SYNOPSIS

TITLE OF THE PROJECT

Email Service Automation System

ABSTRACT & INTRODUCTION

This project is about developing a web-based mail client connecting to windows Server running a Mail Server. This Project has the following main functionality

1. Receiving/Sending/organizing mails.
2. Sending mail using send mail.
3. Performing Admin functions like managing new user, resetting passwords etc.

The software is fully integrated with **CRM** (Customer Relationship Management) as well as **CMS** (Content Management System) solution and developed in a manner that is easily manageable, time saving and relieving one from manual works.

EXISTING SYSTEM:

- Cannot Upload and Download the latest updates.
- No use of Web Services and Remoting.
- Risk of mismanagement and of data when the project is under development.
- Less Security.
- No proper coordination between different Applications and Users.
- Fewer Users - Friendly.
- Manual system needs man power a lot.
- Communication between Patient and administration is a tuff job.
- Difficult to maintain each and patient information in form of files.

PROPOSED SYSTEM:

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

- User friendliness is provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- Readily upload the latest updates, allows user to download the alerts by clicking the URL.
- There is no risk of data mismanagement at any level while the project development is under process.
- It provides high level of security with different level of authentication.

OBJECTIVES

1. Capability to create user Email Accounts by an Administrator or by End users after registering themselves
2. Administrator functionality to Delete User Accounts, Change passwords
3. Capability for End users to login into the system using a browser
4. Capability for logged in users to send/receive/forward/reply/delete mails
5. Invalidate user login on inactive for more than 10mts
6. Address book capability
7. Mark mails as Junk
8. Apply Label to Mail
9. Organize mails in Logical Folders

PROJECT CATEGORY

RDBMS

The Project is developed using Relational Database Management System (RDBMS).

A database system is essentially a sophisticated, computerized record keeping system, a repository for a collection of computerized data files. A database system maintains information and makes that information available on demand, for this purpose a database system provides set of facilities to perform such operations.

The benefits of a database system over any traditional system are obvious as database is integrated as well as shared, thus a database eliminates redundancy and also as a consequence, database lets multiple users access the same piece of data.

The most important advantage of the database is to maintain the integrity i.e. it insures that the change made to the database by authorized users do not result in a loss of data consistency and guard against accidental damage to the database.

Facilities offered by RDBMS:

- Creation of files, Addition of data, Deletion of data, Modification of data.
- Retrieving data collectively or selectively.
- The data stored can be sorted or indexed at users discretion or direction.
- Various reports can be produced from the system. These may either be standardized reports or that may be specifically generated according to specific user definition.
- Mathematical function can be performed and the data stored in the database can be manipulated with functions to perform the desired calculations.
- To maintain data integrity and database use.
- Data integrity for multiple users.
- Providing form-based interface for easy accessibility and data entry.

Tools/Platform, Hardware and Software Requirement Specifications:

SRS Document

It is a reference document or contract between the Staff and the development team. Once the Staff agrees to the SRS document the development team proceeds to develop the product conforming to all the requirements mentioned in the SRS document.

An SRS document should clearly document the following:

1. Functional requirements of the system.
2. Non-functional requirements of the system.
3. Constraints on the system.

1. Functional requirements of the system: Each of the system can be considered as performing a transformation of a set of input data to the corresponding set of output data. The functional requirements of the system should clearly describe each of the functions that the system needs to perform along with the corresponding input and output data set.
2. Non-functional requirements of the system: Non-functional requirements deal with the characteristics of the system that cannot be expressed functionally, e.g., maintainability, portability, Usability, etc. The non-functional requirements also include reliability issues, accuracy of results, human computer interface issues, operating and Physical constraints, etc.
3. Constraints on the system: The constraints on the & u"s of the system may describe certain things that the system should or should not do.

Natures of SRS

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

1. Functionality: What the software is supposed to do?
2. External Interfaces: How does the software interact with people, the system's hardware other hardware and other software.
3. Performance: What is the speed, availability, response time, recovery time, etc., of the various software fundamentals?
4. Attributes: What are the considerations for portability, correctness, maintainability, security, reliability, etc.
5. Design constraints imposed on an implementation: Are there any required standards or effect, implementation language, policies for database, integrity resource limits, operating environment, etc.

Characteristics of a good SRS

An SRS should be

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for Importance and for Stability
- Verifiable
- Modifiable
- Traceable

Correct: There is no tool or procedure that assures correctness. If the software must respond to all button presses within 5 seconds and the SRS stated that “the software shall respond to all button presses within 10 seconds”, then that requirement is incorrect.

Unambiguous: An SRS is unambiguous if and only if every requirement started therein has only are interpretation. In cases, where a term used in a particular context could have multiple meanings, the term should be included in a glossary where its meaning is made more specific.

Complete: An SRS is complete if and only if it includes of the following elements.

1. All significant requirements, whether relating to functionality, performance, design constraints, attributes or external interfaces.
2. Full labels and references to all figures, tables and diagram in the SRS and definition of all terms and units of measure.

Consistent

An SRS is consistent if no subset of individual requirements desorbed in it conflict. There are 3 types of likely conflicts in an SRS:

1. The specified characteristics of real word objects may conflict, e.g.
    a. The format of an output report may be described in are requirements as tabular but in another as textual.
    b. One requirement may state that all lights shall be green while another states that all lights should be blue.
2. There may be logical or temporal conflict between two specified actions, e.g.
    a. Are requirement may specify that the program will add 2 inputs and another may specify that the program will multiply them.

b. Are requirement may state that 'A' must always follows B, while another requires that A&B occur simultaneously.

3. Two or more requirements may describe the same real word object but use different terms for that object. The use of standard terminology and definitions promotes consistency.

Hardware Interface

Server: Pentium IV Machine with 2 GB RAM
Clients: Pentium Series, 1 GB RAM

Software Interfaces

Operating System : Windows 10
Front-end : ASP.NET C# 2017

Back-end : SQL Server 2008
Reporting Tool : Crystal report

Memory Constraints

Server : 2 GB RAM
Recommended 4 GB RAM
1 TB HDD
Clients : 8 GB RAM

Constraints

- Hardware limitations:
- Interface to other applications
- Reliability requirements
- Criticality of the requirement
- Security considerations

ABOUT ASP.NET

ASP.NET 2017 allows users to write programs that break down into modules. These modules will represent the real-world objects and are knows as classes or types. An object can be created out of a class and it is known as an instance of the class. A class can also comprise subclass. For example, apple tree is a *subclass* of the *plant* class and the apple in your backyard is an instance of the apple tree class.

PROBLEM DEFINITION

(Definition of Problem)

It takes considerable skill to determine the true cause of a systems problem. A systems analyst might begin to define the problem by determining if the problem can be classified - cording to one or more common types of systems problems. With knowledge of the Common types of problems, the analyst can diagnose a problem by application its characteristics.

One of the most difficult tasks of system analysis is developing a clear, in-depth under-standing of the problem being investigated, without which it becomes impossible to specify the requirements for a new project with any accuracy. Several questions should be posed for this.

Some of those may be:

- What is the problem?
- How complex is it?
- What are its likely causes?
- Why is it important that the problem be solved?
- What are possible solutions to the problem?
- What types of benefits can be expected once the problem is solved?

Organizations usually face problems or have opportunity due to the following:

- a new product or plant or branch
- a new market or new process
- failure of an existing system
- inefficiency of an existing system
- Structural error in the existing system, etc.

For identifying problems/opportunities, we scan the following:

- the performance of the system
- the information being supplied and its form
- the economy of processing
- the control of the information processing
- the efficiency of the existing system
- the Fire Control System of the data and software
- the Fire Control System of the equipment and personnel, etc

After identification of the problem, it is defined and a general direction or method for solving this problem is determined. Then project boundaries are defined. The management establishes the term of reference as well as the resources to be provided for the project. System development is an iterative process and the first identifiable stage of it is Problem Definition, whose final output is Terms of Reference.

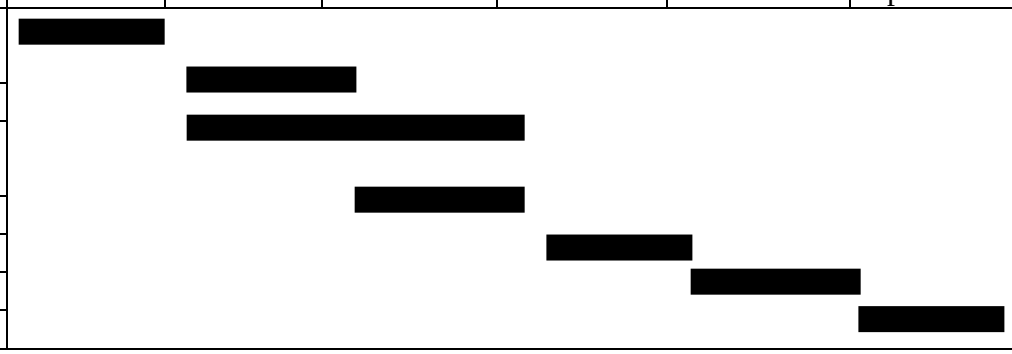
Careful analysis of this system suggests a number of different problems:

1. Problem of Reliability: Current system is not reliable. It seems to vary in quality and Skills from one month to the, next. Sometimes it gives good output, but sometimes the output is worst.
2. Problem of Accuracy: There are too many mistakes in reports and calculations.
3. Problem of timeliness: In the current system the reports and output produced is mostly late and in most of the cases it is useless because it is not on time.
4. Problem of Validity: The output and reports mostly contains misleading information. The transactions and calculations are sometimes not valid.
5. Problem of Economy: The current system is very costly. We have to spend lots of money in manual filing, calculations and remembering the information, but still not get the desired Skills.

Project Planning and Scheduling

The given **Gantt chart** Show the all activities according their time consuming.

Task	02 Oct – 6 Oct	7 Jan – 18 Feb	19 Feb. – 15 Mar	16 Mar-20 Mar	21 Mar.- 25 Mar	26 Mar- 1 April
Preliminary Investigation						
Feasibility Study						
Requirement Analysis						
Design						
Coding						
Testing						
Implement						



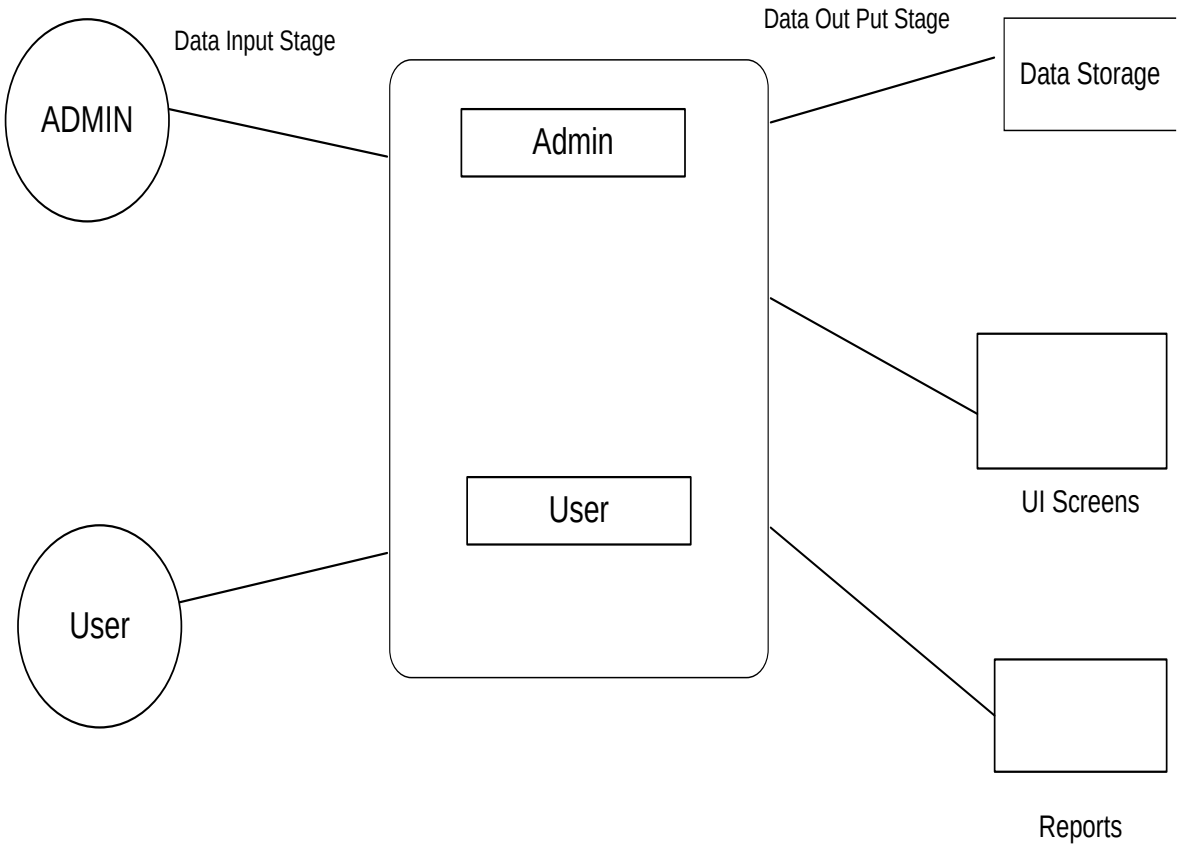
SCOPE OF THE SOLUTION

- In future this application can be work with different multi-center of Applications.
- More modules can be added as per project requirements.
- Multiple languages can be supported by this application.
- The multi-interface can support.

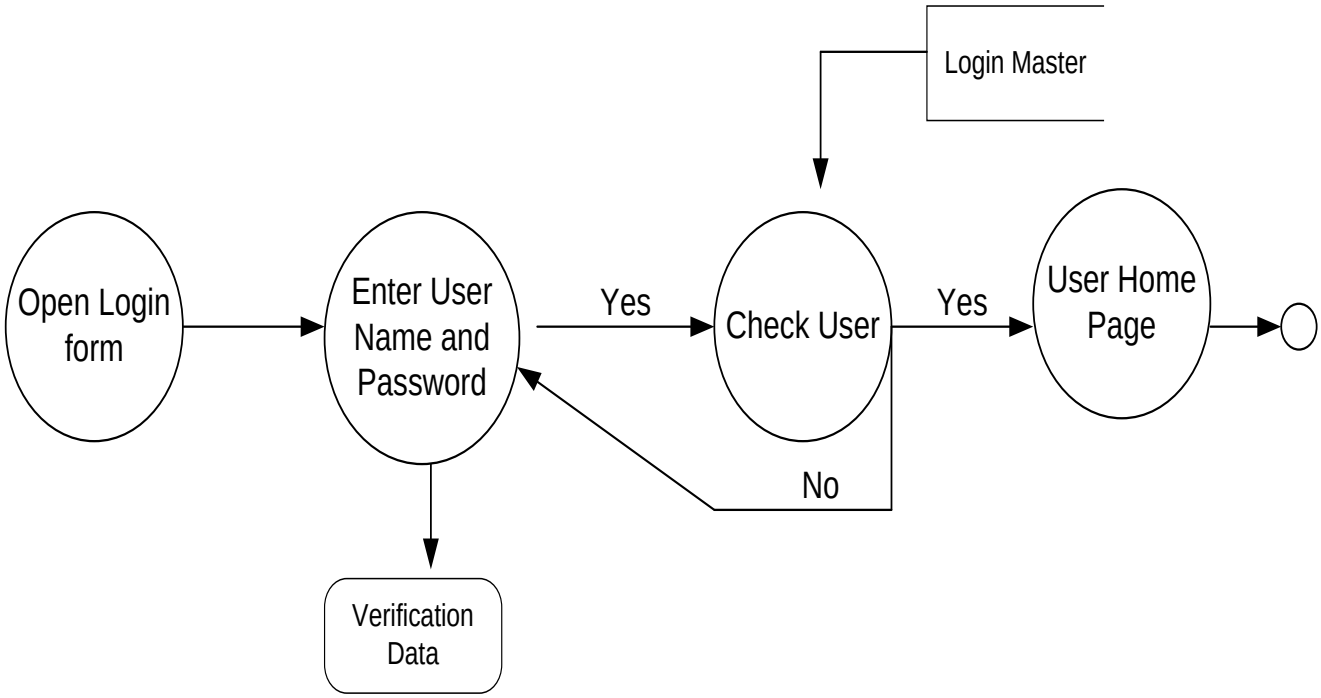
Analysis and Design

Mail Client DFD's

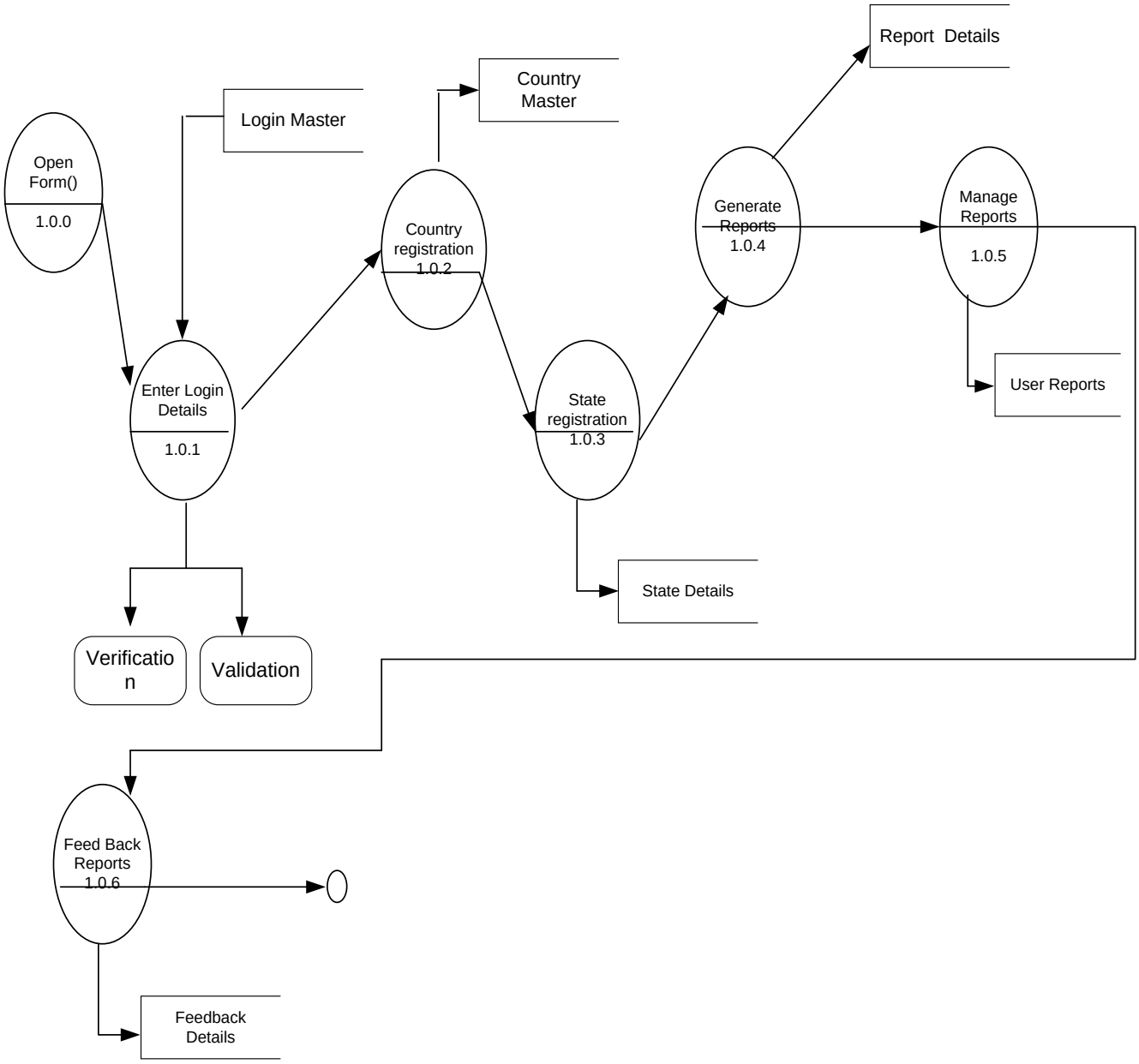
Context Level DFD



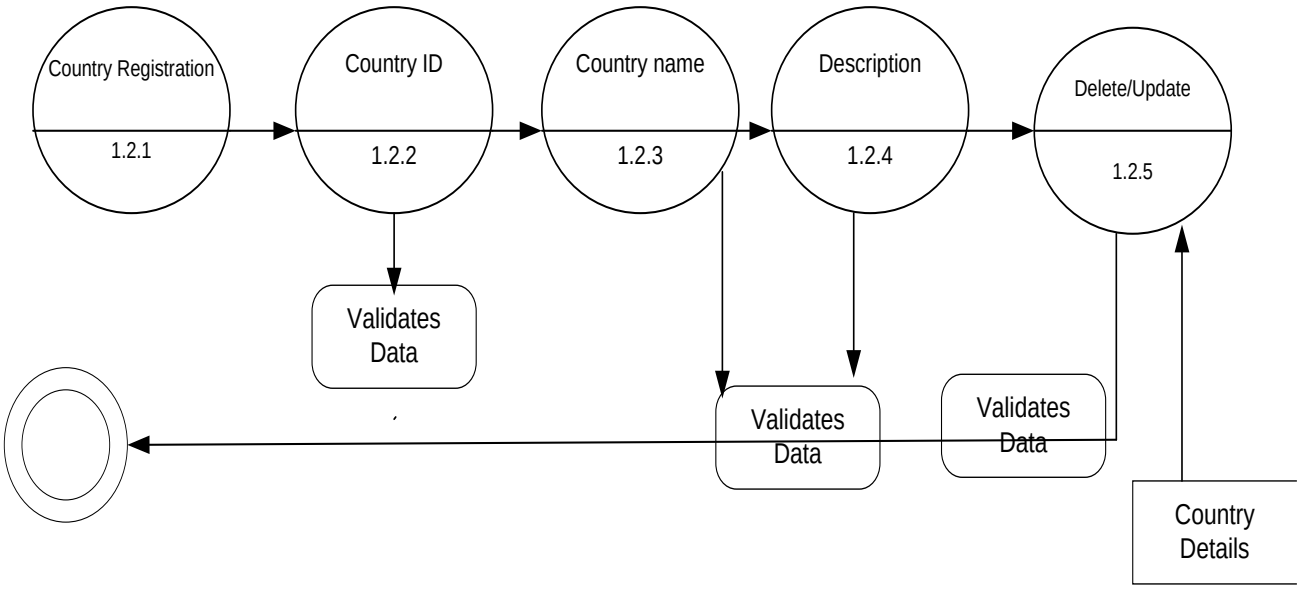
1st Level DFD Login



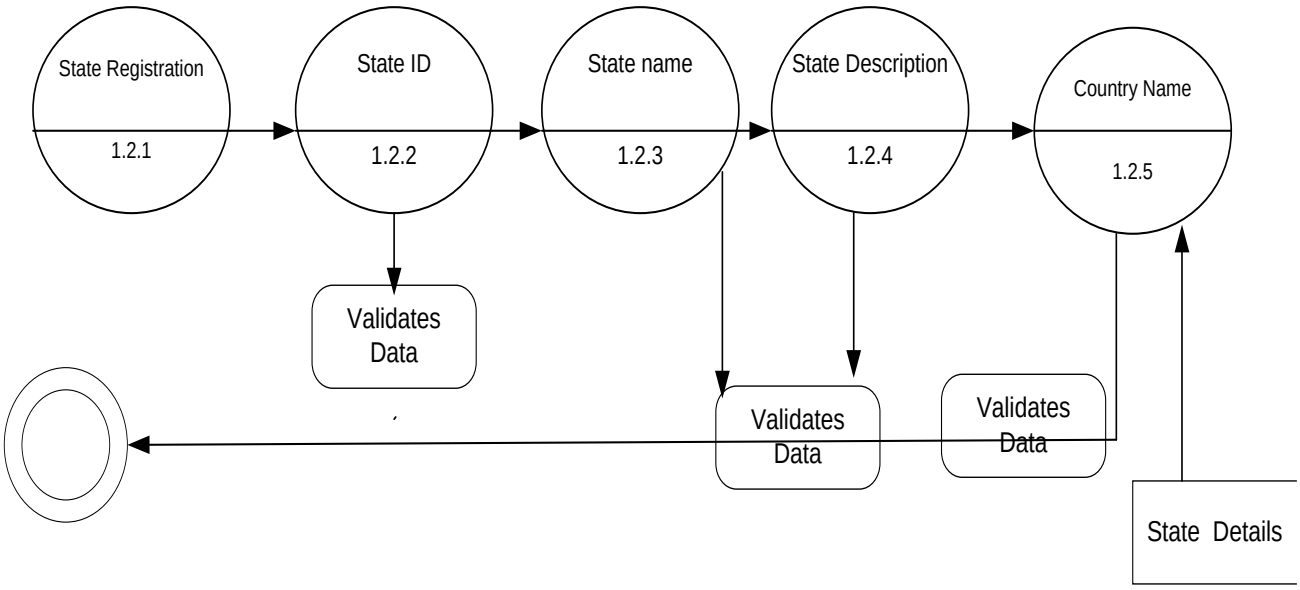
Admin Activities (1st Level)



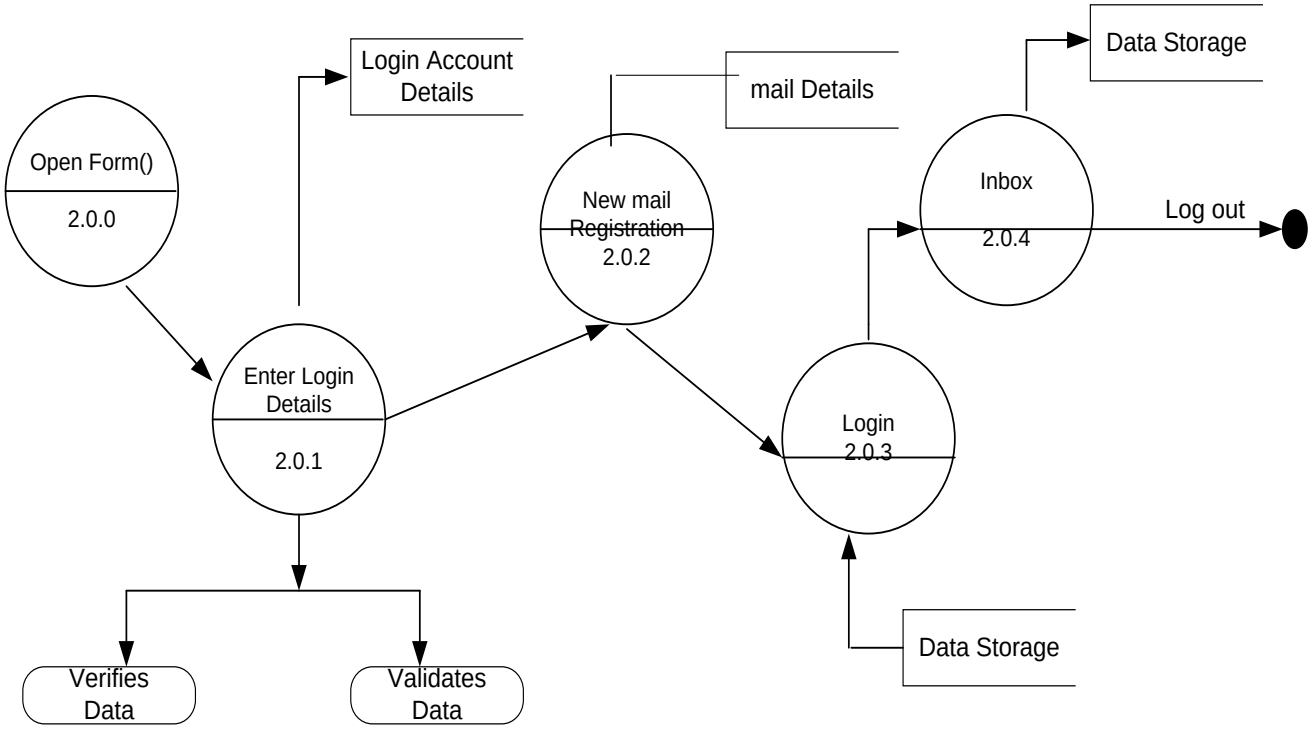
1ST Level for Admin Register Country

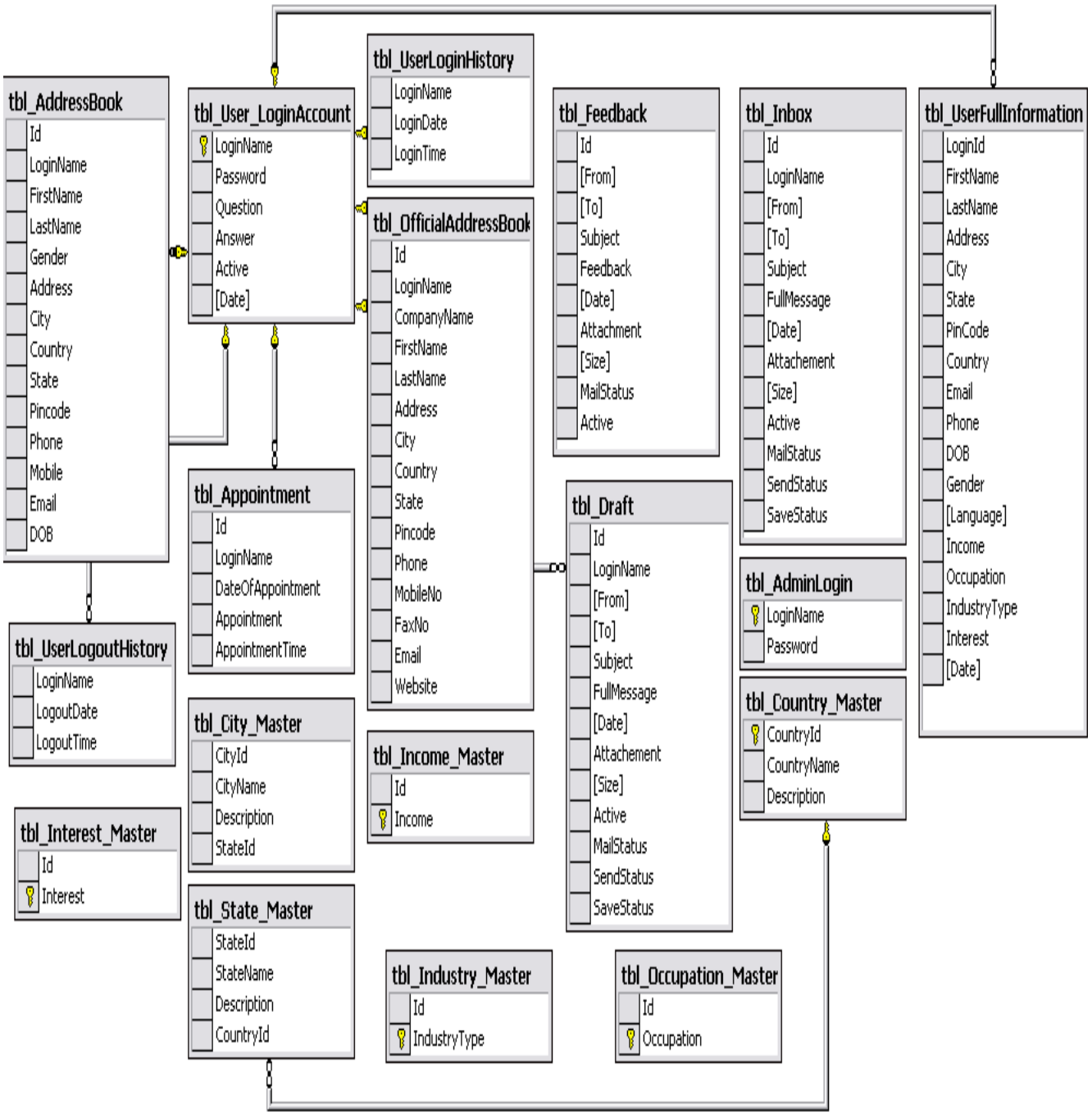


1st Level for Admin Register State



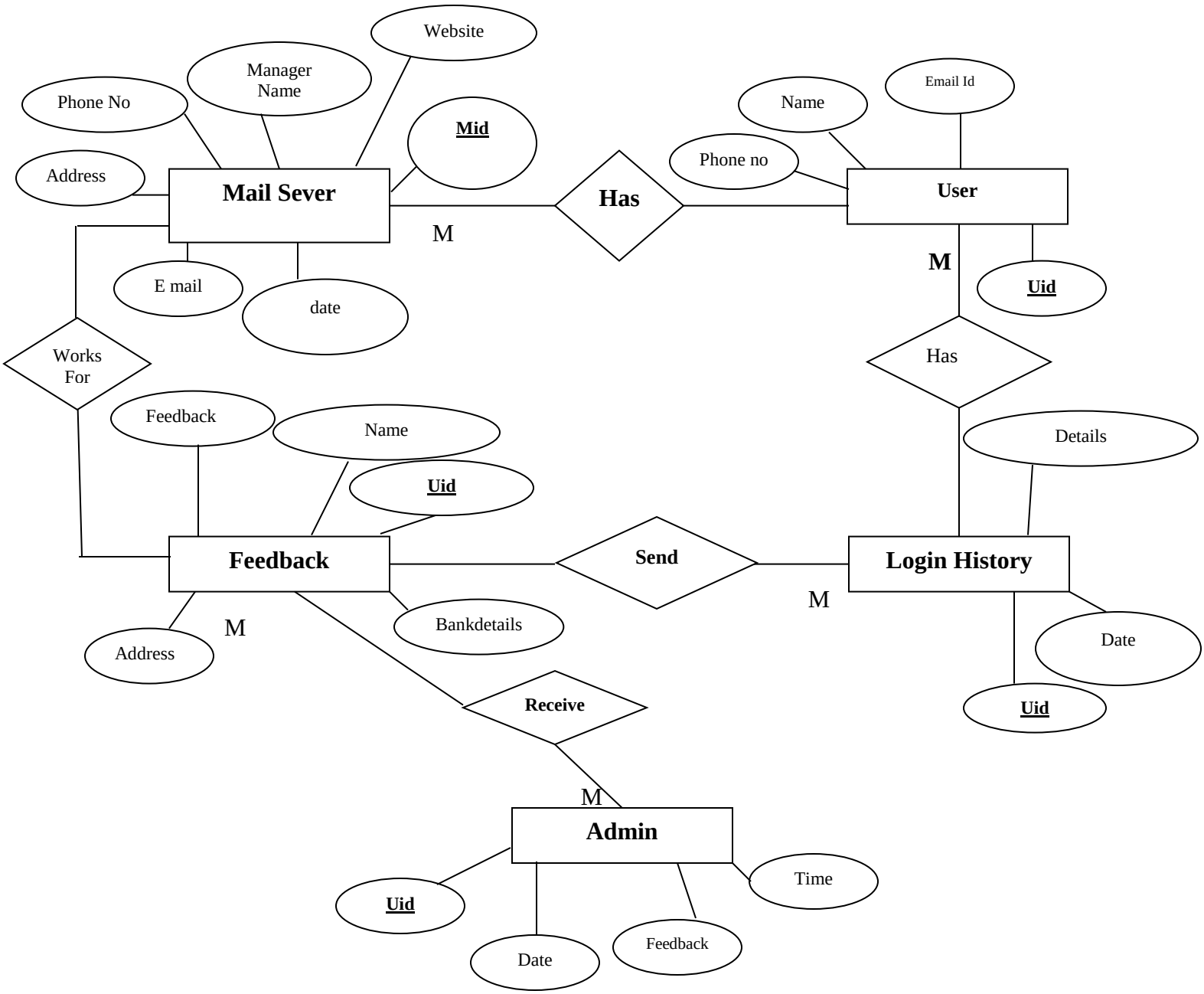
User (Employee) Activities (2nd Level):



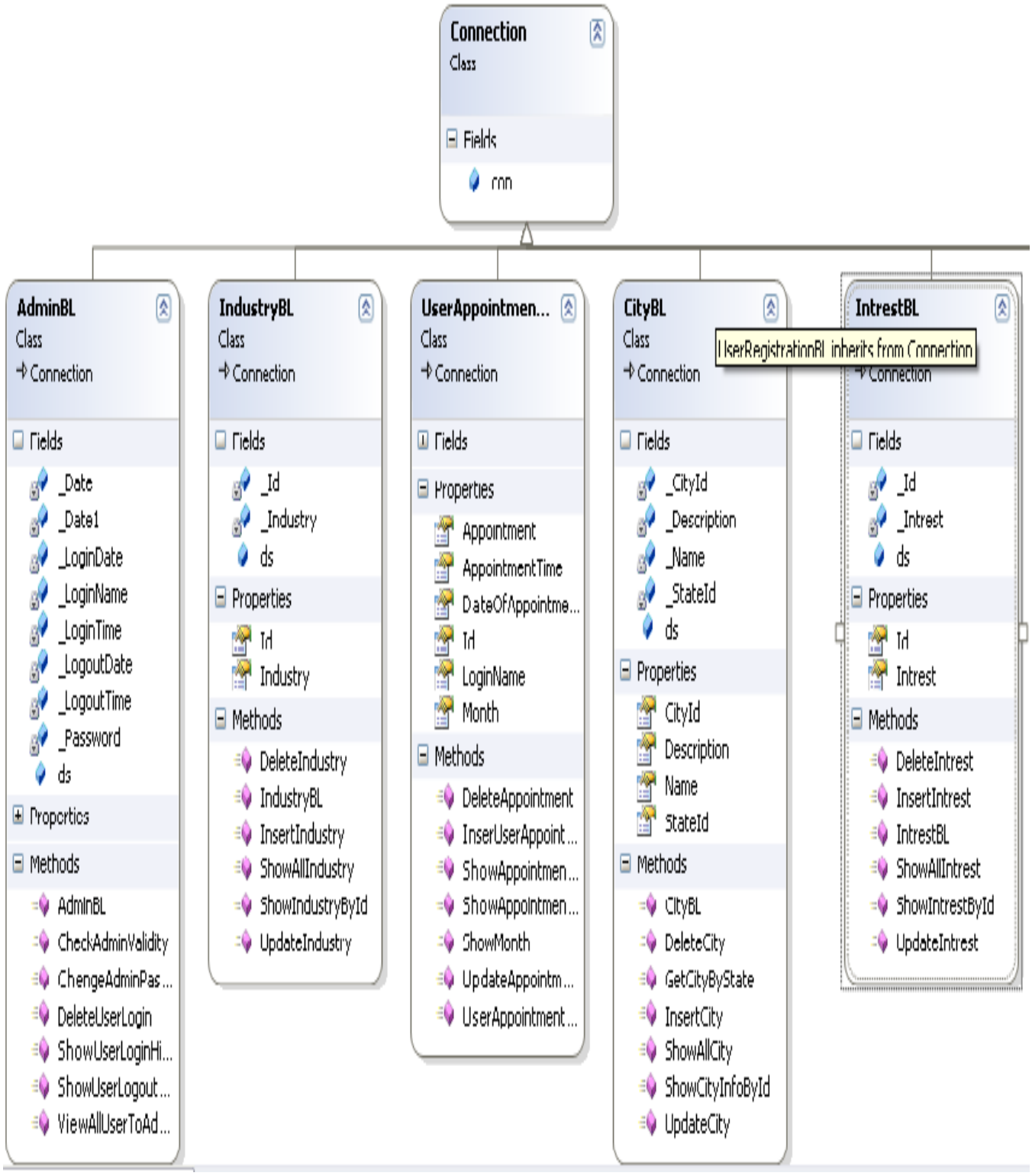

tbl_AddressBook
Id
LoginName
FirstName
LastName
Gender
Address
City
Country
State
Pincode
Phone
Mobile
Email
DOB
tbl_User_LoginAccount
LoginName
Password
Question
Answer
Active
[Date]
tbl_UserLoginHistory
LoginName
LoginDate
LoginTime
tbl_OfficialAddressBook
Id
LoginName
CompanyName
FirstName
LastName
Address
City
Country
State
Pincode
Phone
MobileNo
FaxNo
Email
Website
tbl_Feedback
Id
[From]
[To]
Subject
Feedback
[Date]
Attachment
[Size]
MailStatus
Active
tbl_Inbox
Id
LoginName
[From]
[To]
Subject
FullMessage
[Date]
Attachement
[Size]
Active
MailStatus
SendStatus
SaveStatus
tbl_UserFullInformation
LoginId
FirstName
LastName
Address
City
State
PinCode
Country
Email
Phone
DOB
Gender
[Language]
Income
Occupation
IndustryType
Interest
[Date]
tbl_Appointment
Id
LoginName
DateOfAppointment
Appointment
AppointmentTime
tbl_Draft
Id
LoginName
[From]
[To]
Subject
FullMessage
[Date]
Attachement
[Size]
Active
MailStatus
SendStatus
SaveStatus
tbl_AdminLogin
LoginName
Password
tbl_Country_Master
CountryId
CountryName
Description
tbl_UserLogoutHistory
LoginName
LogoutDate
LogoutTime
tbl_City_Master
CityId
CityName
Description
StateId
tbl_Income_Master
Id
Income
tbl_Interest_Master
Id
Interest
tbl_State_Master
StateId
StateName
Description
CountryId
tbl_Industry_Master
Id
IndustryType
tbl_Occupation_Master
Id
Occupation

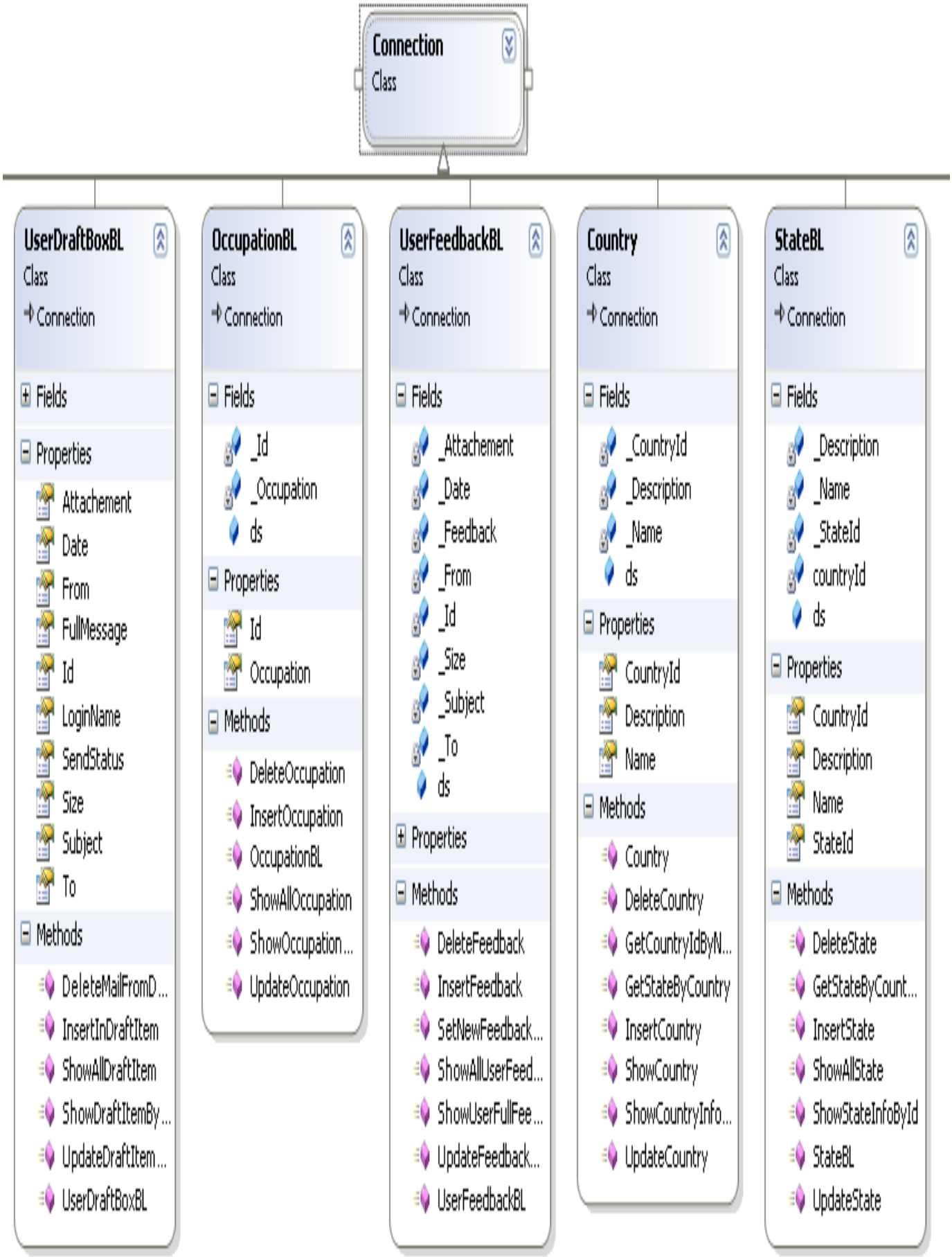
E-R DIAGRAM

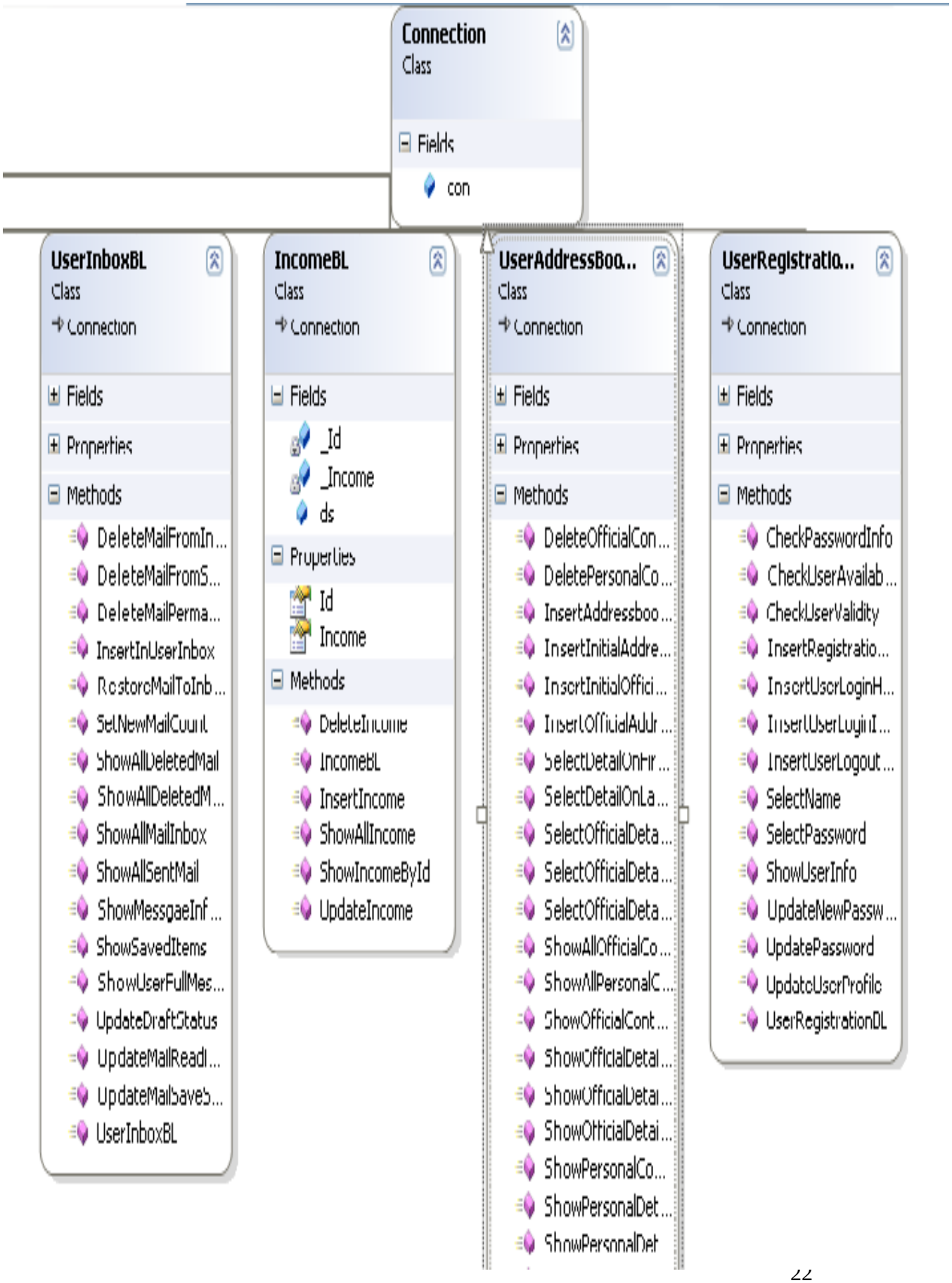
E- R Diagram

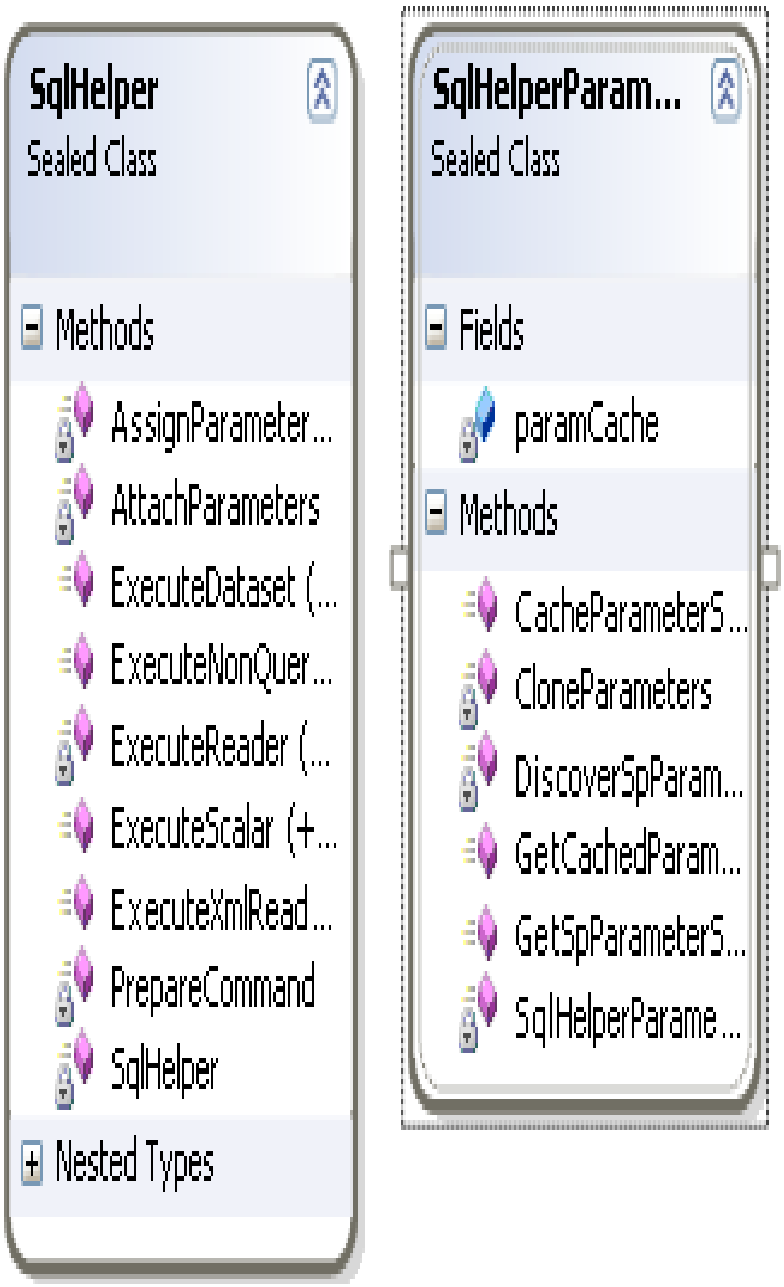

Website
Manager Name
Phone No
Mid
Address
Mail Sever
Has
Name
Email Id
Phone no
User
M
M
E mail
date
Uid
Works For
Has
Feedback
Name
Details
Uid
Feedback
Send
Login History
M
M
Address
Bankdetails
Date
Receive
M
Admin
Time
Uid
Date
Feedback

CLASS DIAGRAM




Connection
Class
UserDraftBoxBL
Class
Connection
Fields
Properties
Attachement
Date
From
FullMessage
Id
LoginName
SendStatus
Size
Subject
To
Methods
DeleteMailFromD...
InsertInDraftItem
ShowAllDraftItem
ShowDraftItemBy...
UpdateDraftItem...
UserDraftBoxBL
OccupationBL
Class
Connection
Fields
_Id
_Occupation
ds
Properties
Id
Occupation
Methods
DeleteOccupation
InsertOccupation
OccupationBL
ShowAllOccupation
ShowOccupation...
UpdateOccupation
UserFeedbackBL
Class
Connection
Fields
_Attachement
_Date
_Feedback
_From
_Id
_Size
_Subject
_To
ds
Properties
Methods
DeleteFeedback
InsertFeedback
SetNewFeedback...
ShowAllUserFeed...
ShowUserFullFee...
UpdateFeedback...
UserFeedbackBL
Country
Class
Connection
Fields
_CountryId
_Description
_Name
ds
Properties
CountryId
Description
Name
Methods
Country
DeleteCountry
GetCountryIdByN...
GetStateByCountry
InsertCountry
ShowCountry
ShowCountryInfo...
UpdateCountry
StateBL
Class
Connection
Fields
_Description
_Name
_StateId
countryId
ds
Properties
CountryId
Description
Name
StateId
Methods
DeleteState
GetStateByCount...
InsertState
ShowAllState
ShowStateInfoById
StateBL
UpdateState


Connection
Class
Fields
con
UserInboxBL
Class
Connection
Fields
Properties
Methods
DeleteMailFromIn...
DeleteMailFromS...
DeleteMailPerma...
InsertInUserInbox
RestoreMailToInb...
SetNewMailCount
ShowAllDeletedMail
ShowAllDeletedM...
ShowAllMailInbox
ShowAllSentMail
ShowMessgaeInf...
ShowSavedItems
ShowUserFullMes...
UpdateDraftStatus
UpdateMailReadI...
UpdateMailSaves...
UserInboxBL
IncomeBL
Class
Connection
Fields
_Id
_Income
ds
Properties
Id
Income
Methods
DeleteIncome
IncomeBL
InsertIncome
ShowAllIncome
ShowIncomeById
UpdateIncome
UserAddressBoo...
Class
Connection
Fields
Properties
Methods
DeleteOfficialCon...
DeletePersonalCo...
InsertAddressboo...
InsertInitialAddre...
InsertInitialOffici...
InsertOfficialAddr...
SelectDetailOnHr...
SelectDetailOnLa...
SelectOfficialDeta...
SelectOfficialDeta...
SelectOfficialDeta...
ShowAllOfficialCo...
ShowAllPersonalC...
ShowOfficialCont...
ShowOfficialDetai...
ShowOfficialDetai...
ShowOfficialDetai...
ShowPersonalCo...
ShowPersonalDet...
ShowPersonalDet
UserRegistratio...
Class
Connection
Fields
Properties
Methods
CheckPasswordInfo
CheckUserAvailab...
CheckUserValidity
InsertRegistratio...
InsertUserLoginH...
InsertUserLoginI...
InsertUserLogout...
SelectName
SelectPassword
ShowUserInfo
UpdateNewPassw...
UpdatePassword
UpdateUserProfile
UserRegistrationDL


SqlHelper
Sealed Class
Methods
AssignParameter...
AttachParameters
ExecuteDataset (...
ExecuteNonQuer...
ExecuteReader (...
ExecuteScalar (+...
ExecuteXmlRead...
PrepareCommand
SqlHelper
Nested Types
SqlHelperParam...
Sealed Class
Fields
paramCache
Methods
CacheParameterS...
CloneParameters
DiscoverSpParam...
GetCachedParam...
GetSpParameterS...
SqlHelperParame...

A complete structure which includes Number of Modules and their description

Number of Modules

The system after careful analysis has been identified to be presented with the following modules:

The modules involved are:

1. **Member registration Module:**
   - Member can register and sign in here. For registration, member has to provide personal details, address details, employment details, account details and they have to agree with policies.
   - Member can sign in by providing their account details (Username and password).
2. **Sending and Receiving mails:**
   - By making use of this module, Members can send/receive/view mails. Many features have been provided to members so that they can 1) manage (view/ edit/ delete) their mails, 2) forward mails, 3) send attachments, 4) send group mail, manage mails in folders etc.
3. **Integrated Security Module:**
   - This module is made is provide security features to the application.
4. **Admin Module:**
   - Admin is a super user and hence responsible for a) Site Maintenance, b) Members Management, c) Mails management and d) Generate various reports.
5. **Login/Logout Date & Time Tracking Module:** Admin can view the Login/Logout time of User. Whenever the user login/logout then Current Date & Time will be stored to view for Admin.
6. **Address book Maintenance:** Here user can maintain the address book for own friend with all Address Contact Info, Birthday, and Marriage Anniversary etc.
7. **CMS (content Management System) Integration:** Using CMS tool we can customize the mail the message with all formatting features.

Data Structure of Each Module

Address Book

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
LoginName	varchar	50	
FirstName	varchar	50	✓
LastName	varchar	50	✓
Gender	varchar	20	✓
Address	varchar	50	✓
City	varchar	50	✓
Country	varchar	50	✓
State	varchar	50	✓
Pincode	varchar	20	✓
Phone	varchar	20	✓
Mobile	varchar	20	✓
Email	varchar	50	✓
DOB	datetime	8	✓

Admin Login

Column Name	Data Type	Length	Allow Nulls
LoginName	varchar	50	
Password	varchar	50	✓

Appointment

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
LoginName	varchar	50	✓
DateOfAppointment	datetime	8	✓
Appointment	varchar	100	✓
AppointmentTime	varchar	30	✓

City Master

Column Name	Data Type	Length	Allow Nulls
CityId	int	4	
CityName	varchar	50	✓
Description	varchar	80	✓
StateId	int	4	✓

Country Master

Column Name	Data Type	Length	Allow Nulls
CountryId	int	4	
CountryName	varchar	50	✓
Description	varchar	80	✓

Draft

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
LoginName	varchar	50	✓
[From]	varchar	100	✓
[To]	varchar	100	✓
Subject	varchar	100	✓
FullMessage	varchar	500	✓
[Date]	datetime	8	✓
Attachement	varchar	50	✓
[Size]	varchar	20	✓
Active	tinyint	1	✓
MailStatus	char	10	✓
SendStatus	char	15	✓
SaveStatus	char	20	✓

Feedback

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
[From]	varchar	50	✓
[To]	varchar	50	✓
Subject	varchar	50	✓
Feedback	varchar	200	✓
[Date]	datetime	8	✓
Attachment	varchar	50	✓
[Size]	varchar	20	✓
MailStatus	char	10	✓
Active	tinyint	1	✓

Inbox

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
LoginName	varchar	50	✓
[From]	varchar	100	✓
[To]	varchar	100	✓
Subject	varchar	100	✓
FullMessage	varchar	500	✓
[Date]	datetime	8	✓
Attachement	varchar	50	✓
[Size]	varchar	20	✓
Active	tinyint	1	✓
MailStatus	char	10	✓
SendStatus	char	10	✓
SaveStatus	char	20	✓

Income

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
Income	varchar	50	

Industry

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
IndustryType	varchar	80	

Interest

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
Interest	varchar	80	

Occupation Master

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
Occupation	varchar	80	

Official Address Book

Column Name	Data Type	Length	Allow Nulls
Id	int	4	
LoginName	varchar	50	
CompanyName	varchar	50	✓
FirstName	varchar	50	✓
LastName	varchar	50	✓
Address	varchar	50	✓
City	varchar	50	✓
Country	varchar	50	✓
State	varchar	50	✓
Pincode	varchar	20	✓
Phone	varchar	20	✓
MobileNo	varchar	20	✓
FaxNo	varchar	20	✓
Email	varchar	50	✓
Website	varchar	50	✓

State Master

Column Name	Data Type	Length	Allow Nulls
StateId	int	4	
StateName	varchar	50	✓
Description	varchar	80	✓
CountryId	int	4	✓

Login Account

Column Name	Data Type	Length	Allow Nulls
LoginName	varchar	50	
Password	varchar	50	✓
Question	varchar	100	✓
Answer	varchar	80	✓
Active	tinyint	1	✓
[Date]	datetime	8	✓

User Full Information

Column Name	Data Type	Length	Allow Nulls
LoginId	varchar	50	✓
FirstName	varchar	50	✓
LastName	varchar	50	✓
Address	varchar	100	✓
City	varchar	50	✓
State	varchar	50	✓
PinCode	varchar	10	✓
Country	varchar	50	✓
Email	varchar	50	✓
Phone	varchar	20	✓
DOB	varchar	50	✓
Gender	varchar	10	✓
[Language]	varchar	50	✓
Income	varchar	50	✓
Occupation	varchar	50	✓
IndustryType	varchar	100	✓
Interest	varchar	300	✓
[Date]	datetime	8	✓

User Login History

Column Name	Data Type	Length	Allow Nulls
LoginName	varchar	50	✓
LoginDate	datetime	8	✓
LoginTime	varchar	20	✓

User Logout History

Column Name	Data Type	Length	Allow Nulls
LoginName	varchar	50	✓
LogoutDate	datetime	8	✓
LogoutTime	varchar	20	✓

Process Logic of each Module

Login

This is a very first module in my project:

1. Enter username and password.
2. If username and password will exists in the table.
3. The software will open.
4. Else you got an error message.

New User

This is new user module:

1. Enter the username and password.
2. Save the records
3. Data will save in the login table permanently.
4. Second time you can enter with your own username and password.

Member Registration details:

Registration module is responsible for member registration and login. While registration, member will be prompted for his 1) login account details (username, password, hint question, answer), 2) his personal details, and 3) his contact address

At time of sign in, Member has to provide username and password.

In Message compose box, Member has to provide Message to send with Email-ID (to whom message has to be sent.).

Outputs:

- On successful registration, member will be provided confirmation mail.
- On successful signing in, member will be placed to My Account page.

The following commands specify access control identifiers and they are typically used to authorize and authenticate the user (command codes are shown in parentheses)

USER NAME (USER)

The user identification is that which is required by the server for access to its file system. This command will normally be the first command transmitted by the user after the control connections are made (some servers may require this).

PASSWORD (PASS)

This command must be immediately preceded by the user name command, and, for some sites, completes the user's identification for access control. Since password information is quite sensitive, it is desirable in general to "mask" it or suppress type out.

TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

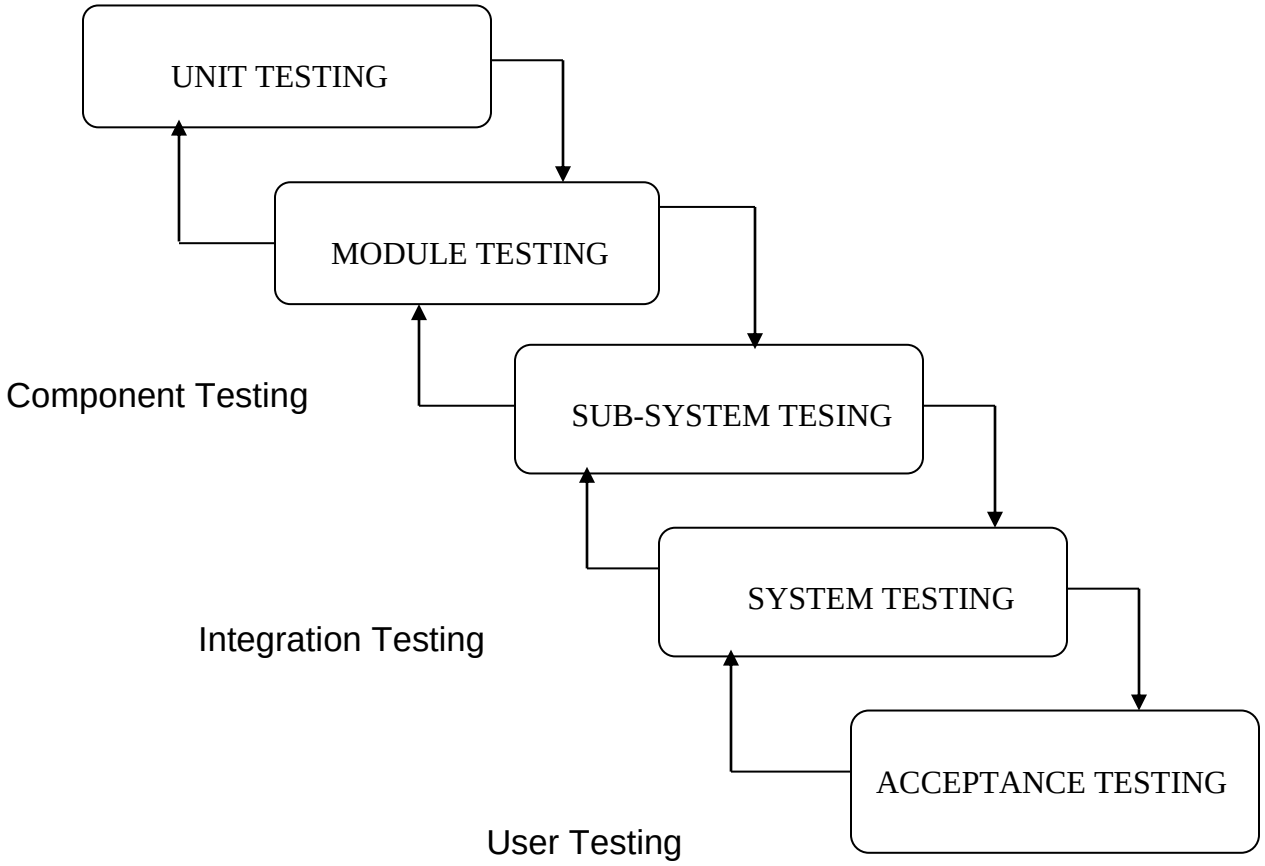
A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation

criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.



Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

Implementation Methodology

Alchemy Software Technologies has worked out the methodology of project implementation. Its basic premise is effective communication between client and contractor. A key element is, therefore, designation of competent and communicative project managers, on both sides, who will be responsible for co-operation and implementation of the project and its changes.

Implementation system of the project involves the following steps:

- **Stage I**: Preparatory work:
  Selection of project team
  Analysis of the operations and activities within the project,
  Development of a project schedule, including a schedule of implementation,
  Determination of a risk in the project,
  Feasibility test,
  Defining a control plan
  Preparation of the training system (implementation and evaluation training which will improve the quality).
-
- **Stage II**: Implementation of the project:
  Product training for project contractors,
  substantive training for project contractors,

Simulation of the conditions for the project - test
Competency test,

- **Stage III**: Project finalization :
  Servicing the communication part according to the agreed model
  Quality control by the project team members,
  evaluation,
  Implementation of changes
  Reporting according to the established model,
  Control of project assumptions,
  Control of project communication,
  Control of changes,
  Control of risk management
  Updates of control plans.

- **Stage IV**: Evaluation of the project after the first month of cooperation:
  Listing of the conclusions driven out of all the control activities in accordance with the agreed plan
  The auditors' conclusions
  Conclusions from all the agreements pertaining to changes on both sides of the contract,
  Ensuring client's satisfaction

The proposed methodology is a proven model based on company's experience, in line with global trends in the field of project management.

List of Report Generation

It is designed by the Seagate Software Inc USA. **Crystal report** provides data access from lot of data source like Access, Oracle, Dbase, Sybase, Paradox and Sql Server. **Crystal report** can publish report to the web in a variety of formats. **Crystal report** can be integrated with Visual Basic, Delphi and ASP like applications.

Crystal report access data through-

1. Direct database file
2. ODBC
3. **Crystal report** Query designer file
4. **Crystal report** Dictionary file

Features of Crystal report

- Crystal report is an excellent reporting tool, which can be used to prepare efficient & professional looking reports in few interactive steps.
- Crystal report now in its 8'Th major revision, remains the market leader & defects standard for business and corporate report writing.
- Crystal report is now bundled over 150 leading software packages, including Microsoft Back office& Visual studio packages together with ERP & accounting packages from different vendors.
- Crystal report developer interface have been designed to work with most popular windows development tools.

Reports

- Member Report
- Sending Mail Report
- Receiving Mail Report
- Address Book Report

Overall network architecture

Three Tier Network Architecture of the Project

3-tier application is a program which is organized into three major disjunctive tiers on layers. Here we can see that how these layers increase the reusability of codes.

These layers are described below.

1. Application layer or Business layer
2. Business layer
   a. Property layer(Sub layer of business layer)
3. data layer

Advantages of three Tier Architecture.

The main characteristic of a Host Architecture is that the application and databases reside on the same host computer and the user interacts with the host using an unfriendly and dump terminal. This architecture does not support distributed computing (the host applications are not able to connect a database of a strategically allied partner). Some managers found that developing a host application take too long and it is expensive. Consequently led these disadvantages to Client-Server architecture.

Client-Server architecture is 2-Tier architecture because the client does not distinguish between Presentation layer and business layer. The increasing demands on GUI controls caused difficulty to manage the mixture of source code from GUI and Business Logic (Spaghetti Code). Further, Client Server Architecture does not support enough the Change Management. Let suppose that the government increases the Entertainment tax rate from 4% to 8 %, then in the Client-Server case, we have to send an update to each clients and they must update synchronously on a specific time otherwise we may store invalid or wrong information. The Client-Server Architecture is also a burden to network traffic and resources. Let us assume that about five hundred clients are working on a data server then we will have five hundred OLEDB/SQL CLIENT connections and several ruffian record sets, which must be transported from the server to the clients (because the Business layer is stayed in the client side). The fact that Client-Server does not have any caching facilities like in ASP.NET, caused additional traffic in the network. Normally, a server has a better hardware than client therefore it is able compute algorithms faster than a client, so this fact is also an additional pro argument for the 3.Tier Architecture. This categorization of the application makes the function more reusable easily and it becomes too easy to find the functions which have been written previously. If programmer wants to make further update in the application then he easily can understand the previous written code and can update easily.

Application layer or Presentation layer

Application layer is the form which provides the user interface to either programmer of end user. Programmer uses this layer for designing purpose and to get or set the data back and forth.

Business layer

This layer is a class which we use to write the function which works as a mediator to transfer the data from Application or presentation layer data layer. In the three tier architecture we never let the data access layer to interact with the presentation layer.

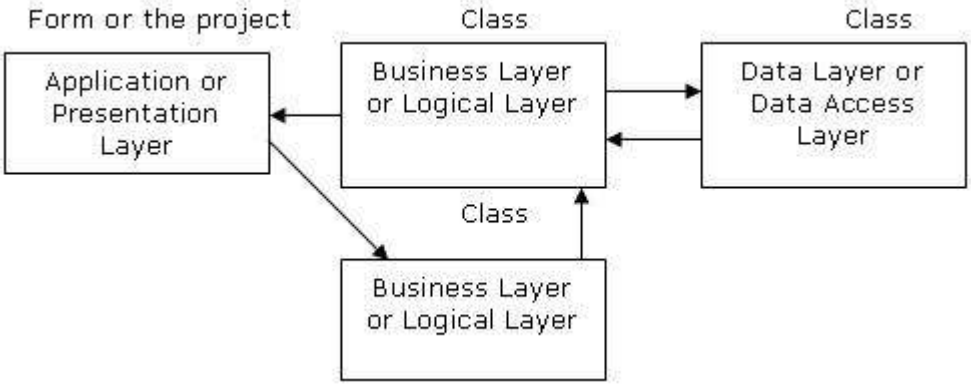
a. Property Layer

This layer is also a class where we declare the variable corresponding to the fields of the database which can be required for the application and make the properties so that we can get or set the data using these properties into the variables. These properties are public so that we can access its values.

Data Access Layer

This layer is also a class which we use to get or set the data to the database back and forth. This layer only interacts with the database. We write the database queries or use stored procedures to access the data from the database or to perform any operation to the database.

Summary



- Application layer is the form where we design using the controls like textbox, labels, command buttons etc.
- Business layer is the class where we write the functions which get the data from the application layer and passes through the data access layer.
- Data layer is also the class which gets the data from the business layer and sends it to the database or gets the data from the database and sends it to the business layer.
- Property layer is the sub layer of the business layer in which we make the properties to send or get the values from the application layer. These properties help to sustain the value in an object so that we can get these values till the object destroys.

Limitation of the Project

- Need .net framework and Ms-Sql server for execute program.
- Financial data of the company are not calculated.

Validation Checks

Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirement.

Validation is to check that “the product made is what was required or not”

Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. Black Box Testing techniques are used exclusively during validation.

After each validation test case has been conducted, one of two possible conditions exists:

- The function or performance characteristics conform to specification and are accepted.

- A deviation from specification is uncovered and a deficiency list is created. Deviation or error discovered at this stage in a project can rarely be corrected prior to scheduled completion. It is often necessary to negotiate with the customer to establish a method for resolving deficiencies.

Software Requirement Specifications

SRS Document

It is a reference document or contract between the customer and the development team. Once the customer agrees to the SRS document the development team proceeds to develop the product conforming to all the requirements mentioned in the SRS document.

An SRS document should clearly document the following:

1. Functional requirements of the system.
2. Non-functional requirements of the system.

3. Constraints on the system.

1. Functional requirements of the system: Each fri fi of the system can be considered as performing a transformation of a set of input data to the corresponding set of output data. The functional requirements of the system should clearly describe each of the functions that the system needs to perform along with the corresponding input and output data set.

2. Non-functional requirements of the system: Non-functional requirements deal with the characteristics of the system that cannot be expressed functionally, e.g., maintainability, portability, Usability, etc. The non-functional requirements also include reliability issues, accuracy of results, human computer interface issues, operating and Physical constraints, etc.

3. Constraints on the system: The constraints on the &u"s of the system may describe certain things that the system should or should not do.

Natures of SRS

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

1. Functionality: What the software is supposed to do?

2. External Interfaces: How does the software interact with people, the system's hardware other hardware and other software.

3. Performance: What is the speed, availability, response time, recovery time, etc., of the various software fundamentals.

4. Attributes: What are the considerations for portability, correctness, maintainability, security, reliability, etc.

5. Design constraints imposed on an implementation: Are there any required standards or effect, implementation language, policies for database, integrity resource limits, operating environment, etc.

Characteristics of a good SRS

An SRS should be

- Correct
- Unambiguous
- Complete
- Consistent
- Ranked for Importance and for Stability
- Verifiable

- Modifiable
- Traceable

Correct: There is no tool or procedure that assures correctness. If the software must respond to all button presses within 5 seconds and the SRS stated that "the software shall respond to all button presses with in 10 seconds", then that requirement is incorrect.

Unambiguous: An SRS is unambiguous if and only if every requirement started therein has only are interpretation. In cases, where a term used in a particular context could have multiple meanings, the term should be included in a glossary where its meaning is made more specific.

Complete: An SRS is complete if and only if it includes of the following elements.

1. All significant requirements, whether relating to functionality, performance, design constraints, attributes or external interfaces.
2. Full labels and references to all figures, tables and diagram in the SRS and definition of all terms and units of measure.

Consistent

An SRS is consistent if no subset of individual requirements desorbed in it conflict. There are 3 types of likely conflicts in an SRS:

1. The specified characteristics of real word objects may conflict, e.g.
    a. The format of an output report may be described in are requirements as tabular but in another as textual.
    b. One requirement may state that all lights shall be green while another states that all lights should be blue.
2. There may be logical or temporal conflict between two specified actions, e.g.
    a. Are requirement may specify that the program will add 2 inputs and another may specify that the program will multiply them.
    b. Are requirement may state that 'A' must always follows B, while another requires that A&B occur simultaneously.
3. Two or more requirements may describe the same real word object but use different terms for that object. The use of standard terminology and definitions promotes consistency.

Implementation of Security Mechanisms

The Security Measure is the most important task, which should be done with much care no need to say why this is so crucial. The security measures starts right from the selection of operating system in that case for Windows XP is best choice because it provides best security in its peer OS as we know this project is very versatile and there are variety of users who uses systems according to their needs so authentication is an major task which has to perform by the system administrator as I have already mentioned that there are different type of access permission like viewing the records modifying etc.

Another security measure I took in this project is **Username** and **Password** provide to selected staff so that no unidentified person can not access the system.

Then I switch the concept of file sharing which is an essential service in these types of companies. The person holding the same position can share particular type of file.

Future Scope and Enhancement the Application

- This System being web-based and an undertaking of Cyber Security Division, needs to be thoroughly tested to find out any security gaps.

- A console for the data center may be made available to allow the personnel to monitor on the sites which were cleared for hosting during a particular period.

- Moreover, it is just a beginning; further the system may be utilized in various other types of auditing operation viz. Network auditing or similar process/workflow based applications...

Bibliography

- Wrox C#.NET 4.0#
- Black Book C#.Net 2017
- Microsoft SQL Server 2008
- And many more internet sites.