0 DFD:

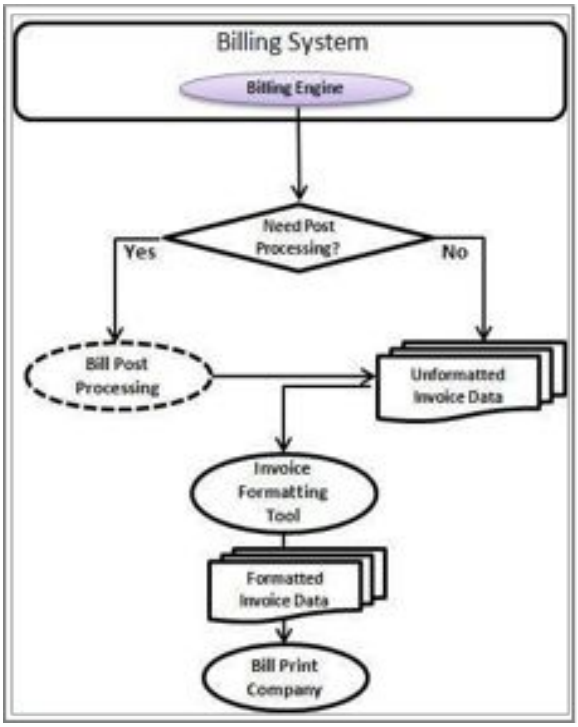


Level 1 DFD:



*here product=plan.

10.3 Behavioral Model



10.4 Data Dictionary

The analysis model encompasses representation of data objects, functions and control. In each representation data objects and/or control items play a role. Therefore, it is necessary to provide an organized approach for representing the characteristics of each data object and control item. This is accomplished with data dictionary.

Data Dictionary for the Telephone Billing System

Title of Object Class	Summary/Description
User	A person that is the user of the billing system
Vendor	A person with administrative privileges such as adding new contacts, deleting contacts if connection is refused by user, generating bills for each user, keeping details of each customer .
Bills	A System that keeps the track of everyone bills i.e details regarding each bills i.e.due date,bill amount, amount after due, bill_id
Payment	A System that keeps the payment track i.e payment done by user, payment printed in bill , payment_id.It first checks payment_id=bill_id,then the history regarding payments
Bill Cycle	A System that keeps the track of bill cycles of each user i.e date of issue, due date .

10.5 ALGORITHM:
1) Login (administrator and customer)

1.Enter username and password in the displayed form.

2.If (username and password are valid) then

Employee is directed to the welcome page

Else The system presents the login form again to the user

3.If (login is successful) then

The system responds by presenting a form to the user with fields for them to enter the following information:

New connection
Existing customer connection
Remove Customer
Administration options
Exit options

2)Adding Customer (administrator)

1.If(New connection is selected)

Fill name and phone number and submit

If(customer already exist)then

print error, go to home page

2. If (exit was pressed) then

The customer is redirected to the home page.

3)Remove Customer(administrator)

1.If(Remove Customer is selected)

Insert customer id and submit

If(customer exists)then

select delete account , go to home page

Else print customer does not exist

2. If (exit was pressed) then

The customer is redirected to the home page.

4) Existing Customer (customer or administrator)

1.If(bill viewing is selected)

then fill details for existing customer

if details are valid,

Print bill , go to home page

Else print customer does not exist

2. If (exit was pressed) then

The customer is redirected to the home page.

2.If(meter entry is selected)

then fill details for existing customer

if details are valid,

see bill amount as per meter reading entry , go to home page

Else print customer does not exist

2. If (exit was pressed) then

The customer is redirected to the home page.

3.If(modify is selected)

then fill details for existing customer

if (details are valid),

modify the details

Else print customer does not exist

2. If (exit was pressed) then

The customer is redirected to the home page.

5) Exit (customer and administrator)

if (existing customer)

select exit to logout and return to home page

If (administrator) then

select exit to logout and return to administrative options page

EXPERIMENT NO 11

DESIGN

11.1 Tables:

TABLE	ATTRIBUTES	TYPES
NAME	FIRST_NAME LAST_NAME	VARCHAR(20) VARCHAR(20)
ADDRESS	HOUSE_NO AREA CITY PINCODE	VARCHAR(10) VARCHAR(10) VARCHAR(10) INT(10)
USER	USER_ID PASSWORD EMAIL CONTACT_NO	INT(10)UNSIGNED VARCHAR(10) VARCHAR(20) INT(10)
VENDOR	VENDOR_CODE VENDOR_ID	VARCHAR(10) VARCHAR(10)
BILL CYCLE	DUE_DATE BILL_DATE FROM_DATE TO_DATE	DATE VARCHAR(10) DATE DATE
PAYMENT	PAYMENT_AMOUNT PAYMENT_ID PAYMENT_TIME	VARCHAR(20) VARCHAR(20) VARCHAR(10)
ROLE	ROLE_ID ROLE_AUTHORITY_TYPE	BIGINT(10)UNSIGNED VARCHAR(10)
SERVICE	SERVICE_NAME SERVICE_ID	VARCHAR(10) INT(10)
BILL	BILL_ID BILL_NUMBER BILL_AMOUNT	VARCHAR(20) INT(20) VARCHAR(20)

Name Table:

FIRST_NAME	LAST_NAME
CHETANA	PATEL
RUPALI	GAWALI
SHAMA	KAMAT
KAVERI	KOTHE

Address Table:

HOUSE_NO	AREA	CITY	PINCODE
B-101	Gokuldham Society, powdar gali ,FIlmcity Road Goregaon(E)	Mumbai	4000032
A-1201	Rustomjee tower, M.G.Road,Malad(E)	Mumbai	4000104
C-901	Anoml Blg, S.V.Road, Vile Parle(E)	Mumbai	400063
A-2402	Shanti Houseing co. Society, L.T.Nagar,Churchgate(W)	Mumbai	400003

User Table:

USER_ID	CONTACT_NO	EMAIL_ID
9769878762	022-27687954	chetanapatel1996@gmail.com
8976543212	65453432136	rupaligawali16@gmail.com
9765432132	022-28765432	shamakamat03@gmail.com
9768765438	022-2879876	kothekaveri@gmail.com

Bill Cycle Table:

DUE_DATE	BILL_DATE	FROM_DATE	TO_DATE
2015-10-26	2015-10-16	2015-09-15	2015-10-15
2015-07-10	2015-07-01	2015-06-01	2015-06-31
2015-08-12	2015-08-02	2015-07-01	2015-08-01
2015-10-16	2015-10-06	2015-09-05	2015-10-05

Payment Table:

PAYMENT_AMOUNT	PAYMENT_ID	PAYMENT_TIME
1576	9087656433	16:09
2345	6768979890	3:00
3254	543216788	14:54
1234	346576876	10:45

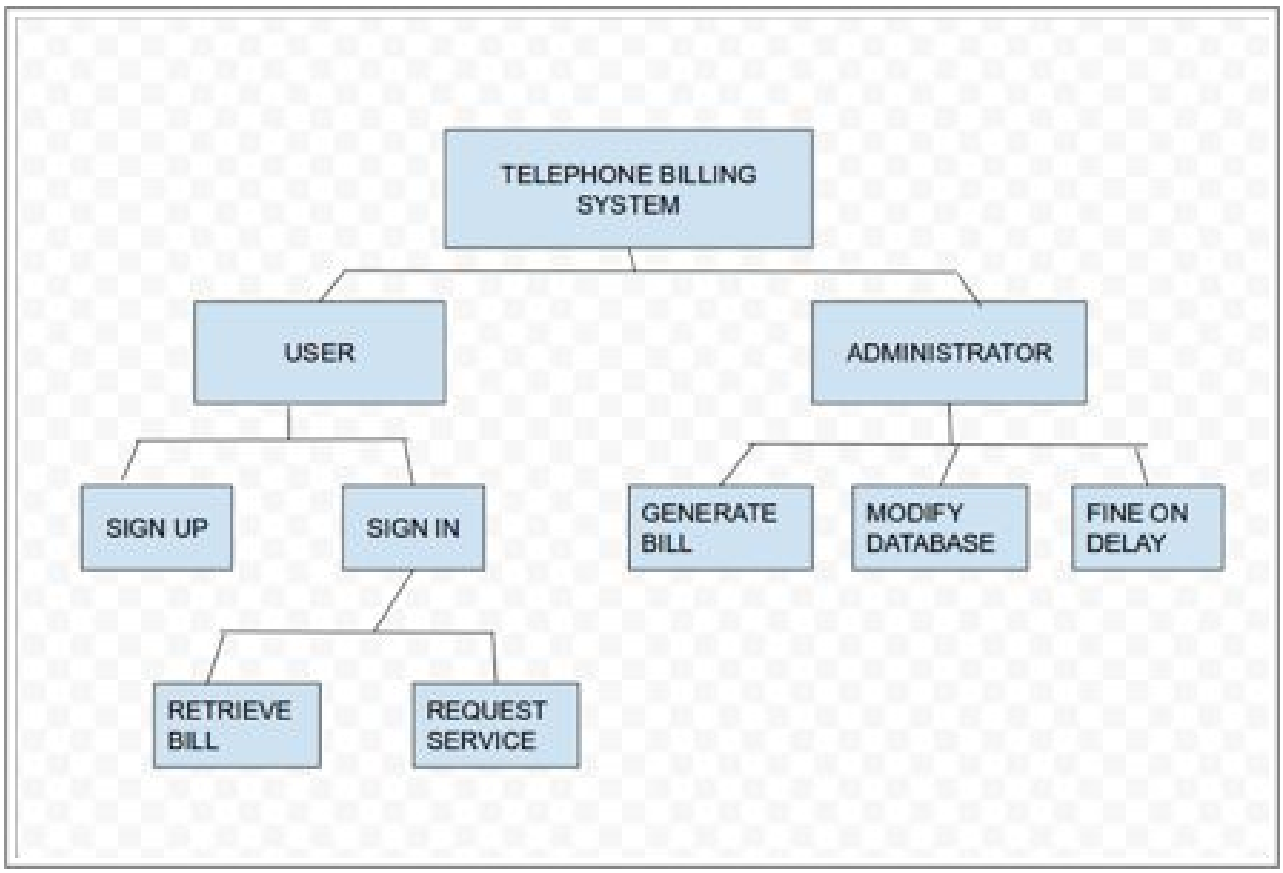
Service Table:

SERVICE_NAME	SERVICE_ID
Network_Problems	1
Bill_payments	2
changing connections	3
Line problems	4

Bill Table:

BILL_ID	BILL_AMOUNT
9087656433	1576
6768979890	2345
543216788	3254
346576876	1234

11.2 Architectural Design



11.3 PSPEC

Adding of customer

1. Customer/Employee enters personal details

2.Enters mobile no.

3.Enters a plan for the number

Adding of number

1.Customer signs in

2.Adds a new number

3.Enters a plan for the number

Modify the customer data

1.Employee enters customer no.

2.Views the list of mobile nos. used by the customer

3.Changes the mobile no. or plan or any other data about the customer.

Calculate bill

1.For a particular customer, employee checks the mobile nos. used by the customer

2.Referring to the Usage table, obtains the plan no. for the particular customer no and the mobile no.

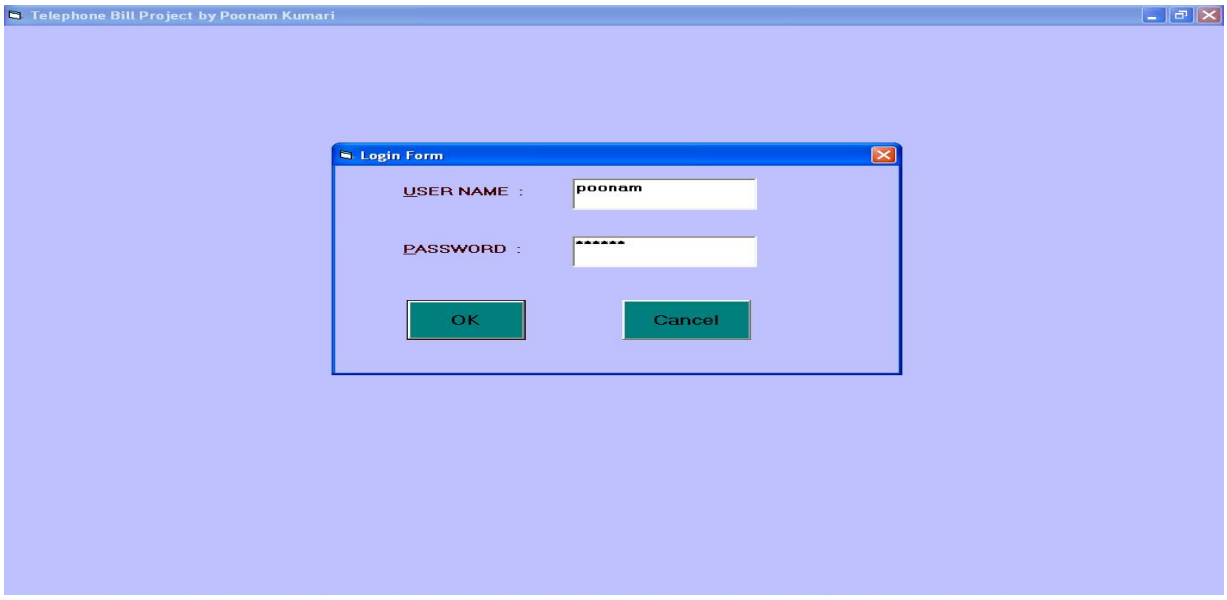
3.Using the Mobile table, obtains the billing information of that mobile no.,plan no. pair.

11.4 User Interface

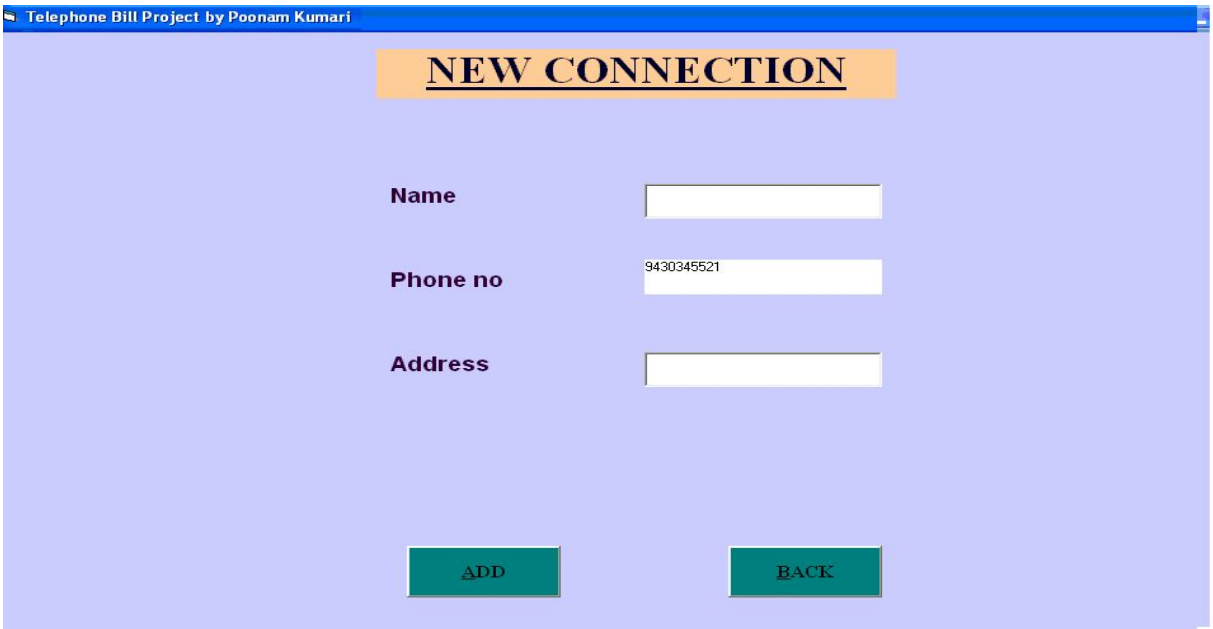
INPUT AND OUTPUT SCREEN

1. LOGIN FORM

2. WELCOME FORM

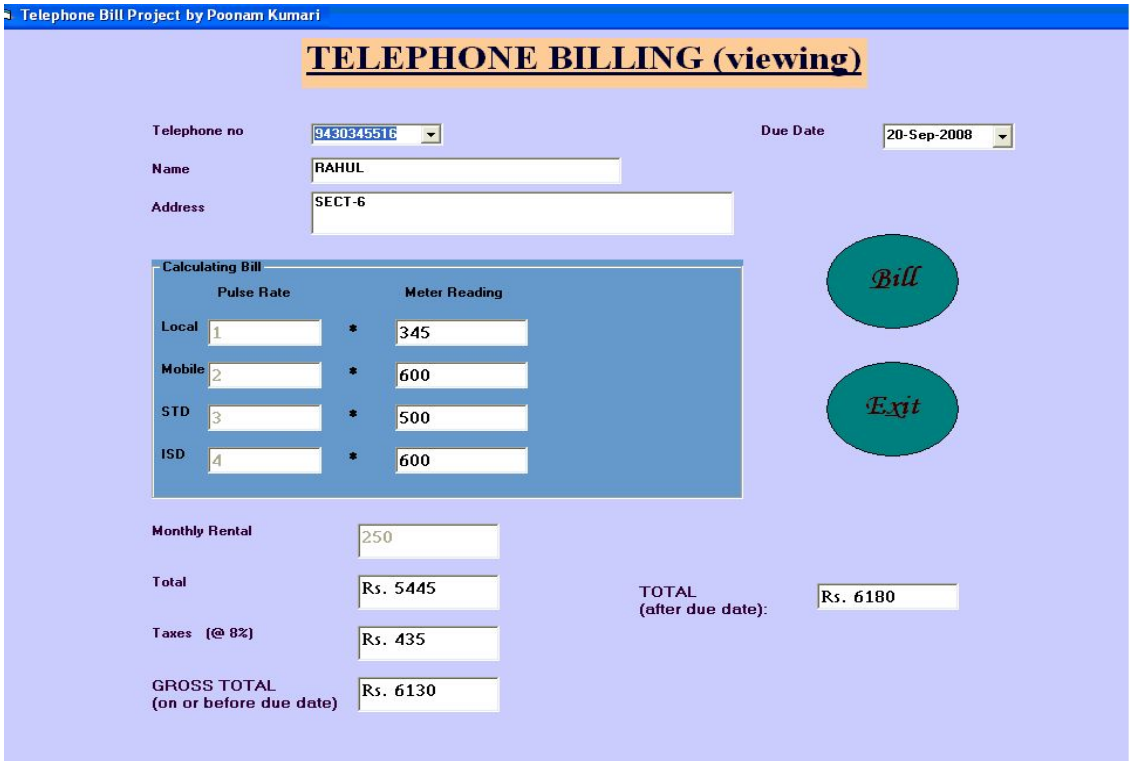


3. ENTRY FORM



4.CUSTOMER RECORD VIEWING FORM

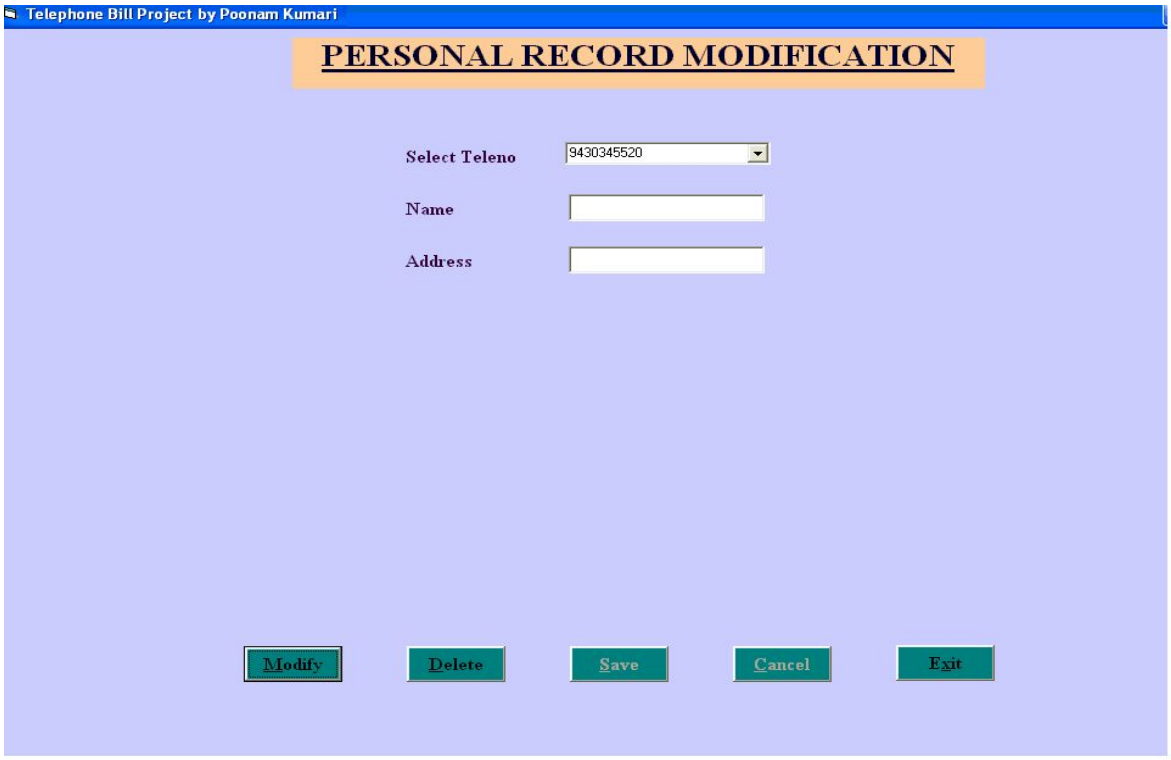
5.CUSTOMER BILL CALCULATION FORM



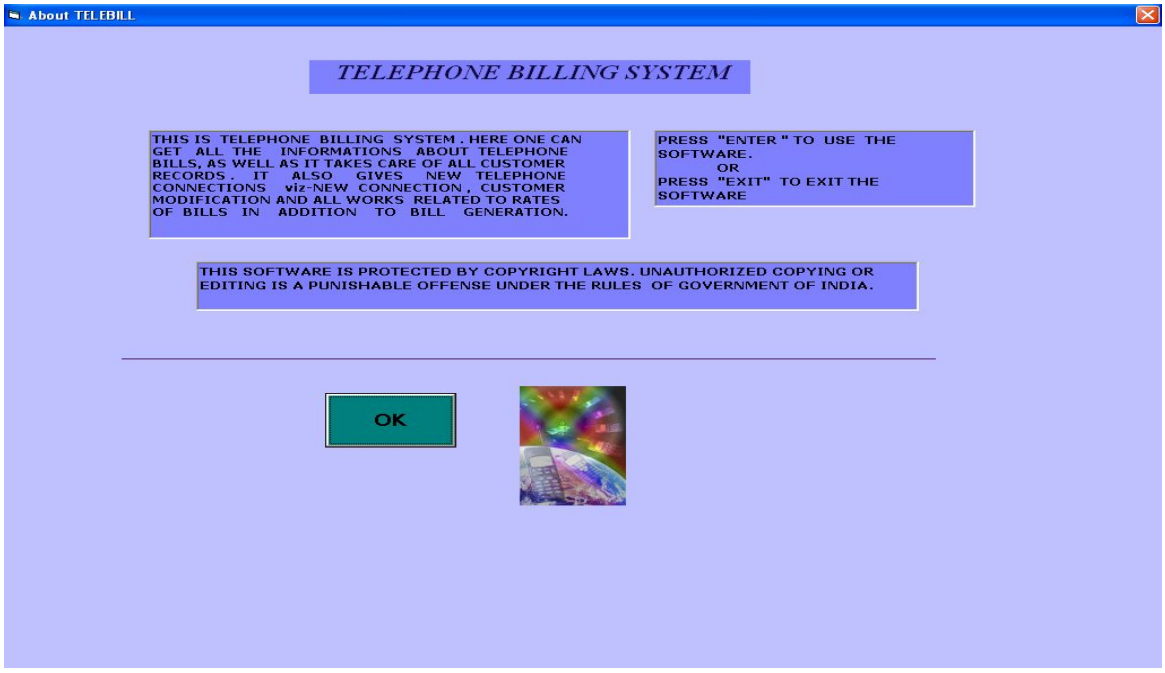
6.ADMINISTRATIVE FORM

7.METER READING FORM

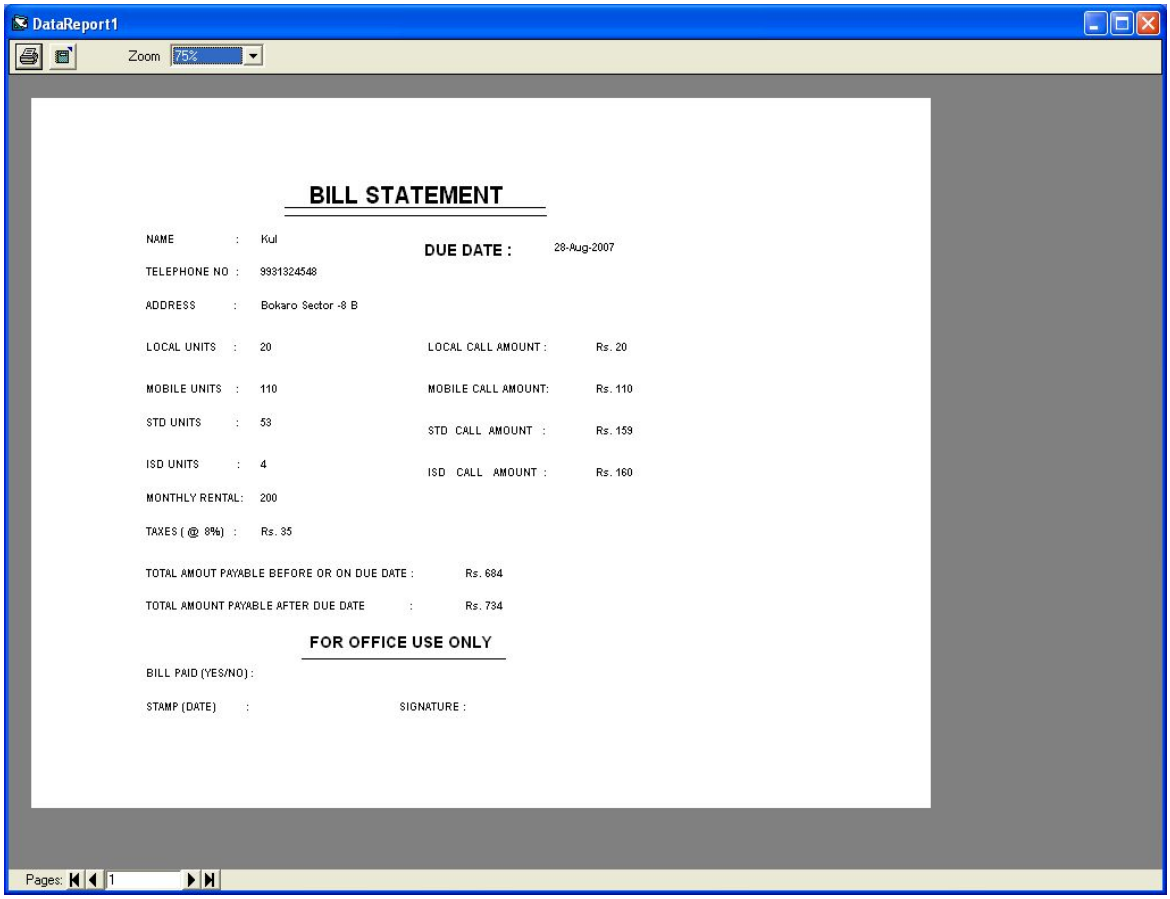
8. CUSTOMER RECORD MODIFICATION FORM



9. HELP FORM



11.BILL REPORT



ASSIGNMENT NO 12

SOFTWARE TESTING

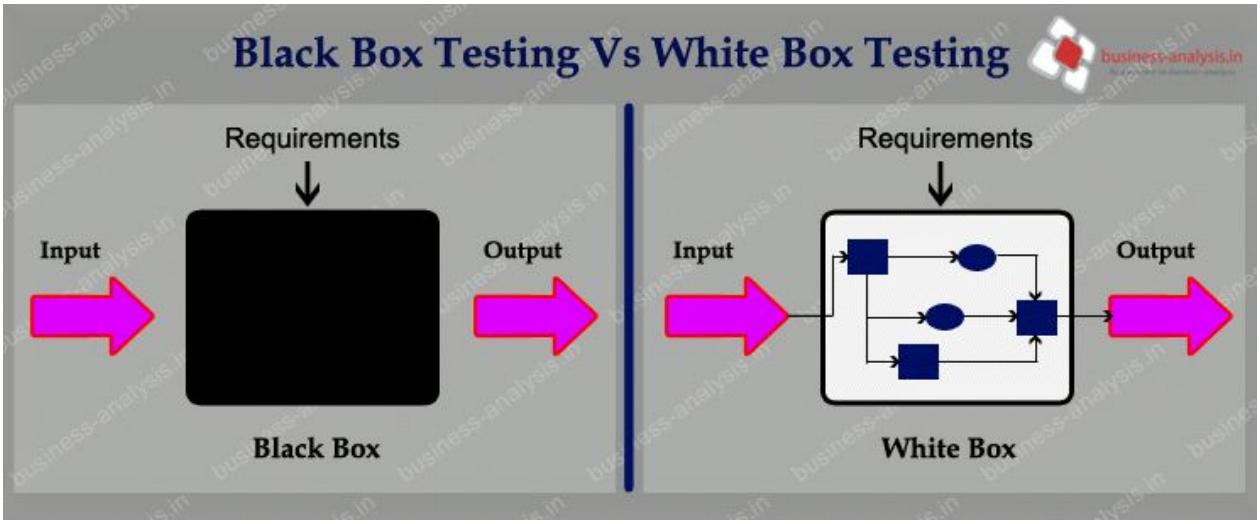
Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing presents an interesting anomaly of the software. During earlier definition and development phases, it was attempted to build software from abstract concept to a tangible implementation.Software testing methods are traditionally divided into white and black-box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

White Box Testing :

White box testing is a testing technique, that examines the program structure and derives test data from the program logic/code. The other names of glass box testing are clear box testing, open box testing, logic driven testing or path driven testing or structural testing.

Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. Independent Testing Team usually performs this type of testing during the software testing life cycle.This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.



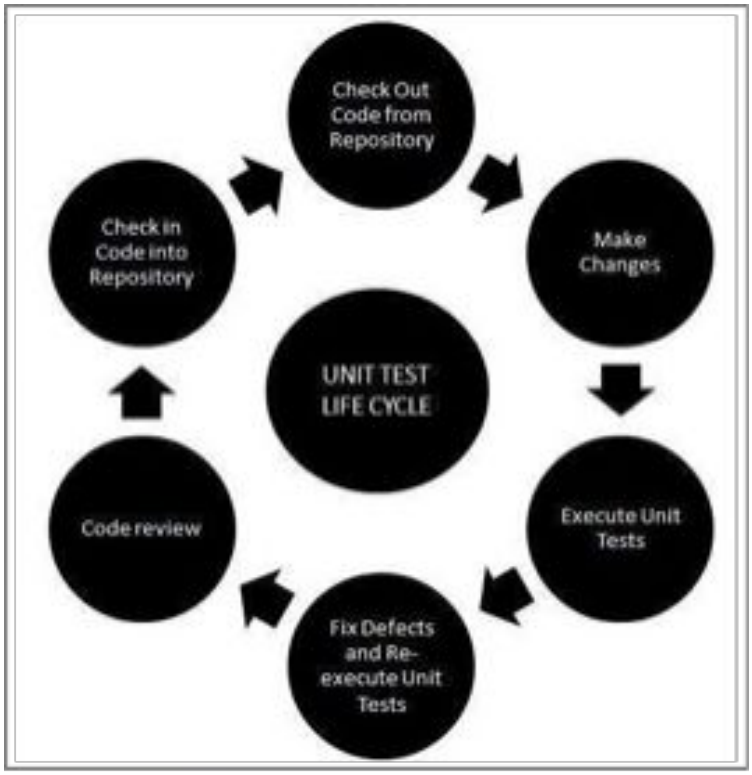
The testing phase involves the testing of the developed system using various set data. Presentation of test data plays a vital role in system testing. After preparing the test data the system under study was tested using test data. While testing the system by using test data errors were found and corrected. A series of tests were performed for the proposed system before the system was ready for implementation. The various types of testing done on the system are:

- Unit Testing
- Integration Testing
- Validation Testing

12.1 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design, the module. It comprises the set of test performed by the programmer prior to integration of the unit into larger system. The testing was carried out during the coding stage itself. In this step each module is found to be working satisfactorily as regards to the expected output from the module**.**

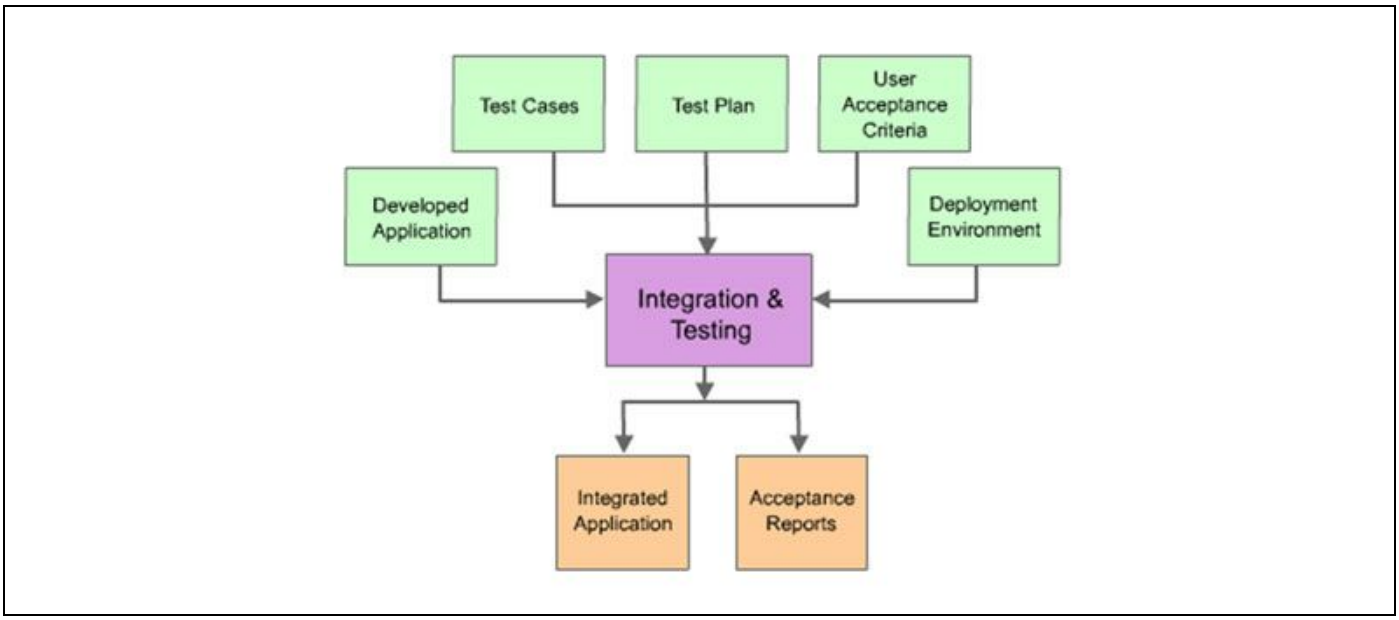
TEST NAME	TEST ID	TEST CASE	STEPS	TEST DESCRIPTION	EXPECTED RESULT	ACTUAL RESULT	SEVERITY	STATUS
UNIT TESTING	**UT0**	sign up	enter the login details . -id, password	new user should be created	login details submitted to database and new user is created	new user created successfully	mild	pass
	UT1	retrieve	enter the request -ex.bill amount	request is accepted and processed	user gets the respective data	retrieve is successful	mild	fixed
	UT2	add	fill details in the form	new entry should be created in database	new entry is created in the database	entry created successfully	mild	pass
	UT3	delete	search for entry to be deleted	entry should be deleted in database	entry is deleted from the database	deletion is successful	mild	fixed
	UT4	modify	search for entry to be modified	make necessary changes	entry is modified	modification is successful	mild	pass



12.2 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover error associated within the interface. The objective is to take unit tested modules and build a program structure that has been dictated by design. All modules are combined in this step. The entire program is tested as whole. And chaos in interfaces may usually result. A set of errors is encountered in such a case.

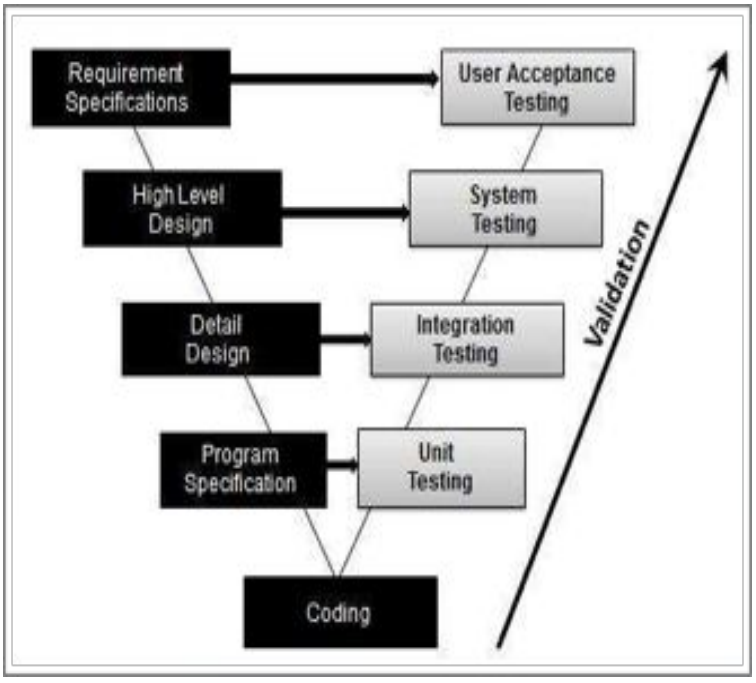
TEST NAME	TEST ID	TEST CASE	STEPS	TEST DESCRIPTION	EXPECTED RESULT	ACTUAL RESULT	SEVERITY	STATUS
INTEGRATION TESTING	**IT0**	sign up	enter the login details- id, password	new entry should be added to all related entities	new entry is added to all related entities	new entry is added to all related entities successfully	severe	pass
	IT1	retrieve	enter the request ex, bills due	all related entities should respond to the request	user retrieves data from all related database	retrieve is successful	severe	pass
	IT2	add	fill details in the form	new entry should be added to all related entities	new entry is added to all related entities	new entry is added to all related entities successfully	severe	pass
	IT3	delete	search for entry to be deleted	entry should be deleted from all related tables	entry is deleted from all the related tables	entry deleted successfully	severe	pass
	IT4	modify	search for entry to be modified	using triggers ensure that there is no redundancy	entry is modified with the help of trigger	execution successful hence modified	severe	pass



12.3 VALIDATION TESTING

Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment. It answers to the question, Are we building the right product?
Validation Testing - Workflow:
Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.

TEST NAME	TEST ID	TEST CASE	STEPS	TEST DESCRIPTION	EXPECTED RESULT	ACTUAL RESULT	SEVERITY	STATUS
VALIDATION TESTING	**VT0**	sign up	enter the login details- id, password	attribute is entered in string format instead of float, error displayed	error showing invalid format	error displayed correctly	negligible	fixed
	VT1	retrieve	enter the request ex. account details	attribute not found	error showing invalid request	error displayed correctly	negligible	pass
	VT2	add	fill details in the form	new entry with same primary key should not be added	error showing duplicate entry	error displayed correctly	negligible	fixed
	VT3	delete	search for entry to be deleted	entry is not found	error showing invalid search	error displayed correctly	negligible	pass
	VT4	modify	search for entry to be modified	entry is not found	error showing invalid request	error displayed correctly	negligible	pass



ASSIGNMENT NO 13

CONCLUSION

The 'Telephone Billing System' aims towards the automatization of the billing process of a service provider. The system has computerized the management of the extensive data of the customers and the telephone use of all of them. The process of billing requires monitoring every call of every customer of the company. The maintenance of data and its processing to generate a bill is very tedious and error prone, hence we need computerization. The system made, keeps records of the call summary of all customers, number of calls of each customer and the cost that each should be charged based on their chosen plan, and thus generates the bill accordingly. The records can be edited and updated much more conveniently using this system. The system also has a good user interface to enhance the user experience and there are interfaces connecting the various modules together to allow navigation through the various modules and pages. This system has made the billing process lucid for both the service providing company and the customers, and thus saves many effectively utilizable man-hours.
The highlight of this project is the security of the customer's personal information. The main aim of this project is to minimize the human work and at the same time make it speedy. It is easy to understand and reduces the paperwork. Maintenance of database as well as overall project will become easy.

13.1 Final Thoughts

In the course of making this project, we learnt that there are a large number of minute aspects which are crucial for the accuracy of the functions of this project and hence must be implemented without any leniency. Every single module has a significant role in the functionality and all the modules are interrelated and interdependent, hence integrating appropriately is very important. We learnt how to work efficiently in a team,we understood what teamwork is, along with strengthening the concepts of software engineering.We feel satisfied that the aim of our project is accomplished.

Thankyou.