SOFTWARE REQUIREMENT SPECIFICATION

of Online ERP Solution

FOR MEGHNA LIFE INSURANCE

The system is designed in such a way that it accepts and stores the input data, process and produce output under the direction of a detailed step by step stored programmed instruction. This system includes Client Dairy, Client Details, add new Client, Agent Diary, Agent Details, add new Agent, Policy Details, Renewals, Receive, Payment, Inventory, HR and Accounts Details information's and gives details based on the entire operation of Meghna Life Insurance. This system is necessary for Storing Information, assessing Workload and hence their efficiency. The System provides the adequate information to the concern for its smooth run.

Prepared By

Techsol Technologies Bangladesh Limited

ISO 9001:2015 Certified

1. INTRODUCTION

1.1. Overview of the Project

The project entitled as **"ERP Solution for Meghna Life Insurance"** is developed in a manner to help the total operation process of the company. It will be developed using .Net Framework as Front-End and Oracle (Version as per the license of Meghna Life Insurance) as the Back-End tool.

The system is designed in such a way that it accepts and stores the input data, process and produce output under the direction of a detailed step by step stored programmed instruction. This system includes Client Diary, Client Details, add new Client, Agent Diary, Agent Details, add new Agent, Policy Details, Renewals, Receive, Payment, Inventory, HR and Accounts Details information's and gives details based on the entire operation of Meghna Life Insurance. This system is necessary for Storing Information, assessing Workload and hence their efficiency. The System provides the adequate information to the concern for its smooth run.

1.2. About Existing System

1.2.1. Manual and Semi-Automatic System:

This system involves manual involvements, which may lead to problems. It is very tedious to handle large volume of data and requires more time to handle the information, which results in slow processing.

Thus the computerization will make it very easy and accurate at all the processing stages. Report preparation requires more time so by computerizing, it will be very easy for the management. So the existing system is to be changed and put for computerization.

1.2.2. Drawbacks of the Existing System

1. Tedious to handle large volume of data.
2. Time consumption.
3. Maintaining the manual registers is cumbersome.
4. Decision making is very slow.

2. SYSTEM ANALYSIS

2.1. Proposed System

The Proposed System has been designed using .Net Framework as front-end and Oracle (Version as per the license of Meghna Life Insurance) as back-end. The proposed system is a computerized on with GUI interface. It takes care of the process under an easy environment to work with.

2.1.1. Proposed System has the following facilities:

The new system is under-friendly.

- The systems provide all possible options to new entry, editing, updating and searching the Policy, Client, Agent and any sorts of transaction related to operation.
- The system have the search options in many ways, it is very user friendly.
- The system is highly portable.
- The system provides very good database Management System.

3. SYSTEM REQUIREMENT SPECIFICATION

3.1. Server Specification

Processor	Minimum Intel Xeon E3-1230 Quad-core - 3.2GHz - 4 Core
RAM	Minimum 64GB DDR3 ECC
Disk Controller	RAID 1
Hard Drive	2TB SATA
Second Hard Drive	2TB SATA
Operating System	Windows Server 2012 Standard (x64)
Database Software	Oracle RDBMS (Version as per the license of Meghna Life Insurance)
Control Panel	Website Panel
SmarterBundle	SmarterBundle by SmarterTools
Mail Server Software	SmarterMail Pro Leased - Unlimited
SmarterStatsPro - Leases	SmarterStats Pro Leased - 250
CommTouch Anti-Spam - Leases	Leased CommTouch 250
Commtouch Zero-hour Antivirus - Leases	Leased CommTouch AV 250

Exchange ActiveSync - Leases	Leased ActiveSync 25
Software Firewalls	Microsoft Windows Firewall
Support Level	Full Management
Bandwidth	10 TB
Port Speed	100Mbps
IP Addresses	5 IPs
AntiVirus	McAfee Total Protection
NAS Backup	20GB NAS

3.2. Software Specification

Front-End : .Net Framework, AJAX

Back-End : Oracle (Version as per the license of Meghna Life Insurance)

Operation System : Windows Server 2012 (Minimum)

4. BENEFIT OF USING .NET FRAMEWORK AS FRONT END

ASP.NET Web Forms and MVC are two web frameworks developed by Microsoft - they are both good choices. Neither of the web frameworks are to be replaced by the other nor are there plans to have them 'merged' into a single framework. Continued support and development are done in parallel by Microsoft and neither will be 'going away'.

Each of these web frameworks offers advantages/disadvantages - some of which need to be considered when developing a web application. A web application can be developed using either technology - it might make development for a particular application easier selecting one technology versus the other and vice versa.

4.1. ASP.NET Web Forms:

- **Development supports state**: Gives the illusion that a web application is aware of what the user has been doing, similar to Windows applications. I.e. makes 'wizard' functionality a little bit easier to implement. Web forms does a great job at hiding a lot of that complexity from the developer.

- **Rapid Application Development (RAD)**: the ability to just 'jump in' and start delivering web forms. This is disputed by some of the MVC community, but pushed by Microsoft. In the end, it comes down to the level of expertise of the developer and

what they are comfortable with. The web forms model probably has less of a learning curve to less experienced developers.

- **Larger control toolbox**: ASP.NET Web Forms offers a much greater and more robust toolbox (web controls) whereas MVC offers a more primitive control set relying more on rich client-side controls via jQuery (JavaScript).

- **Mature**: It's been around since 2002 and there is an abundance of information with regards to questions, problems, etc. Offers more third-party control - need to consider your existing toolkits.

4.2. ASP.NET MVC:

- **Separation of concerns (SoC)**: From a technical standpoint, the organization of code within MVC is very clean, organized and granular, making it easier (hopefully) for a web application to scale in terms of functionality. Promotes great design from a development standpoint.

- **Easier integration with client side tools (rich user interface tools)**: More than ever, web applications are increasingly becoming as rich as the applications you see on your desktops. With MVC, it gives you the ability to integrate with such toolkits (such as jQuery) with greater ease and more seamless than in Web Forms.

- **Search Engine Optimization (SEO) Friendly / Stateless**: URL's are more friendly to search engines (i.e. mywebapplication.com/users/ 1 - retrieve user with an ID of 1 vs mywebapplication/users/getuser.aspx (id passed in session)). Similarly, since MVC is stateless, this removes the headache of users who spawn multiple web browsers from the same window (session collisions). Along those same lines, MVC adheres to the stateless web protocol rather than 'battling' against it.

- **Works well with developers who need high degree of control**: Many controls in ASP.NET web forms automatically generate much of the raw HTML you see when a page is rendered. This can cause headaches for developers. With MVC, it lends itself better towards having complete control with what is rendered and there are no surprises. Even more important, is that the HTML forms typically are much smaller than the Web forms which can equate to a performance boost - something to seriously consider.

- **Test Driven Development (TDD)**: With MVC, you can more easily create tests for the web side of things. An additional layer of testing will provide yet another layer of defence against unexpected behaviour.

Authentication, authorization, configuration, compilation and deployment are all features that are *shared* between the two web frameworks.

5. BENEFIT OF USING ORACLE AS BACK END

5.1. Customer Satisfaction

One advantage of using Oracle databases is due to Oracles belief in customer satisfaction. For example, all Oracle databases are backward compatible. This allows businesses to upgrade their systems without a complete overhaul of their database system. This provides efficient and low-cost updates. In addition, new versions of Oracle databases provide new features while keep the popular features from older versions. This ensures that their product is based on the customer's function rather than what is cost-effective for Oracle.

5.2. Functionality

Oracle databases are used for practically all corporation level applications. Thus, Oracle databases dominate the banking industry. Ten of the world's top banks use Oracle databases to conduct their business. This is primarily due to Oracle's functionality. They provide a combination of high-level technology and integrated business solutions. This is a perfect for practically all corporations that have huge amounts of data to store and access.

5.3. Reliability

Another important advantage offered by Oracle databases are their reliability. Oracle is a database that delivers excellent performance when challenged with demanding tasks. The ACID test, which is an important tool used to ensure the integrity of data stored, was easily passed by Oracle databases. This test is important since reliable data storage is the main purpose of a database. Thus, Oracle databases have been proven to deliver high integrity of data storage.

5.4. Flashback Technology

Oracle databases incorporate Flashback technology, which is a significant advantage. In the event of an application outage, due to any number of reasons, it is important not to lose data stored on a database system. Oracle's Flashback technology allows for efficient recovery of data incorrectly deleted or lost. Thus, Flashback technology essentially removes human error and increases database recovery time. In the end, this feature allows for a simplified management and administrative process.

5.5. Other Advantages

Oracle databases have also successfully featured the four properties that all database systems must have. These four properties are atomicity, consistency, isolation and durability. All four of the properties are well maintained by Oracle databases, thus providing a reliable and competent database system.

6. SYSTEM DESIGN

Design is the first stepping stone for the development of any system. The process of system is very elaborate and should be carried with almost care as the actual system get its shape at this phase. This consists of four main stages.

They are

1. Input design
2. Output design
3. Screen design
4. Code design
5. Database design

6.1. Input Design:

The input design is the link between the information system and the user. It comprises the director specification and procedures for data preparation and those steps that are necessary to put transaction data into a usable form for processing data entry. The activity if putting data in to the computer for processing can be achieved by inputs, focuses on controlling the amount of inputs required, controlling errors, avoiding delay, avoiding extra steps and keeping the process simple.

System analyst decides the following input design details:

1. Why data to input?
2. What medium is used?
3. How the data should be arranged or coded?
4. The dialog to guide users in providing input
5. Data item and transaction needing validation to detect errors
6. Methods for performing input validation and steps to follow when error occur.

6.2. Output Design:

Computer output is the most important and direct source of information to the user. Output design is a process that involves designing necessary output that have to be given to various users according to their requirements. Efficient, intelligible output design will improve the system relationship with the user and help indecision making. Since the reports are directly required by the management for taking decision and to draw conclusion, they must be designed with almost care to the user. The options for the output and report are given in the system menu. When designing output, system analyst must accomplish the following:

1. Determine the information to present.

2. Determine whether to display, print, or “speak” the information and select the output to intended receipts.
3. Arrange the present of information acceptable format.
4. Determine how to distribute the output to intended receipts.

6.3. Screen Design:

Screen design begins with recognition that the screen is composed of different areas. Layout tools assist the analyst in specifying the content of the signal and multiple design formats. All screen have been provided with menus, push buttons facilities, a icons and controls buttons such as Add/Delete/Update/Find/Clear/Exit etc. The main screen consists of main menu from which we can move to another forms or screen. In designing output screens we need area for:

1. Heading and titles.
2. The content of display.
3. Message and instructions.
4. Sometimes explanations for information in the report.

6.4. Code Design:

A code is a group of characters designed to identify the data items of the same type uniquely. When hospital volume of data are handled, it is essential that each one identified, selected and retrieved quickly and easily and there arises the need for the codes. Codes should be designed carefully and should be meaningful. It should abide the general rules of the code designing and should be understood by the user.

6.5. Database Design:

Database files are the key source of the information into the system. The final design must satisfy user needs in terms of completeness, integrity, performance and other factors. The main objective of the database design is to store the captured data efficiently and facilitate its timely retrieval.

7. DESIGN NOTATIONS

7.1. Table Design:

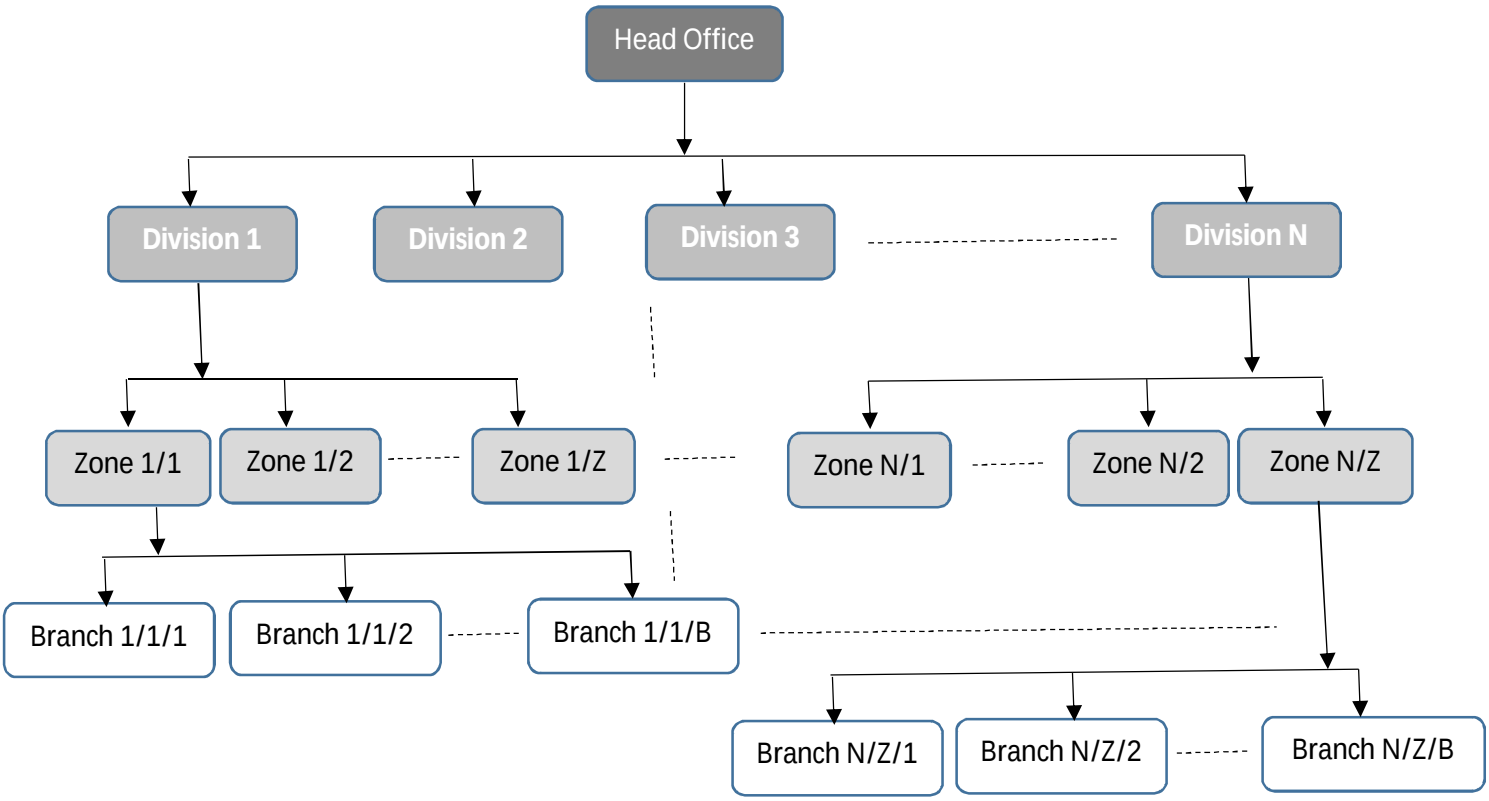
7.1.1. Branch Category Master :

Generally Meghna follows 3 Categories of Branches. These are Division-->Zone-->Branch. But considering the future Scope we are deciding to make it unlimited level. So the Table Structure for this table will be

Field Name	Data Type
Category ID	Integer [PK]
Category	Varchar
Parent Category	Integer

7.1.2. Branch Master:

Meghna will follow following tree to organize their process throughout the country. But as we are considering the Organizational tree as unlimited depth. So we are proposing the following flow of organogram.



So the Table Structure for this table will be

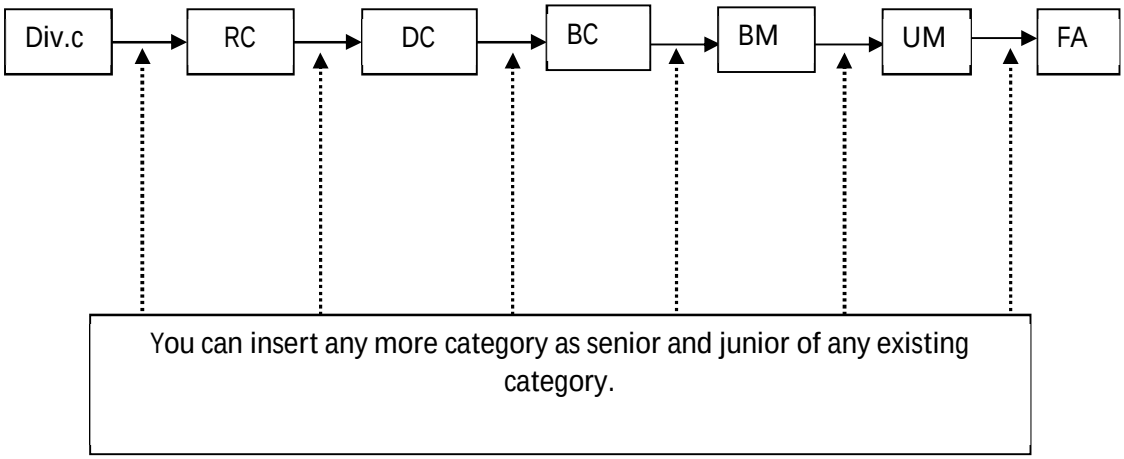
Field Name	Data Type	Description
Branch ID	Integer (Auto Generate) [PK]	
Branch Category ID	Integer	
Branch Name	Varchar	
Branch Name in Bengali	Varchar	
Address	Varchar	
Address in Bengali	Varchar	
Contact No	Varchar	
Mail ID	Varchar	
Division	Integer	Linked with the table "**Division Master**"
District	Integer	Linked with the table "**District Master**"
Pin Code	Integer	Linked with the table "**Pin Code Master**"
Admin In charge	Integer	Linked with the Table **"Employee Master"**
Development In charge	Integer	Linked with the table "**Agent**" by Agent ID
Minimum Amount Allowable to Keep at Branch	Currency	
Minimum Days Allowable to Keep the Collection	Real	

7.1.3. Agent Category Master:

Generally Meghna follows 7 Categories of Development Officers or Agent. These are Div. C-->RC-->DC-->BC-->BM-->UM-->FA. But considering the future Scope we are deciding to make it unlimited level. So the Table Structure for this table will be

Field Name	Data Type	Description
Agent Category ID	Integer [PK]	
Agent Category	Varchar	
Abbreviation	Varchar	
Senior Category	Integer	
Status	Boolean	If "True", then Employee of Meghna; If "No" then Employee ofCentral Organization

The flow will be as like as the following:



7.1.4. Academic Qualification Master :

The field naming "Academic Qualification" at "Insurance Proposal Form" will be a dropdownlist. So the list of items of the dropdownlist will populate from this master. So that Admin and Add/Delete/Modify the list of items under this field. So the table structure for this master will be as like as following:

Field Name	Data Type
Academic Qualification ID	Integer (Auto Generate) [PK]
Academic Qualification	Varchar

7.1.5. Master for Purpose of Insurance Policy:

The field naming "Purpose of Insurance Policy" at "Insurance Proposal Form" will be a dropdownlist. So the list of items of the dropdownlist will populate from this master. So that Admin and Add/Delete/Modify the list of items under this field. So the table structure for this master will be as like as following:

Field Name	Data Type
Purpose of Insurance Policy ID	Integer (Auto Generate) [PK]
Purpose of Insurance Policy	Varchar

7.1.6. Master for Relation:

The field naming "Relation" of Nominee or other at "Insurance Proposal Form" or "Other" will be a dropdownlist. So the list of items of the dropdownlist will populate

from this master. So that Admin and Add/Delete/Modify the list of items under this field. So the table structure for this master will be as like as following:

Field Name	Data Type
Relation ID	Integer (Auto Generate) [PK]
Relation	Varchar

7.1.7. Occupation Category Master:

This table contains category of Occupation. So the table structure for this master will be as like as following:

Field Name	Data Type
Occupation Category ID	Integer (Auto Generate) [PK]
Occupation Category	Varchar

7.1.8. Occupation Master:

The field naming “Occupation” at “Insurance Proposal Form” will be a dropdownlist. So the list of items of the dropdownlist will populate from this master. So that Admin and Add/Delete/Modify the list of items under this field. So the table structure for this master will be as like as following:

Field Name	Data Type	Description
Occupation ID	Integer (Auto Generate) [PK]	
Occupation Category	Integer	Linked with Occupation Category Master
Occupation	Varchar	

7.1.9. Department Master:

To create the operator of the software, it is require integrate the Core Operation Module with the HR Module. So it is very important to create the HR Based Organogram. The very first step to create HR based Organogram, is to create Departments. The table structure for the department master with unlimited depth of department will be as following.

Field Name	Data Type
Department ID	Integer (Auto Generate) [PK]
Department Name	Varchar
Parent Department	Integer

7.1.10. Designation Master:

The second step to create HR based Organogram, is to create Designation under Departments. Same designation might be there under single department. So the table structure for the designation master with unlimited depth will be as following.

Field Name	Data Type
Designation ID	Integer (Auto Generate) [PK]
Designation Name	Varchar
Department	Integer

7.1.11. Employee Master:

This table contains information of Employees, who will operate the software. **This table will be the intermediate common table to be hooked up with HR Payroll Module**. The structure of this table is given below:

Field Name	Data Type	Description
Employee ID	Integer (Auto Generate) [PK]	
Employee Name	Varchar	
Permanent Address	Varchar	
Present Address	Varchar	
Contact No	Varchar	
Mail ID	Varchar	
NID No	Varchar	
TIN No	Varchar	
Blood Group	Varchar	
Marital Status	Boolean	
Marital Status	Varchar	
Father Name	Varchar	
Mother Name	Varchar	
Department	Integer	Linked with the table "**Department Master**"
Designation	Integer	Linked with the table "**Designation Master**"
Posting Branch	Integer	Linked with the table "**Branch Master**"
Appointment Date	Date	
Emergency Contact No	Varchar	
Bank Account No	Varchar	
Bank Name	Varchar	
Bank Branch	Varchar	

7.1.12. User Master:

Super Admin will create the user from the database of Employee, which will be there at HR Module. Details of Employee related tables will be mentioned later at HR Module. But this table contains User Credential for those Employees, who will operate the software. The structure of this table is given below:

Field Name	Data Type
User ID	Varchar [PK]
Password	Varchar
Employee ID	Integer

7.1.13. User Privilege:

This table will contain the Links against User ID, which will be permitted to access the sections of the software. The structure of this table is given below:

Field Name	Data Type
User ID	Varchar
URL	Varchar

7.1.14. Project Master of Policy:

The table "Project Master" of Policy will be used to categorize all the policy and office for better searching throughout the software. The proposed structure for this table is given below.

Field Name	Data Type
Project ID	Integer (Auto Generate) [PK]
Project	Varchar
Project Abbreviation	Varchar

7.1.15. Project Office Master of Policy:

The table "Project Office Master" of Policy will be used to categorize all the office with the project for better searching throughout the software. The proposed structure for this table is given below.

Field Name	Data Type
Project ID	Integer
Branch ID	Integer

7.1.16. Insurance Plan Master:

This table will contain the master plan of each policy against corresponding category. The proposed structure for this table is given below:

Field Name	Data Type	Description
Project	Integer	
Plan ID	Varchar [PK]	
Name of Policy	Varchar	
Minimum Premium No Payable to Get Full Maturity against death	Integer	Null for Child Policy
Matured Percentage for Death before Maturity	Real	Null for Child and Pension Policy
Child Policy Maturity Scheme	Boolean	If Yes, then Ref Table "**Child Policy Maturity Master**" as Ref Page 14; Other Wise Null
Pension Policy Maturity Scheme	Boolean	If Yes, then Ref Table "**Pension Policy Maturity Master**" as Ref Page 17; Other Wise Null
Pension Policy Premium	Boolean	If True, then Ref Table "**Pension Policy Premium Master**"; else "**Policy Premium Master**" as Ref Page 18
Accident Policy Premium	Boolean	If True, then Ref Table "**Accident Policy Premium Master**"; else "**Policy Premium Master**" as Ref Page 31
Stipend Allowed For Permanent Disability	Boolean	If True, then Ref Table "**Stipend For Permanent Disability for Accident Policy**" Ref Page 34
Is it DDB	Boolean	If True, then Ref Table "**Disease Master**". Ref Page 37
DDB Percentage	Real	If "**Is it DDB**" is True, then it is applicable. Otherwise it is NULL.
Adjust Premium if DDB Taken	Boolean	If "**Is it DDB**" is True, then it is applicable. Otherwise it is NULL.

Adjust Matured Amount if DDB Taken	Boolean	If "**Is it DDB**" is True, then it is applicable. Otherwise it is NULL.
Simple Maturity Scheme	Boolean	Applicable only for Policy Type as Ref Page 12
Complex Maturity Scheme	Boolean	Applicable only for Policy Type as Ref Page 13
Stipend for Education Policy	Boolean	Applicable only for Education Policy. Ref Page 23. If Yes Ref Table "**Stipend Master for Education Policy**" and "**Condition for Stipend of Education Policy**"
Default Tenure of Stipend	Real	Applicable only for Education Policy. Ref Page 23. For Other Value will be NULL
Compound Interest on First Year Stipend	Real	Applicable only for Education Policy. Ref Page 23. For Other Value will be NULL
Minimum Acceptable Age	Real	
Maximum Acceptable Age	Real	
Profit Allowed for Death Before Maturity	Boolean	
Associate Policy Allow	Boolean	
Is it Associate Plan	Boolean	If the Policy is Under Associate Policy
Exception Depending Upon Master Policy	Boolean	If Yes, then it will link with "**Exception of Associate for Master**".
Tenure Depends Upon Main Policy	Boolean	If Yes, Then the Max tenure will be as like as Mail Policy, as this is associate plan.
Discount Policy of Premium	Boolean	If Yes, it will relate with the table Premium Discount Policy
Minimum Insurance Amount	Currency	
Maximum Insurance Amount	Currency	
Medical test Required	Boolean	If True, Then Ref Table "**Medical Test Requirement Master**" and "**Policy Test Relation Master**" else NULL

No of days allowed for incomplete Proposal	Real	
Grace Period	Real	
Lapse Period for minimum Year of Renewals	Real	
Maximum Revival Period in Years	Real	
Percentage of Fine per year for Revival On Renewal	Real	
Extra Requirements for Lapse Policy Revival	Boolean	If True, the Ref Table "**Extra Requirements for Revival**" else NULL
Loan Percentage Allow	Real	

7.1.17. Child Policy Maturity Master:

This table will contain the master plan of each policy against corresponding category. The proposed structure for this table is given below:

Field Name	Data Type	Description
Plan ID	Varchar	
Maturity after Death before Maturity For Child Policy	Boolean	Applicable only for Child Policy. Related with the table Maturity after Death before Maturity For Child Policy as Ref Page 14
Stipend Allow After Death of Premium Provider and Death of Child before Maturity	Boolean	Applicable only for Child Policy as Ref Page 14. True for Child Policy. Else Null
Matured Percentage for Death of Premium Provider and Death of Child before Maturity	Real	Applicable only for Child Policy as Ref Page 14. For Else NULL
Stipend Percentage Per Month	Real	Applicable only for Child Policy as Ref Page 14

7.1.18. Expiration Master of Policy:

Duration of year for expiry against every policy is different. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Plan ID	Varchar	
Year of Expiration Allowed	Real	Year of Expiration of Policy Premium, Pension Policy Premium, and Accident Policy Premium will depend upon this field.
Age Allowed at Expiry	Real	For Child Policy (Ref Page : 14)

7.1.19. Simple Maturity Master of Policy: (For Policy Type as Ref Page 12)

As some policy return 100% sum assured amount with profit after the expiration of it and some policy return on partial basis with partial percentage, so it is necessary to create this table. The proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Plan ID	Varchar	
Percentage Year of Expiration	Real	
Return Percentage	Real	
Return Percentage to Nominee if Policy Holder Dead	Real	Applicable for the Policy at Page 28
Return Premium Only if Matured	Boolean	Applicable for the Policy at Page 26

7.1.20. Complex Maturity Master of Policy: (For Policy Type as Ref Page 13)

As some policy returns some percentage after certain year, by dividing it into certain hopping period and rest of the amount of sum assured at the time of maturity with profit after the expiration of it, so it is necessary to create this table. The proposed structure of corresponding table is given below:

Field Name	Data Type
Plan ID	Varchar
Return From the Year	Real
Net Return Percentage Before Maturity	Real
Hopping Period	Integer

7.1.21. Complex Maturity Slab of Policy: (Ref Page 13)

This table will contain the Slab of Complex Maturity for those policies, which is under Complex Maturity Master. The rest percentage after the Net Return before Maturity will be applied after the expiry. The proposed structure of corresponding table is given below:

Field Name	Data Type
Plan ID	Varchar
Return At Year	Real
Return Percentage	Real

7.1.22. Maturity after Death before Maturity For Child Policy:

This table will contain the Slab of Maturity after Death before Maturity for Child Policy. The proposed structure of corresponding table is given below:

Field Name	Data Type
Plan ID	Varchar
From Length of Policy	Real
To Length of Policy	Real
Maturity Percentage	Real

7.1.23. Premium Frequency Master:

This table will contain Slab of Premium against the Local Currency of Base Assured Amount/ Stipend Amount of Local Currency of Base Assured Amount /Year for Education Policy. The proposed structure of this table is given below.

Field Name	Data Type	Description
Plan ID	Varchar	Applicable only for Plan based Premium frequency, else NULL
Frequency ID	Integer	
Frequency	Real	Monthly, Yearly, Half Yearly or Quarterly.
Default Premium Frequency	Real	If policy holder not paying yearly, consider it if dead during partial period of Default Frequency

7.1.24. Policy Premium Master:

This table will contain Slab of Premium against the Local Currency of Base Assured Amount/ Stipend Amount of Local Currency of Base Assured Amount /Year for Education Policy. The proposed structure of this table is given below.

Field Name	Data Type	Description
Plan ID	Varchar	
Base Assured Amount	Currency	
Age	Real	
Year of Expiration	Real	(Value allowed, depending upon the value of the field “Year of Expiration Allowed” under the table “Expiration master of Policy” against the “Plan ID”.)
Premium Amount	Currency	
Premium Frequency	Integer	Monthly, Yearly, Half Yearly or Quarterly. Field Linked with Premium Frequency Master.

		If NULL, then it will depend upon the choice of Policy Holder.
Unit Premium Type	Boolean	If True, Then Unit Premium; Otherwise it will depends upon premium table. Ref Page 20

7.1.25. Accident Policy Premium:

This table will contain Slab of Premium against the Local Currency of Base Assured Amount. The proposed structure of this table is given below.

Field Name	Data Type	Description
Plan ID	Varchar	
Base Assured Amount	Currency	
Class of Occupation	Integer	This field will fetch Ref Data from the Table "**Class of Occupation Master**"
Premium Amount For Accidental Death	Currency	
Premium Amount For Accidental Disability	Currency	
Maximum Age for Male For Accidental Death	Real	
Maximum Age for Female For Accidental Death	Real	
Maximum Age for Male For Accidental Disability	Real	
Maximum Age for Female For Accidental Disability	Real	
Allow to Surrender The Policy after Premium Date	Boolean	
No of Days Allow to Surrender The Policy after Premium Date	Real	If "Allow to Surrender The Policy after Premium Date"=True, then allow; else NULL. "0" for No Limit.
No Premium if Permanent Disable	Boolean	
Month of Continuous Disability	Real	If "No Premium if Permanent Disable"=True, then applicable otherwise NULL.

7.1.26. Pension Policy Premium Master:

This table will contain Slab of Premium against the assured amount with Pension Amount of Local Currency 1000/Year for Pension Policy. The proposed structure of this table is given below.

Field Name	Data Type
Plan ID	Integer
Age	Real
Age of Pension Start	Real
Premium Amount	Currency

7.1.27. Pension Maturity Master:

This table will contain Terms and Condition after Maturity of Pension Plan. The proposed structure of this table is given below.

Field Name	Data Type
Plan ID	Varchar
Minimum Pension Years	Real
Multiple of Pension Eligible for Nominee If Policy Holder Dead	Real
Salable %age of Yearly Pension	Real
Allowable No of Years to Sale the Pension	Real
Pension Depends Upon Paid Premium	Boolean

7.1.28. Premium Discount Policy :

Discount against the Premium will be applied as pre the frequency of Policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Plan ID	Varchar	
Premium Frequency	Real	
Discount Type	Varchar	Default Value Fixed/Multiply/Percentage
Discount Amount	Real	

7.1.29. Stipend Master for Education Policy :

This table will contain Stipend Logic for Education Policy. It will be calculated by multiple of stipend amount 10000 for 1st Year. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Plan ID	Varchar
Effective Date	Date
Year of Stipend	Real
Principle Stipend Amount	Currency
Tenure of Policy	Real
Bonus Amount	Currency
Disbursement Period	Real

7.1.30. Condition for Stipend of Education Policy :

This table will contain Special Logic of Premium Calculation for Education Policy, Depending upon the Tenure of Stipend. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Plan ID	Varchar
Base Premium Amount for Calculation	Currency
Minimum Stipend Tenure	Real
Maximum Stipend Tenure	Real
Percentage for Multiple of Tenure for Premium Amount	Real
Stipend Amount After Death of Premium Provider for Stipend of 10000 for 1st Year	Currency
Net Amount for pre surrender after paying Minimum premium for Stipend of 10000 for 1st Year	Currency
Base Amount Percentage of stipend for last quarter for stipend surrender	Real
Transfer of Stipend Allow	Boolean

7.1.31. Exception of Associate for Master :

This table will contain Special Logic of Maximum Insured Amount depending against which Main Policy taken. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Plan ID	Varchar
Main Plan ID	Varchar
Maximum Insured Amount Allowed	Currency

7.1.32. Accident Master :

This table will contain type of Accident allowable or not allowable. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Accident ID	Integer (Auto Generate) [PK]	
Type of Accident	Varchar	
Allowed For Death	Boolean	Yes, if allowed; No if not allowed.
Allowed For Disability	Boolean	Yes, if allowed; No if not allowed.
Days Limit of Claim	Real	

7.1.33. Accident Document Master :

This table will contain the documents required as General or the proof of Accident allowable. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Accident ID	Integer	
Document ID	Integer	Linked with the table "**Document Master**"

7.1.34. Document Master :

This table will contain the documents required as General or the proof of Accident allowable. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Document ID	Integer (Auto Generate) [PK]
Type of Documents	Varchar

7.1.35. Sub-Document Master :

This table will contain the documents required as General or the proof of Accident allowable. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Sub Document ID	Integer (Auto Generate) [PK]
Type of Documents	Varchar

7.1.36. Document Sub-Document Relation Master :

This table will contain the documents and Sub-Document relation required as General or the proof of Accident allowable. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Document ID	Integer	Linked with the table " **Document Master**"
Sub Document ID	Integer	Linked with the table "**Sub-Document Master**"

7.1.37. Class of Occupation Master :

This table will contain the classification as per occupation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Occupation	Integer	This field will be linked with "**Occupation Master**"
Class	Integer	

7.1.38. Occupation Restriction Master:

This table will contain Occupation wise restrictions. So the table structure for this master will be as like as following:

Field Name	Data Type	Description
Occupation Category	Integer	Linked with Occupation Category Master
Occupation	Integer	Linked With Occupation Master
Extra Premium for Base Amount	Currency	
Allow Permanent Disability Scheme	Boolean	
Extra Premium for Permanent Disability Scheme	Currency	
Allow Accident Policy	Boolean	
Extra Premium for Accident Policy	Currency	
Allow Other	Boolean	
Extra Premium For Other	Currency	

7.1.39. Disability Schedule :

This table will contain the classification of disability after accident and for this cause percentage of return. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Disability ID	Integer (Auto Generate) [PK]
Category of Disability	Varchar
Sub Category of Disability	Varchar
Percentage of Return	Real
Completely Disable	Boolean
Allowable for Premium Redemption	Boolean
Allowable for Insurance	Boolean

7.1.40. Stipend For Permanent Disability for Accident Policy :

This table will contain the stipend details due to permanent Disability for Accident Policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type

Plan ID	Varchar
Percentage of Insurance for Stipend	Real
Stipend Frequency	Real
No of Years For Stipend	Real

7.1.41. Disease Master :

This table will contain type of Disease allowable or not allowable for DDB. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Disease ID	Integer (Auto Generate) [PK]	
Disease	Varchar	

7.1.42. Medical Test Requirement Master :

This table will contain Medical Test requirements against policy. So the proposed structure of corresponding table is given below:

Field Name	**Data Type**	**Description**
Test Logic ID	Integer (Auto Generate) [PK]	
Insured Amount From	Currency	
Insured Amount To	Currency	
From Age	Real	
To Age	Real	
Test Required	Integer	Linked with the Tale "**Medical Test Master**" by "**Test ID**"

7.1.43. Medical Test Master :

This table will contain List of Medical Tests. So the proposed structure of corresponding table is given below:

Field Name	**Data Type**
Test ID	Integer (Auto Generate) [PK]
Test Name	Varchar
Abbreviation	Varchar
Month of Validity	Real

7.1.44. Policy Test Relation Master :

This table will contain List of Medical Tests or Test Logic Requirements required Against Policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Plan ID	Varchar
Test ID	Integer
Test Logic ID	Integer
Maximum Insured Amount	Currency

7.1.45. Policy Occupation Relation Master :

This table will contain List of Medical Tests or Test Logic Requirements required Against Policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Plan ID	Varchar	
Gender	Varchar	
Occupation ID	Integer	
Female Type ID	Integer	Applicable, if Gender is Female
Maximum Age of Proposal	Real	
Maximum Age of Maturity	Real	
Maximum Insurance	Currency	
Document ID	Integer	If Document Required, Otherwise NULL
Extra Payable Percentage on Premium	Real	
Illiterate	Boolean	
Lien Option Percentage	Real	
Medical Certificate Required	Boolean	
Type of Doctor	Integer	

7.1.46. Female Type Master :

This table will contain List of Medical Tests or Test Logic Requirements required Against Policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Female Type ID	Integer (Auto Generate) [PK]
Female Type	Varchar

7.1.47. Doctor Type Master :

This table will contain Type of Doctors allowed. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Doctor Type ID	Integer (Auto Generate) [PK]	
Doctor Type	Varchar	Govt. Doctor/Civil Surgeon

7.1.48. Extra Requirements for Revival :

This table will contain extra requirements to revive the lapsed policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Lapse Period From	Real	
Lapse Period To	Real	
Age From at the Time of Revival	Real	
Age To at the Time of Revival	Real	
Insured Amount From	Currency	
Insured Amount To	Currency	
Test ID	Integer	Linked with the table "**Medical Test Master**"
Transfer To HO	Boolean	If True, Then case should be under scope of Head Office

7.1.49. Charge of Modification :

This table will contain extra requirements to revive the lapsed policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Modification Type ID	Integer (Auto Generate) [PK]
Modification Type	Varchar
Charges	Currency

7.1.50. Stamp Fee Master :

This table will contain stamp fee assigned by Govt. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Stamp Fee ID	Integer (Auto Generate) [PK]
WEF	Date
Base Amount	Currency

Stamp Fee	Currency

7.1.51. Duplicate Policy Requirements Master :

This table will contain Commercial Requirements for Duplicate Policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
WEF	Date	
Stamp Fee Required	Boolean	Field will be linked with the table "**Stamp Fee Master**"
Other Official Fees Required	Boolean	Field will be Linked With the Table "**Duplicate Policy Fee Master**"

7.1.52. Duplicate Policy Fee Master :

This table will contain stamp fee assigned by Govt. So the proposed structure of corresponding table is given below:

Field Name	Data Type
WEF	Date
Head	Varchar
Fee	Currency

7.1.53. Agent Commission Master :

This table will contain plan wise agent commission structure. So the proposed structure of corresponding table is given below:

Field Name	Data Type
WEF	Date
Plan ID	Varchar
Agent Category ID	Integer
1st Commission	Real
Renewal Commission	Real

7.1.54. Agent :

This table will contain plan wise agent commission structure. So the proposed structure of corresponding table is given below. Agent ID will be Centralized except FA:

Field Name	Data Type	Description
Joining Date	Date	
Agent Category ID	Integer	
Agent ID	Integer (Auto Generate) [PK]	
Agent Name	Varchar	

Agent Name in Bengali	Varchar	
Senior ID	Integer	
Address	Varchar	
Address in Bengali	Varchar	
Division	Integer	Linked with the Table "**Division Master**"
District	Integer	Linked with the table "**District Master**"
Upazila	Integer	Linked with the table "**Upazila Master**"
PIN Code	Varchar	Linked with the table "**Pin Code Master**"
DOB	Date	
Contact No	Varchar	
Mail ID	Varchar	
NID No	Varchar	
Lsc No	Varchar	
Lsc Expiry	Date	
Bank Name	Varchar	
AC No	Varchar	
Bank Branch	Varchar	
Mobile Bank Account No	Varchar	
Mobile Account Name	Varchar	
Status	Boolean	If "True", then Active; If "False", then "Inactive".
WEF	Date	

7.1.55. Customer :

This table will contain the personal information required for Customer as per Proposal. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Customer ID	Integer (Auto Generate) [PK]	
Signup Date	Date	
Name	Varchar	
Name in Bengali	Varchar	
Father Name	Varchar	
Mother Name	Varchar	
DOB	Date	
Birth Place	Varchar	
Occupation	Integer	Linked with the Table "Occupation Master"

Employer	Varchar	
Academic Qualification	Integer	Linked with the Table "Academic Qualification Master"
Permanent Address	Varchar	
Permanent Address in Bengali		
Present Address	Varchar	
Present Address in Bengali	Varchar	
Division	Integer	Linked with the Table "**Division Master**"
District	Integer	Linked with the table "**District Master**"
Upazila	Integer	Linked with the table "**Upazila Master**"
PIN Code	Varchar	Linked with the table "**Pin Code Master**"
Photo	Binary/Varchar (URI/Path of the Image)	
Mobile No	Varchar	
Nationality	Varchar (Bangladesh Default)	
TIN No	Varchar	
NID No	Varchar	
Bank	Varchar	
Bank Account No	Varchar	
Bank Branch	Varchar	
Mobile Bank Account No	Varchar	
Mobile Account Name	Varchar	

7.1.55. Division Master :

This table will contain the list of Division for address. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Division ID	Integer (Auto Generate) [PK]	
Division	Varchar	

7.1.56. District Master :

This table will contain the list of District as per Division for address. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description

District ID	Integer (Auto Generate) [PK]	
District	Varchar	
Division	Integer	Linked with the Table "**Division Master**"

7.1.57. Upazila Master :

This table will contain the list of Upazila as per District for address. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Upazila ID	Integer (Auto Generate) [PK]	
Upazila	Varchar	
District	Integer	Linked with the table "**District Master**"

7.1.58. Pin Code Master :

This table will contain the list of Pin Code as per Upazila for address. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Pin Code	Integer [PK]	
Upazila	Integer	Linked with the table "**Upazila Master**"

7.1.59. Proposal Info :

This table will contain the Policy information required for Customer as per Proposal. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Proposal No	Varchar [PK]	
Ref No	Varchar	Given by Branch
Customer ID	Integer	Linked with the Table "Customer"
Age	Real	
Agent ID	Integer	Linked with the table "Agent"
Branch ID	Integer	Linked with the table "Branch Master"
Application Date	Date	
Plan ID	Varchar	Linked with the table "Insurance Policy Master"
Tenure	Real	
Insured Amount	Money	
Annual Pension	Money	

Unit	Real	
Is it Child Policy	Boolean	
Name of Child	Varchar	
DOB of Child	Date	
Age of Child	Real	
Associate Plan ID	Varchar	
Premium	Money	
Premium Payment Frequency	Integer	Linked with the table "**Premium Frequency Master**"
Status	Varchar	
Bank	Varchar	
Bank Account No	Varchar	
Bank Branch	Varchar	
Mobile Bank Account No	Varchar	
Mobile Account Name	Varchar	

7.1.60. Proposal Nominee Info :

This table will contain the nominee information against policy required for Customer as per Proposal. So the proposed structure of corresponding table is given below:

Field Name	**Data Type**	**Description**
Proposal No	Varchar	Linked with the table "Proposal Info"
Priority	Real	
Nominee Name	Varchar	
Nominee Relation	Integer	Linked with the table "**Master for Relation of Nominee**"
Nominee DOB	Date	
Nominee Age	Real	
Is he Guardian of Nominee	Boolean	
Is he Alternate Nominee	Boolean	
Photo of Nominee	Binary/Varchar (URI/Path of the Image)	
ID Proof Certificate No	Varchar	
ID Proof Document of Nominee	Binary/Varchar (URI/Path of the Image)	

7.1.61. Policy Based Document Master :

This table will contain the documents required as per plan type. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Plan ID	Varchar	Linked with the table “Insurance Policy Master”
Document ID	Integer	Linked with “Document Master”
Sub Document ID	Integer	Linked with “Sub Document Master”

7.1.62. Proposal Document Info :

This table will contain the document information against policy required for Customer as per Proposal. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Proposal No	Varchar	Linked with the table “Proposal Info”
Plan ID	Varchar	Linked with the table “Proposal Info”
Document ID	Integer	Linked with “Document Master”
Sub Document ID	Integer	Linked with “Sub Document Master”
Document	Binary/Varchar (URI/Path of the Image)	

7.1.63. Policy Info :

This table will contain the Policy information required for Customer as per Proposal after underwritten. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Policy No	Varchar [PK]	
Proposal No	Varchar	Linked with the table “Proposal Info”
Ref No	Varchar	Given by Branch
Customer ID	Integer	Linked with the Table “Customer”
Age	Real	
Agent ID	Integer	Linked with the table “Agent”
Branch ID	Integer	Linked with the table “Branch Master”
Application Date	Date	
Signup Date		
Plan ID	Varchar	Linked with the table “Insurance Policy Master”
Tenure	Real	
Insured Amount	Money	
Annual Pension	Money	
Unit	Real	
Is it Child Policy	Boolean	

Name of Child	Varchar	
DOB of Child	Date	
Age of Child	Real	
Premium	Money	
Premium Payment Frequency	Integer	Linked with the table "**Premium Frequency Master**"
Age Admitted	Boolean	
Required Stamp Category	Integer	Linked with "Stamp Category Master"
Required Stamp Amount	Money	
Original Receipt No	Integer	
Status	Varchar	
Document ID for Age Proof	Integer	Linked with "Document Master" and "Sub-Document Master"
Status Date	Date	
Next Due Date	Date	
Risk Date	Date	
Bank	Varchar	
Bank Account No	Varchar	
Bank Branch	Varchar	
Mobile Bank Account No	Varchar	
Mobile Account Name	Varchar	
Tabara	Currency	
Agent Bill No	Varchar	

7.1.64. Associated Proposal Info :

This table will contain the Associated Policy information required for Customer as per Proposal. So the proposed structure of corresponding table is given below:

Field Name	**Data Type**	**Description**
Proposal No	Varchar (PK)	
Main Proposal No	Varchar	Linked With the Table **"Proposal Info"**
Ref No	Varchar	Given By Branch
Customer ID	Integer	Linked with the Table **"Customer"**
Associated Plan ID	Varchar	Linked with the Table **"Insurance Policy Master"**
Premium	Money	

Premium Payment Frequency	Integer	Linked with the Table "**Premium Frequency Master**"
Status	Varchar	
Tenure	Real	
Insured Amount	Money	

7.1.65. Associated Policy Info :

This table will contain the Associated Policy information required for Customer as per Proposal after underwritten. So the proposed structure of corresponding table is given below:

Field Name	**Data Type**	**Description**
Main Policy No	Varchar	Linked With the Table **"Policy Info"** and table **"Associated Proposal Info"**
Policy No	Varchar [PK]	
Proposal No	Varchar	Linked with the Table **"Associated Proposal Info"**
Ref No	Varchar	Linked with the Table **"Associated Proposal Info"**
Customer ID	Integer	Linked with the Table **"Customer"**
Associated Plan ID	Varchar	Linked with the Table **"Insurance Policy Master"**
Premium	Money	
Premium Payment Frequency	Integer	Linked with the Table "**Premium Frequency Master**"
Tenure	Real	
Insured Amount	Money	

7.1.66. Extra Premium :

This table will contain the Extra Premium Information against policy after underwritten. So the proposed structure of corresponding table is given below:

Field Name	**Data Type**	**Description**
Policy No	Varchar	Linked With the Table **"Policy Info"**
Extra Premium Type	Varchar	
Extra Premium	Currency	

7.1.67. Collection Info :

This table will contain the Payment Collection Information against policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type
PR No	Integer (Auto generate) (Pk)
Date	Date
Policy No	Varchar
Type	Varchar
Amount	Currency

7.1.68. PR Info :

This table will contain the PR Information against policy. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Date	Date	
Policy No	Varchar	
Fine	Currency	
Amount	Currency	
Branch	Integer	Linked With the Table **"Branch Master"**
PR No	Integer	Linked With the Table **"Collection Info"**
Effective Date	Date	
DCS No	Integer	Linked With the Table **"DCS Info"**

7.1.69. DCS Info :

This table will contain DCS Information. So the proposed structure of corresponding table is given below:

Field Name	Data Type
DCS Date	Date
DCS No	Integer (Auto generate) (Pk)
Amount	Currency

7.1.70. DCS Deposit Info :

This table will contain DCS Information. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Deposit Date	Date	
Paying Slip No	Varchar	
Amount	Currency	
DCS No	Integer	Linked With the Table **"DCS Info"**

7.1.71. Policy Nominee Info :

This table will contain the nominee information against policy required for Customer as per Proposal after underwritten. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Policy No	Varchar	Linked with the table "Policy Info"
Proposal No	Varchar	Linked with the table "Proposal Info"
Priority	Real	
Nominee Name	Varchar	
Nominee Relation	Integer	Linked with the table "**Master for Relation of Nominee**"
Nominee DOB	Date	
Nominee Age	Real	
Is he Guardian of Nominee	Boolean	
Is he Alternate Nominee	Boolean	
Photo of Nominee	Binary/Varchar (URI/Path of the Image)	
ID No	Varchar	
ID Proof Document of Nominee	Binary/Varchar (URI/Path of the Image)	

7.1.72. Policy Document Info :

This table will contain the document information against policy required for Customer as per Proposal after underwritten. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Policy No	Varchar	Linked with the table "Policy Info"
Proposal No	Varchar	Linked with the table "Proposal Info"
Plan ID	Varchar	Linked with the table "Proposal Info"
Document ID	Integer	Linked with "Document Master"
Sub Document ID	Integer	Linked with "Sub Document Master"
Document	Binary/Varchar (URI/Path of the Image)	

7.1.73. Policy Renewal Info :

This table will contain the renewal information against policy required for Customer as per Proposal after underwritten. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Policy No	Varchar	Linked with the table “Policy Info”
Installment Date	Date	
Installment No	Real	
Installment Amount	Currency	
Payment Date	Date	
Penalty	Currency	
Tx No	Integer (Auto Generate) [PK]	
Branch ID	Integer	
Receipt Amount	Currency	
Agent ID	Integer	

7.1.74. Policy Commission Info :

This table will contain the Gap and Spot commission information for Agent as per Proposal after underwritten. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Policy No	Varchar	Linked with the table “Policy Info”
Proposal No	Varchar	Linked with the table “Proposal Info”
Plan ID	Varchar	Linked with the table “Proposal Info”
Agent ID	Integer	Linked with “Agent”
Installment No	Real	
Installment Date	Date	
Payment Date	Date	
Commission Percentage	Real	Linked with “Agent Commission Master”
Commission Amount	Money	
Commission Payment Date	Date	
Original Receipt No	Integer	
Branch ID	Integer	

7.1.75. Chart of Accounts :

This table will contain the Chart of Accounts to manage accounting module. **This table will be the intermediate common table to be hooked up with HR Payroll Module**. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Account ID	Integer (Auto Generate) [PK]
Account Head	Varchar
Parent ID	Integer

7.1.76. Accounting Voucher :

This table will contain the All Accounting Voucher to manage accounting transactions. **This table will be the intermediate common table to be hooked up with Accounting Module**. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Voucher No	Integer (Auto Generate) [PK]	
Voucher Type	Integer	Linked with "Voucher Type Master"
Voucher Date	Date	
Account Head	Integer	Linked with the table "**Chart of Accounts**"
Debit	Currency	
Credit	Currency	
Note	Varchar	
Ref No	Varchar	
Branch ID	Integer	Linked with "Branch Master"

7.1.77. Stamp Category Master :

This table will contain the all types of stamp category required for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Category ID	Integer (Auto Generate) [PK]
Stamp Category	Integer

7.1.78. Stamp Master :

This table will contain the available stamp as per the corresponding category required for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Stamp ID	Integer (Auto Generate) [PK]	
Stamp Category	Integer	Linked with the table of "**Stamp Category Master**"
Value of Stamp	Currency	
Date of Opening	Date	
Opening Quantity	Real	
Opening Value	Currency	

7.1.79. Stamp Requisition From Branch :

This table will contain the data related to the requisition of stamp as per the corresponding category required for Operation from Branch. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Requisition ID	Integer (Auto Generate) [PK]	
Stamp ID	Integer	Linked with the table of "**Stamp Master**"
Date	Date	
Branch	Integer	Linked with the table of "**Branch Master**"
Quantity	Real	
Stamp Value	Currency	
Total Value	Currency	

7.1.80. Stamp Requisition For Policy :

This table will contain the data related to the requisition of stamp as per the corresponding category required for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Requisition ID	Integer (Auto Generate) [PK]	
Stamp ID	Integer	Linked with the table of "**Stamp Master**"
Date	Date	
From Branch	Integer	Linked with the table of "**Branch Master**"

Policy No	Varchar	Linked with the table of "**Policy Info**"
Stamp Value	Currency	
Quantity	Real	
Total Value	Currency	

7.1.81. Stamp Purchase :

This table will contain the data related to the purchase of stamp as per the corresponding category required for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Purchase ID	Integer (Auto Generate) [PK]	
Stamp ID	Integer	Linked with the table of "**Stamp Master**"
Date	Date	
Branch	Integer	Linked with the table of "**Branch Master**"
Quantity	Real	
Unit Value	Currency	
Other Charge	Currency	
Total	Currency	
Vendor	Integer	Linked with the table "Vendor Master"

7.1.82. Stamp Transfer :

This table will contain the data related to the transfer of stamp as per the corresponding category required for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Tx ID	Integer (Auto Generate) [PK]	
Stamp ID	Integer	Linked with the table of "**Stamp Master**"
Date	Date	
From Branch	Integer	Linked with the table of "**Branch Master**"
To Branch	Integer	Linked with the table of "**Branch Master**"
Quantity	Real	
Stamp Value	Currency	

Total Value	Currency	

7.1.83. Stamp Issue :

This table will contain the data related to the issue of stamp against policy as per the corresponding category required for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Tx ID	Integer (Auto Generate) [PK]	
Stamp ID	Integer	Linked with the table of "**Stamp Master**"
Date	Date	
From Branch	Integer	Linked with the table of "**Branch Master**"
Policy No	Varchar	Linked with the table of "**Policy Info**"
Quantity	Real	
Stamp Value	Currency	
Total Value	Currency	
Value bared by Company	Boolean	

7.1.84. Item Category Master :

This table will contain the all types of items category required to procure for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Category ID	Integer (Auto Generate) [PK]
Category	Integer

7.1.85. Vendor Master :

This table will contain the details information of Vendor. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Vendor ID	Integer (Auto Generate) [PK]
Vendor Name	Varchar
Mail ID	Varchar
Mobile No	Varchar
Address	Varchar
Opening Value	Currency

7.1.86. Item Master :

This table will contain the available items as per the corresponding category required to procure for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Item ID	Integer (Auto Generate) [PK]	
Item Category	Integer	Linked with the table of "**Item Category Master**"
Avg Value of Purchase	Currency	
Date of Opening	Date	
Opening Quantity	Real	
Opening Value	Currency	

7.1.87. Item Requisition From Branch :

This table will contain the data related to the requisition of items as per the corresponding category required to procure for Operation from Branch. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Requisition ID	Integer (Auto Generate) [PK]	
Item ID	Integer	Linked with the table of "**Item Master**"
Date	Date	
Branch	Integer	Linked with the table of "**Branch Master**"
Quantity	Real	

7.1.88. Item Requisition From Department :

This table will contain the data related to the requisition of items as per the corresponding category required to procure for Operation from Department. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Requisition ID	Integer (Auto Generate) [PK]	
Item ID	Integer	Linked with the table of "**Item Master**"
Date	Date	
Branch	Integer	Linked with the table of "**Branch Master**"
Department	Integer	Linked with the table of "**Department Master**"

Quantity	Real	

7.1.89. Item Purchase :

This table will contain the data related to the purchase of items as per the corresponding category required to procure for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Purchase ID	Integer (Auto Generate) [PK]	
Item ID	Integer	Linked with the table of "**Item Master**"
Date	Date	
Branch	Integer	Linked with the table of "**Branch Master**"
Quantity	Real	
Unit Value	Currency	
Other Charge	Currency	
Total	Currency	
Vendor	Integer	Linked with the table "Vendor Master"
Requisition From	Varchar	
Requisition ID	Integer	Linked with the table "**Item Requisition From Branch**" or "**Item Requisition From Department**"
Approval No	Integer	

7.1.90. Item Transfer :

This table will contain the data related to the transfer of items as per the corresponding category required to procure for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Tx ID	Integer (Auto Generate) [PK]	
Item ID	Integer	Linked with the table of "**Item Master**"
Date	Date	
From Branch	Integer	Linked with the table of "**Branch Master**"
To Branch	Integer	Linked with the table of "**Branch Master**"
Quantity	Real	

7.1.91. Item Issue :

This table will contain the data related to the issue of item against policy as per the corresponding category required to procure for Operation. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Tx ID	Integer (Auto Generate) [PK]	
Item ID	Integer	Linked with the table of "**Item Master**"
Date	Date	
From Branch	Integer	Linked with the table of "**Branch Master**"
Department ID	Integer	Linked with the table of "**Department Master**"
Quantity	Real	

7.1.92. Change Request :

This table will contain the track record related with any sorts of Modification. So the proposed structure of corresponding table is given below:

Field Name	Data Type
Request Ticket ID	Integer (Auto Generate) [PK]
Request User	Varchar
Request	Varchar
Date of Request	DateTime
Approval Status	Varchar
Approval Date	Date
Remarks	Varchar

7.1.93. Cash Transfer From Branch To Branch :

This table will contain the track record related with any sorts of cash Transfer from one branch to another. This will create Internal Journal Voucher at the accounts module. So the proposed structure of corresponding table is given below:

Field Name	Data Type	Description
Tx ID	Integer (Auto Generate) [PK]	
Date	Date	
Amount	Currency	
From Branch	Integer	
To Branch	Integer	

Remarks	Varchar	
JV No	Varchar	Linked with the "Voucher No" of the table "Accounting Voucher"

7.1.94. Cash Transfer From Branch To Bank :

This table will contain the track record related with any sorts of cash Transfer from one branch to any Bank Account. This will create Internal Journal Voucher at the accounts module. So the proposed structure of corresponding table is given below:

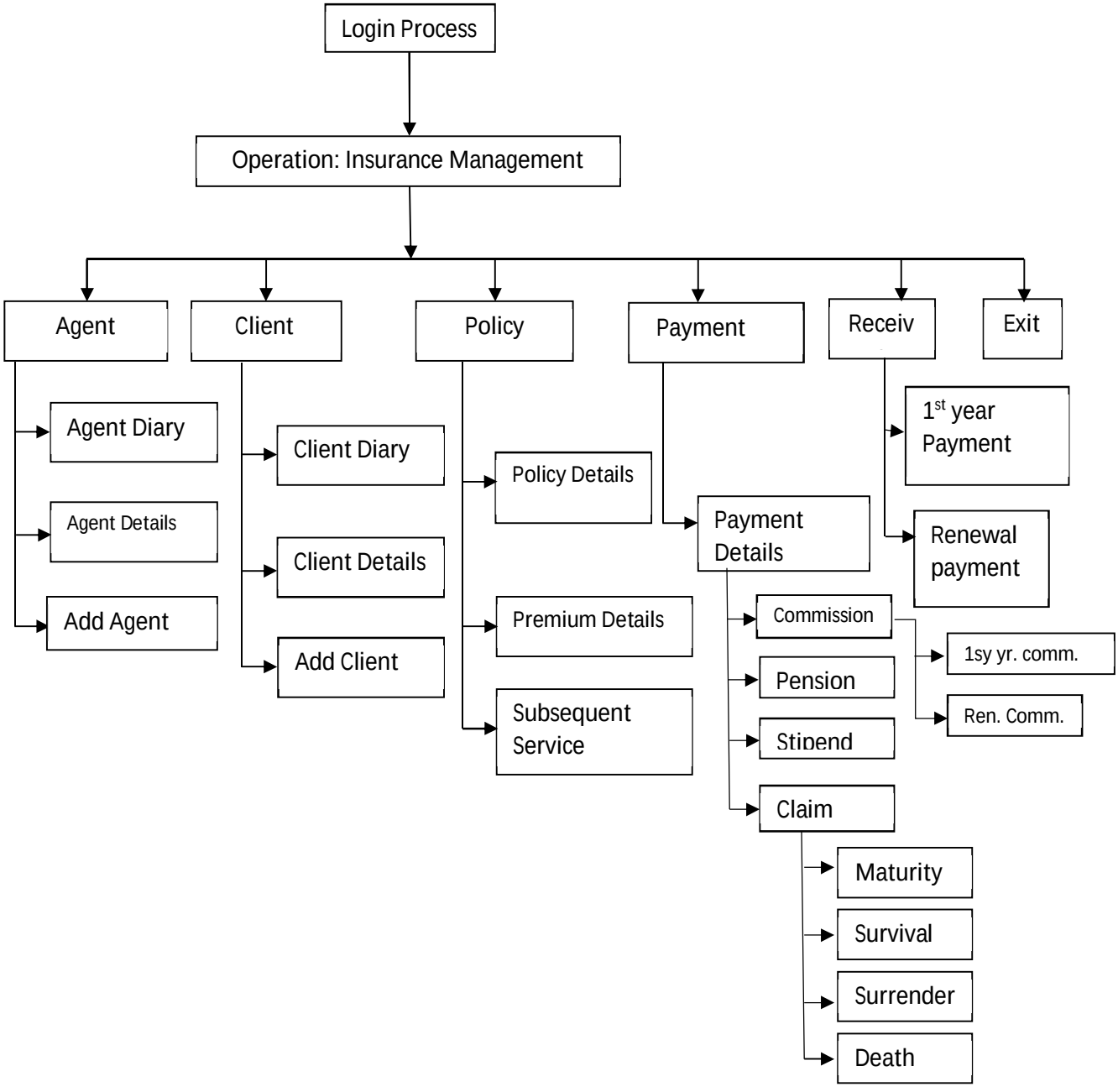
Field Name	**Data Type**	**Description**
Tx ID	Integer (Auto Generate) [PK]	
Date	Date	
Amount	Currency	
From Branch	Integer	
Bank	Integer	Linked with the table "**Bank Master**"
Remarks	Varchar	
JV No	Varchar	Linked with the "Voucher No" of the table "Accounting Voucher"

7.1.95. Bank Master :

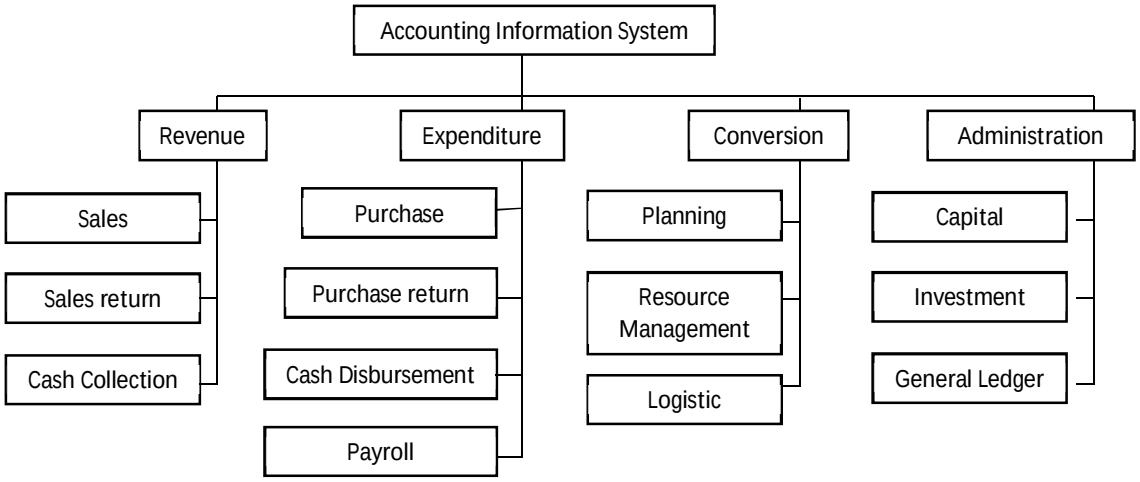
This table will contain the list of Bank Accounts. So the proposed structure of corresponding table is given below:

Field Name	**Data Type**	
Bank ID	Integer (Auto Generate) [PK]	
Bank	Varchar	
Account No	Varchar	
Branch	Integer	

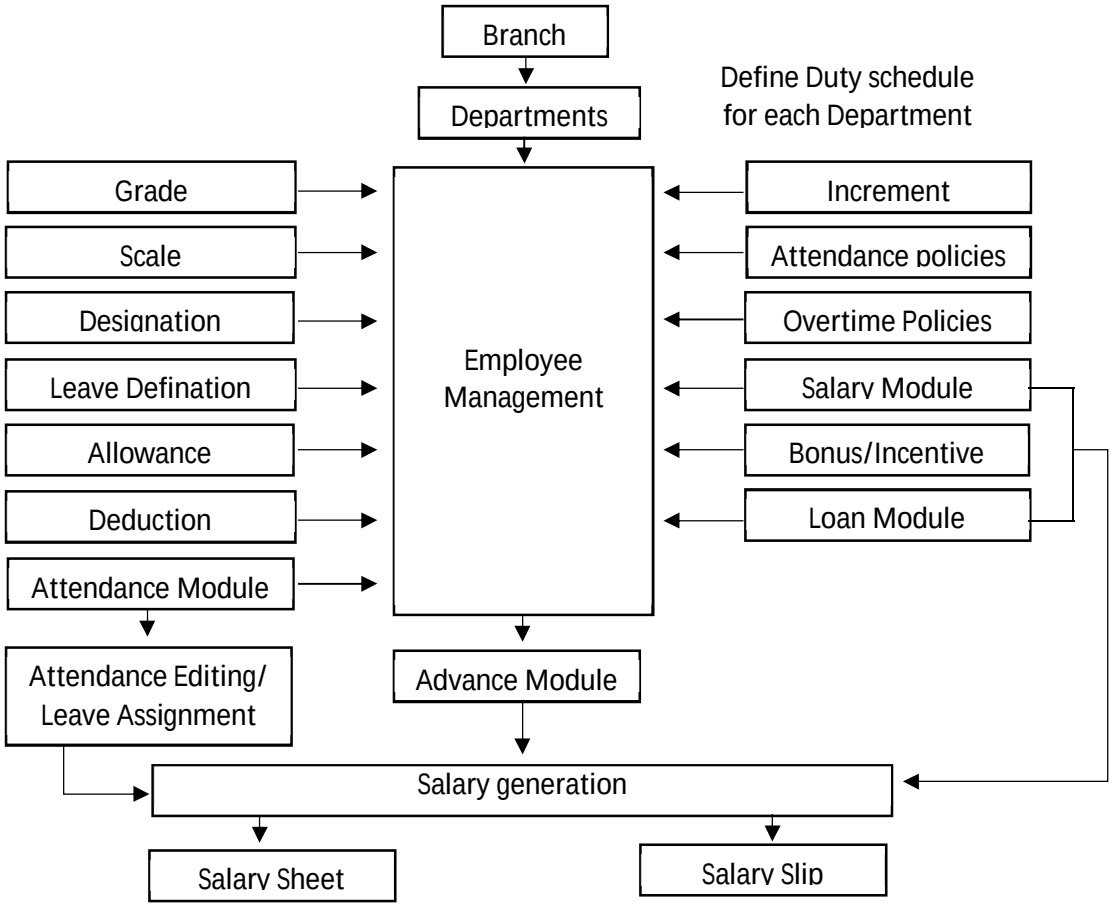
7.2. System Flow Diagram for the Operation module of Insurance Management System



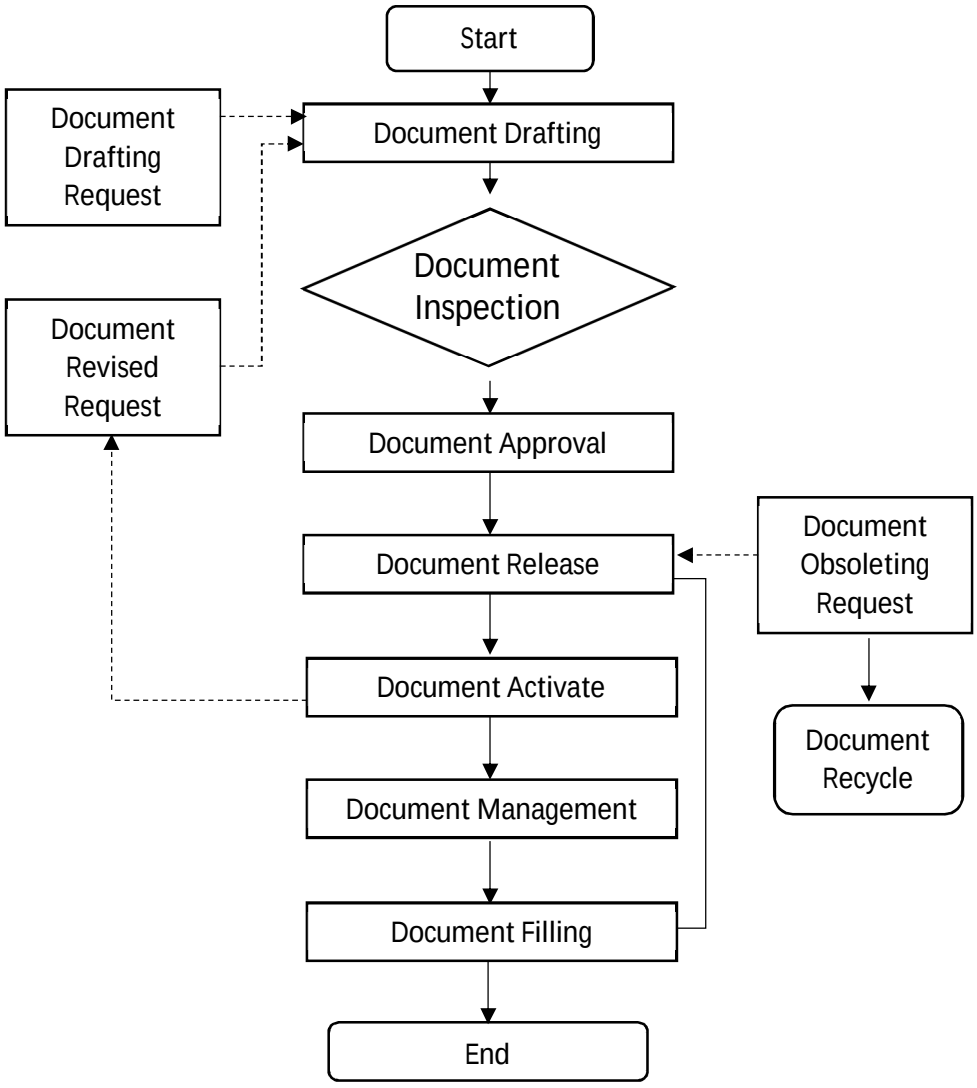
7.3. System Flow Diagram for the Accounting Information System Module



7.4. System Flow Diagram for the HR & Payroll Module

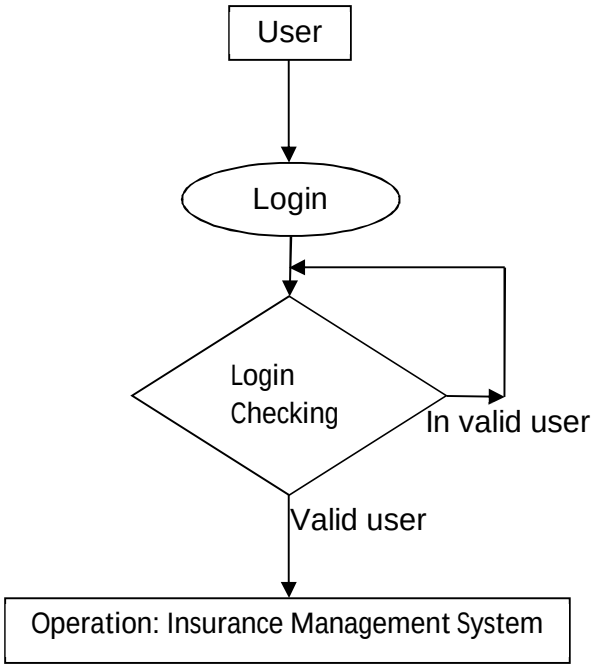


7.5. System Flow Diagram for the Document Management Module

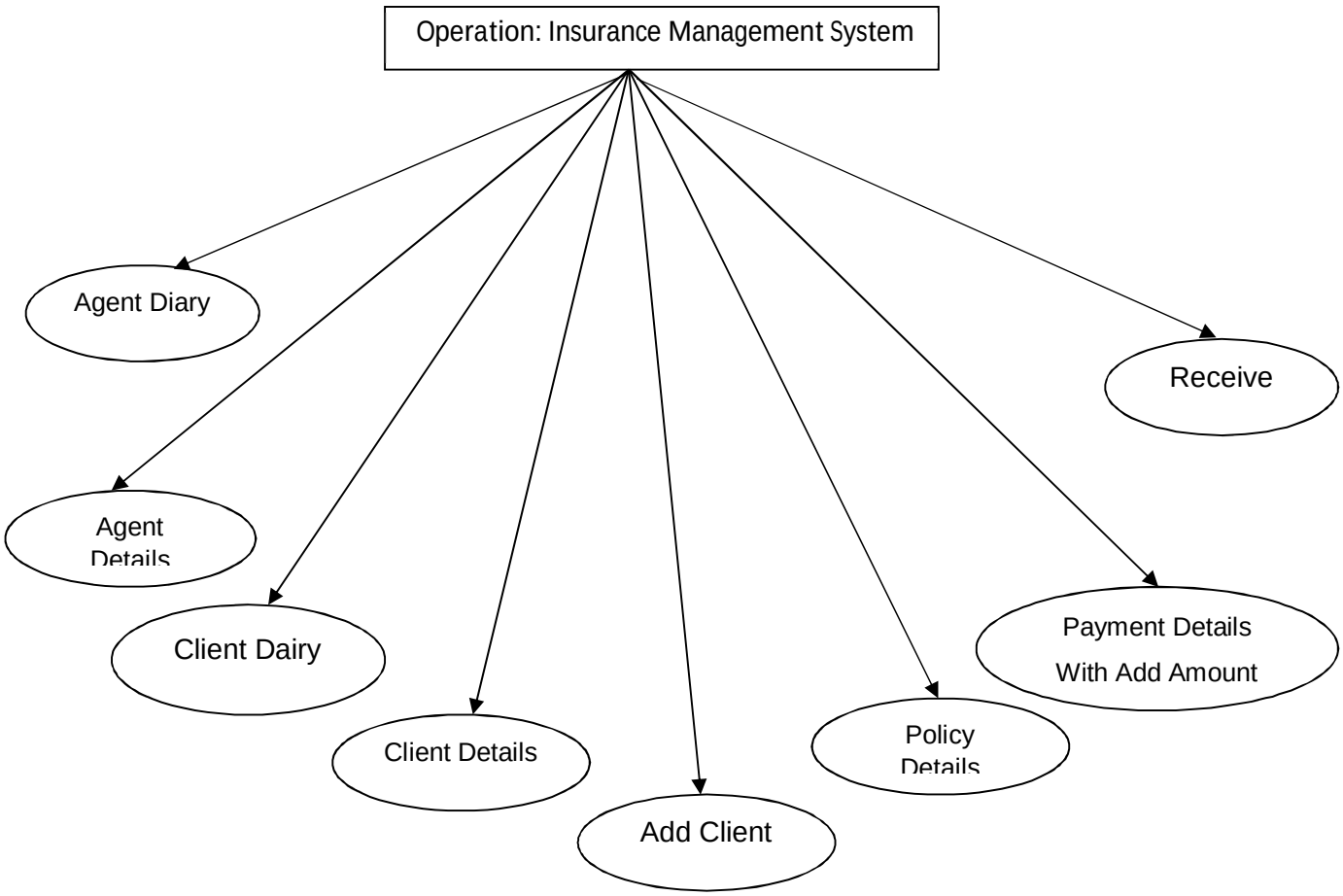


7.6. Data Flow Diagram for the Operation module of Insurance Management System

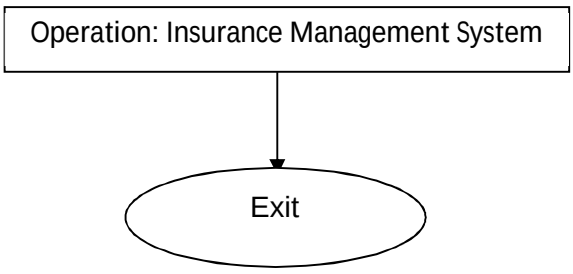
Level: 0 DFD



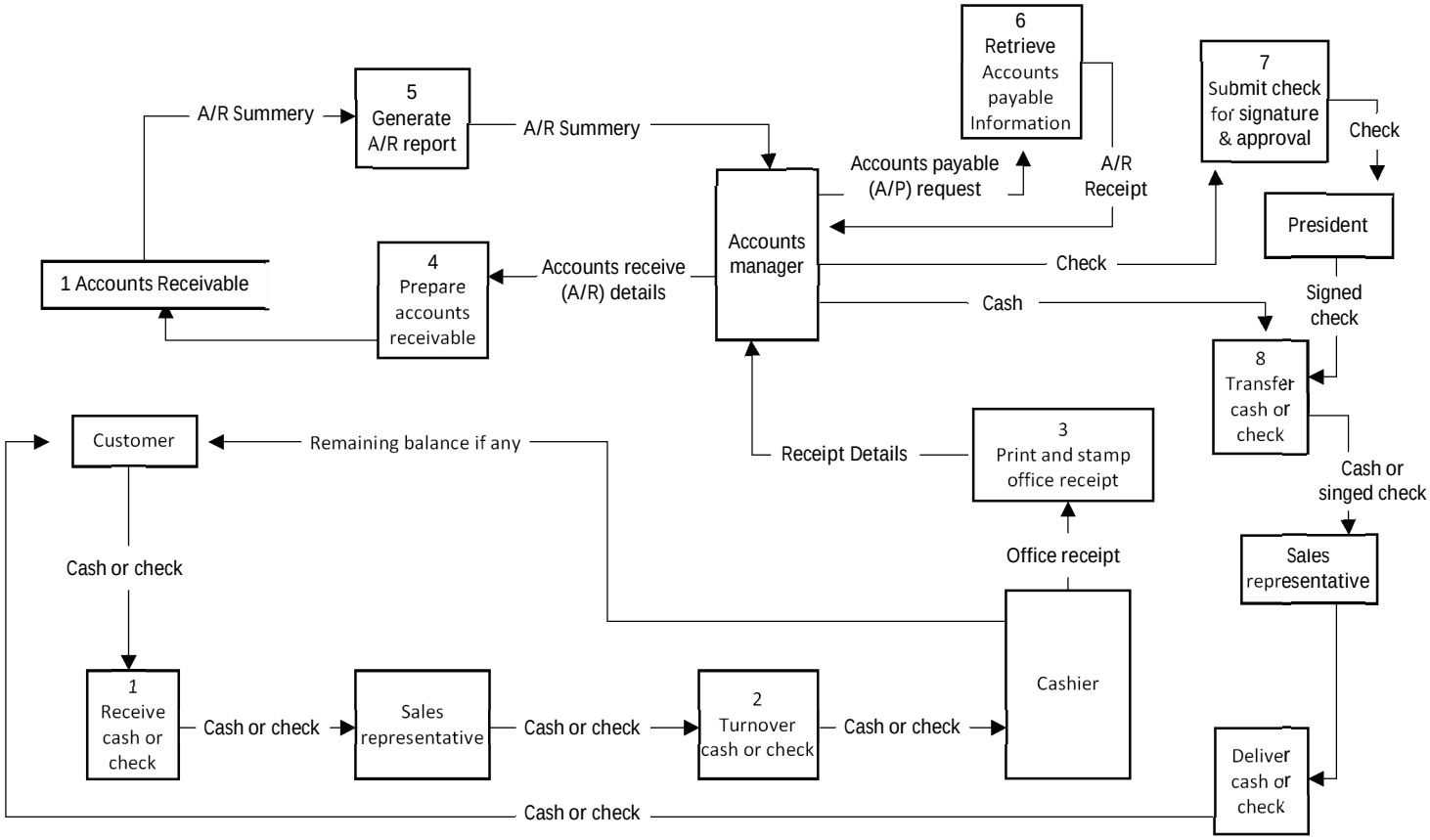
Level: 1 DFD



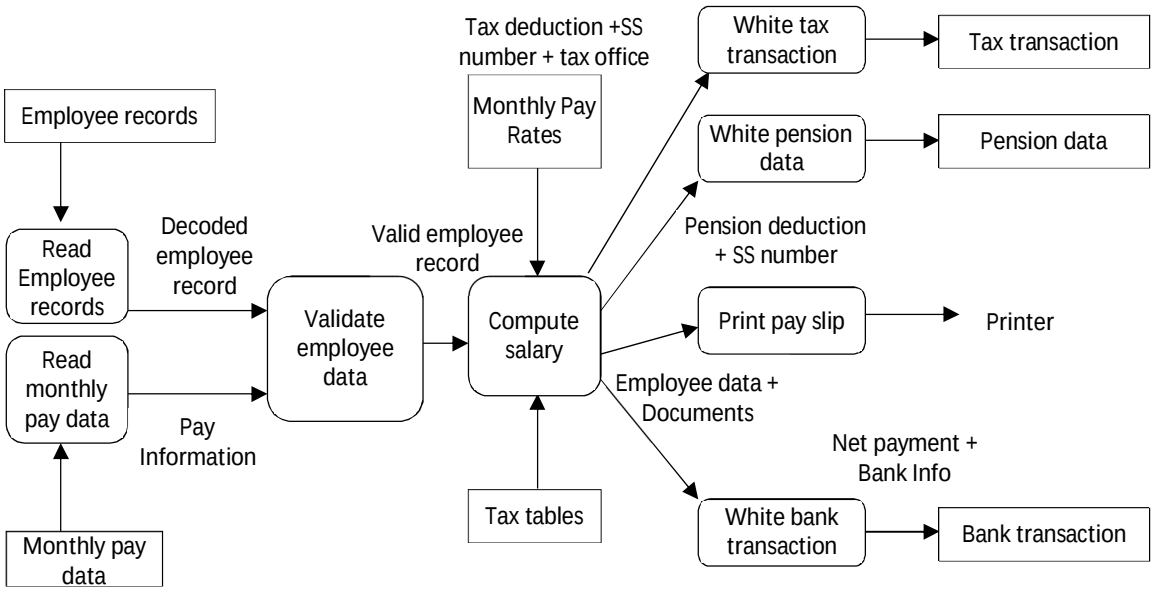
Level: 2 DFD



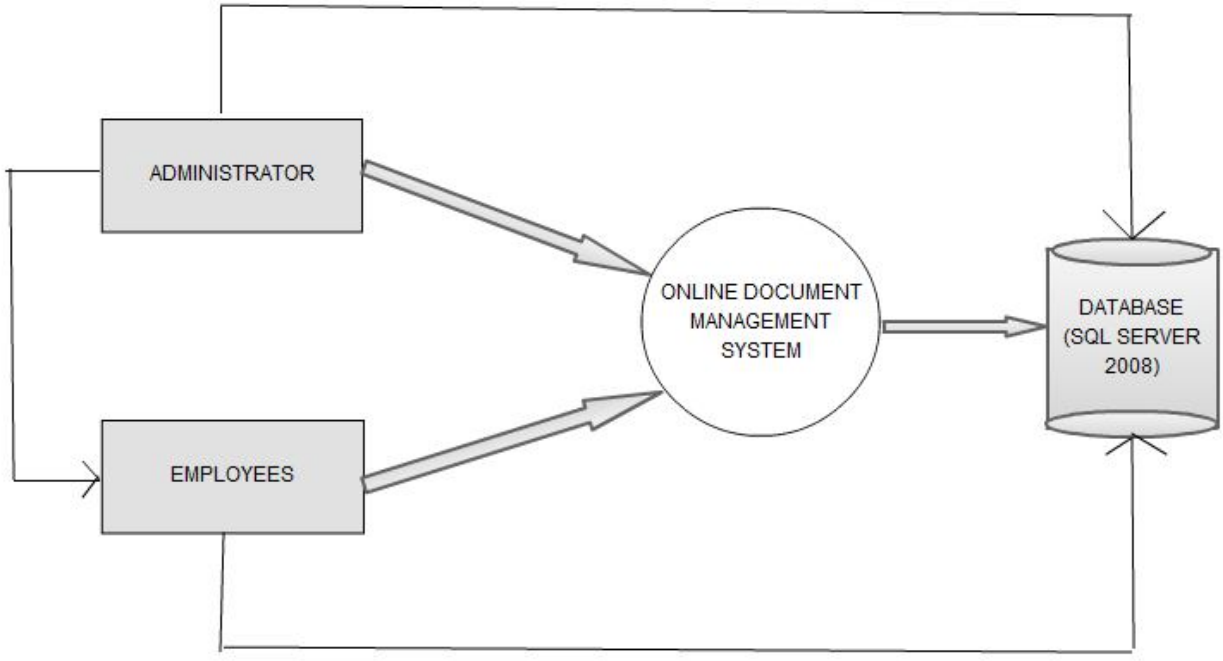
7.7. Data Flow Diagram for Accounting Module



7.8. Data Flow Diagram for HR and Payroll Module



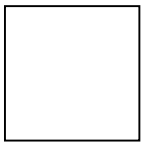
7.9. Data Flow Diagram for Document Management Module



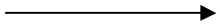
A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change or transform data throughout a system. It is a structured analysis and design tool that can be used for flowcharting in place of, or in association with, information oriented and process oriented system flowcharts. When analysts prepare the DFD, they specify the user

needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply a physical implementation. The DFD reviews the current physical system, prepares input and output specification, specifies the implementation plan etc., Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

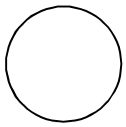
A "Square" defines a source (originator) or destination of a system data.



An "Arrow" identifies data flow. It is a pipeline through which information flows.



A "Circle" represents a process that transforms incoming data flow(s) into outgoing data flow(s).



An "Open Rectangle" is a data store.

Steps to Construct Data Flow Diagrams

Steps are commonly used to construct a DFD

- Process should be named and numbered for easy reference. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right.
- When a process is exploded in to lower level details they are numbered.

8. SYSTEM TESTING

8.1. System Testing

Software testing is the process of executing the program with the sole interest of finding an error. A good test is the one that a high probability of finding the yet undiscovered error. This test focuses on each module separately assuring that it functions properly as a unit.

The enables users to detect errors and correct it without affecting any other modules. Whenever the program was not executing the required function, it must be corrected to get the required result. Thus all the modules, master, transaction, queries, reports, are tested separately with test data. Errors isolated and corrected. To check error-handling paths, uncorrected data is entered and tested.

8.1. Test Methodologies

8.1.55. Unit Testing

In this, each module of the program is tested individually to find out whether retriever, manipulate and store any loss of data.

For example, machine code, company code and power generation hours won't negative or zero values, this condition was tested successfully by entering zero and negative values. Similarly, if you want to input correct date you enter some other date system won't accept.

8.2.2. Validation

Data Validation is done to see whether the corresponding entries in different tables done correctly. Proper validation checks are done in case of insertion and updating of tables, in order, to see that no duplication of data has occurred. Validation succeeds when the software functions in a manner as expected by the customer.

8.2.3. Interaction Testing

In this testing, all modules of the application are combined together and running is made. This ensures the co-ordination between different modules of the program. This test starts from main menu to all the modules. Form term main menu we can to the entire application module and to the main module.

8.2.4. Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. The outputs generated or displayed by the system under consideration are tested by asking the users about the format required by them. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

9. IMPLEMENTATION

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned a controlled it can cause chaos and confusion.

Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be totally new, replacing an existing manual or automated system or it may be a major modification to an existing system. Proper implementation is essential to provide a reliable system to meet the organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it.

The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after thorough testing is done and if it is found to be working according to the specifications. The system personnel checks the feasibility of the system.

The most crucial stage is achieving a new successful system and giving confidence on the new system for the user that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover. The more complex the system being implemented, the more involved will be the system analysis and the design effort required just for implementation. The system implementation has three main aspects. They are education and training, system testing and changeover.

The implementation stage involves following tasks.

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover phase.
- Evaluation of the changeover method.

The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and the same time users are trained in the new procedures.

9.1. Implementation Procedures

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended users and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. The people who are not sure that the software is meant to make their job easier. In the initial stage, they doubt about the software but we have to ensure that the resistance does not build up as one has to make sure that

- The active user must be aware of the benefits of using the system
- Their confidence in the software is built up
- Proper guidance be imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual processes won't take place.

10. SYSTEM MAINTANANCE

The maintenance phase of the software cycle is the time in which a Software product performs useful work.

After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. There may be social, technical and other environmental changes, which affects a system, which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. So only through proper system maintenance procedures, the system can be adapted to cope up with these changes.

Software maintenance is of course, far more than "finding mistakes". We may define maintenance by describing four activities that are undertaken to after a program is released for use.

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. During the use of any large program, errors will occur and be reported to the developer.

The process that includes the diagnosis and correction of one or more errors is called corrective maintenance.

The second activity that contributes to a definition of maintenance occurs because of the rapid change that is encountered in every aspect of computing. Therefore, adaptive maintenance- an activity that modifies software to properly interfere with a changing environment is both necessary and commonplace.

The third activity that may be applied to a definition of maintenance occurs when a software package is successful. As the software is used, recommendations for new capabilities, modifications to existing functions, and general enhancements are received from users. To satisfy requests in this category, perfective maintenance is performed. This activity accounts for the majority of all effort expended on software maintenance.

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability, or to provide a better basis for future enhancements. Often called preventive maintenance, this activity is characterized by reverse engineering and re-engineering techniques.

11. CONCLUSION AND FUTURE ENHANCEMENT

The **"ERP Solution"** has been developed to fill all the requirements of the Meghna Life Insurance. The system will be tested with the sample data and found to be executing at its maximum performance.

The system is fully user interactive with command buttons for selecting various options to navigate other table processing and manipulate the effective handling of Client Details and Payment Details. Report from the system provides complete details about the Policy to the Insurance Agency, hence the daily report taken so that any future change in the current system will improve the efficiency of the generating process.

By using this software, they can take quick decisions and preventive actions based on the details given by the system. Due to the software, I hope quality will be improved, problems will be solved.

It is user friendly system provided with options, which can be utilized by the desired operations. The new system overcomes the problems encountered with the old system.

********** Thank You **********