1. Introduction

1.1 Definition

For optimal sales and marketing management processes, robust functionalities for managing logistics facilities are needed. Support for sales and marketing management helps recording and tracking materials on the basis of both quality and value.

1.2 Purpose

Sales & Marketing is an integral part of most companies. It is one of the most critical sections which have a direct impact on the profitability of the company.

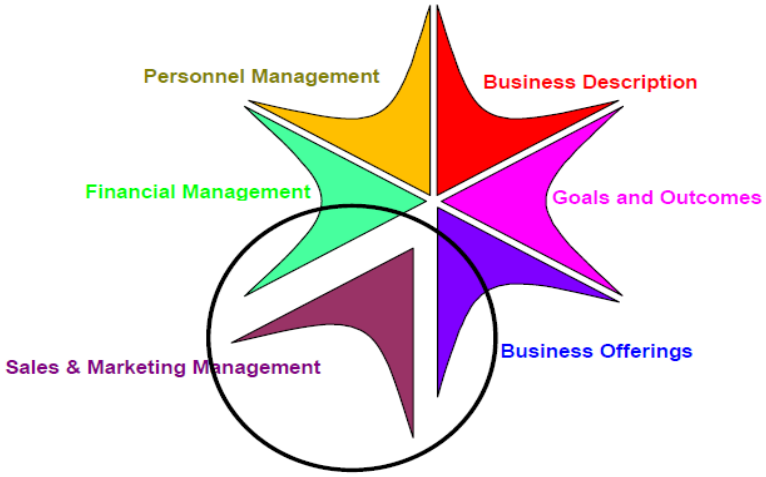

Fig. 1(a): Sales and Marketing Management System

Sales And Marketing staff has to rely mostly on the old systems or paper work to access information about the potential customer. The main perspective of this system is to solve all this problems. The system will help the Persons and Managers to manage to Sales and Marketing System in better way.

1.3 Objective

The objective of this product is to provide a Sales and Marketing Management System that will ease the communication process between Person and Manager. The system will help the Persons and Managers to manage the activities and targets in a better way. The system will be more efficient to track the progress of activities. It will be cost effective and time

saving. The system will provide more accurate data and no paper work is required. The system will also give more user-friendly environment.

1.4 Scope

This system helps in tracking records so that past records can be verified through them and one can make decisions based on the past records. This system completes the work in a very less time resulting in less time consumption and high level of efficiency.

This system is developed in such a way that even a naive user can also operate the system easily. The calculations are made very quickly and the records are directly saved into databases and the databases can be maintained for a longer period of time. Also this system provides high level of security for data. The advantages are,

- Ensure data accuracy
- Proper control of the higher authority
- Minimize manual data entry
- Greater efficiency
- Better service
- User friendliness and interactive
- Minimum time required

1.5 Front end and Back end

- Front End: Sales and Marketing Management System UI is developed using JavaFx 1.3 and CSS.
- Back End: The back end database is developed in ORACLE 10*g* and coding is developed in JavaFx 1.3 itself.

1.6 Technology and Literature Review

JavaFX is a full-fledged development platform for RIAs and has many advantages over other equivalent technologies in the market. Out of all, there are some key factors that differentiate JavaFX significantly.

- RIAs for all screens

  JavaFX provides a unified development and deployment model for building expressive RIAs across desktop, browser, mobile, and TV.

- Rich client platform

  JavaFX makes it easy and intuitive to integrate graphics, video, audio, animation, and rich text.

- Ease of use

  JavaFX Script is an easy-to-learn, easy-to-implement language that is statically typed, offering a declarative syntax that makes it easy to program in a visual context without worrying about the internals.

- Powerful runtime

  JavaFX leverages the extreme ubiquity, power, performance and security of the JRE.

- Time-to-market

  JavaFX offers a dramatically shortened production cycle for designers and developers through its designer–developer workflow. JavaFX allows you to incorporate multimedia assets from popular third-party design tools such as Adobe Illustrator and Photoshop using the JavaFX Production Suite.

- Ready-made mass market

  JavaFX allows you to distribute your RIAs widely, more quickly and easily across billions of Java-powered devices.

- Preserve your investment

  You can reuse your existing Java libraries in JavaFX and thus preserve the investment you've already made in Java.

- Cross-browser functionality

  JavaFX provides a uniform user experience across all browsers on multiple platforms.

- Enterprise Integrations

  With JavaFX you are ready to integrate a rich UI with a complex enterprise back-end.

- Proven Security Model

  JavaFX provides broader system access with the proven Java security model.

1.7 Functions and main modules of new system

In this system, there are mainly three users which are Administrator, Manager and Person. Detailed list of all functional requirement of this system performed by the three users as mentioned above are described as follows:

1.7.1 Administrator

Administrator can perform the following activities:

- Login

  Administrator can log into the system by providing their username and password. Administrator will be redirected to the home page of Administrator if given right user name and password. If the given information will not match with the database entries, then administrator will get a message regarding the failure of the login attempt.

- Add Department

  This functionality will allow administrator to add department like Sales Department or Marketing Department with its details like Description of department as mentioned in Data Dictionary (Appendix 1 – table dept_master).

- Delete/View Department

  This functionality will allow administrator to delete/view department's details as mentioned in Data Dictionary (Appendix 1 – table dept_master).

- Add Designation

  This functionality will allow administrator to add designation (like manager, person) with its details as mentioned in Data Dictionary (Appendix 1 – table design_master).

- Delete/View Designation

  This functionality will allow administrator to delete/view designation's details as mentioned in Data Dictionary (Appendix 1 – table design_master).

- Add Product

  Administrator can add products as default quantity as 0.

- Delete/View Product

  This functionality will allow Administrator to view or delete product from system.

- Add Manager

  Administrator can add Managers by adding its details as mentioned in Data Dictionary (Appendix 1 – table salesmanager_master).

- Delete/View Manager

  This functionality will allow Administrator to remove Manager. Administrator can also view the list of Manager.

- Logout

  This functionality will allow Administrator to logout from the system.

1.7.2 Manager

Manager can perform the following activities:

- Login

  Manager can log into the system by providing their username and password. Manager will be redirected to the home page of Manager if given right user name and password. If the given information does not match with the database entries, then the Manager will get a message regarding the failure of the login attempt.

- Assign Targets

  Into this functionality, Manager can assign targets to their Persons like target value and their due date for that.

- View Targets

  This functionality will allow manager to monitor their assigned targets to Persons, target due date and their current status which are achieved, not achieved and pending.

- Add Company

  This functionality will provide platform to add companies into the system with which the Managers are dealing with. Company details must be entered in the system as mentioned in Data Dictionary (Appendix 1 – table company_master)

- View/Edit/Delete Companies

  This functionality will allow Manager to view the companies and edit or update or delete the companies.

- Add Client

  This functionality will allow Manager to add clients of already added companies and its details with which the system deals with. The details of clients which must be entered are described in detail in Data Dictionary (Appendix 1 – table client_master).

- View/Edit/Delete Clients

  This functionality will allow Manager to view all the clients with which the system is dealing with. Managers can also edit or update or delete the clients.

- Add Stock

  This functionality will allow Manager to add stocks of product.

- View Stock

  This functionality will allow Manager to view all the available stocks of product.

- Add Person

  This functionality will allow Manager to add new Persons who is working under that particular Manager. Persons' details which must be entered to add new Person are described in Data Dictionary (Appendix 1 - table salesman_master).

- Delete/View Person

  This functionality will allow Manager to remove Person who is working under that Manager. Manager can also view the list of Persons.

- View Previous Sales

  This functionality will allow Manager to view previous selling of the products by Pie Chart. Managers can also select a duration for which they want to view the previous selling.

- Edit Profile Details

  This functionality will allow manager to edit their profile as personal details.

- Logout

  This functionality will allow Manager to logout from the system.

1.7.3 Person

Person can perform the following activities:

- Login

  Person can log into the system by providing their username and password. Person will be redirected to the home page of Person if given right user name and password. If the given information does not match with the database entries, then the Person will get a message regarding the failure of the login attempt.

- Add Activity

  This section will allow person to add activity like campaigns, launch of new products and seminar. Into this section, Persons have to provide details like activity name, activity for, activity due date, activity details, activity leader and selection of activity members as mentioned in Data Dictionary (Appendix 1 – table activity_master).

- View/Delete Activity

  Using this functionality person can view activities added by the person itself or other persons. Here, the person specifies the duration for which he wants to view the activities. He can also delete the activities added by him.

- Add Lead (Lead type will be Face to Face, Telephonic, Email, SMS, etc.)

  This functionality will allow person to select details like client name, details etc. This section describes type of communication by which person can communicate to the clients like as face to face, telephonic, e-mail or SMS etc.

- View/Delete Lead

  This functionality will allow person to view leads (type of communication) added by the person itself or other persons. Here, the person specifies the duration for which he wants to view the leads. He can also delete the leads added by him.

- New Sell

  This functionality will allow person to add new selling details. Here the person will select client and add the number of product he sold to the client and he can add the value in the database.

- View Target

  This functionality will allow person to view assigned target to him by manager.

- Edit Profile Details

  This functionality will allow person to edit their profile as personal details.

- Logout

  This functionality will allow person to logout from the system.

2. Project Management

2.1 Project Plan

2.1.1 Project Development Approach

- **Development Strategy: Waterfall Method**

  Waterfall Model has been adopted as an approach for development of the system. Most of the requirements of the project are fixed and already thought of; very less functionality updating is expected in future. So Waterfall model is the right approach for our project.

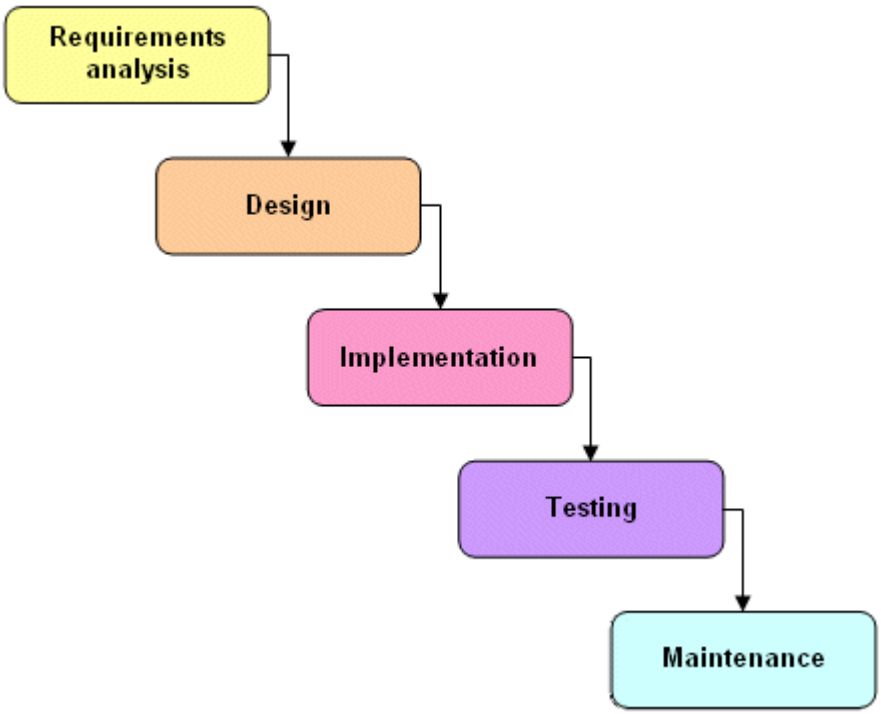

Fig. 2(a): Waterfall Model

Justification:

As waterfall model is used for the system whose requirements and functionalities are already known and as our system falls in this category we have chosen waterfall model as an approach to develop our project.

2.1.2 Planning of Project

The planning of this project has followed the approach of Waterfall Development. The first phase being of Analysis and Requirement gathering deals with analysing the project and preparing a system requirement specification along with software project management and planning. Then comes the designing phase in which the various UML diagrams are made for our better understanding. The basic prototype of our database design is also prepared.

2.2 Milestones and Deliverables

- **Milestones**
  - Analysis & requirement gathering.
  - Learning all the stuff which required to completing this project.
  - Designing – UI.
  - Designing – Database.
  - Coding.
  - Testing.

- **Deliverables**
  - **Software Requirement Specification.**

    The SRS document states in precise in explicit language those functions and capabilities a software system must provide. Also states any required constraints by which the system must abide. It contains all the necessary functional and non-functional requirements of the application. SRS accomplishes four major goals:

    1. It provides feedback to the customers.
    2. It decomposes the problems into different component parts.
    3. It serves as an input to the design specification.
    4. It serves as a product validation check.
    5. The SRS document contains all diagrams (use cases, sequence, activity and class), functional and non-functional requirements for the website.

- **Software Project Management Plan.**

  Software project management is the art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, monitored and controlled. Listing the details of the major functions, project resources, staff representations and scheduling.

- **Analysis and design of the System.**

  Analysis and Design contains the design issues and the UI diagrams. The design issues for any application are:

  1. Page load efficiency.
  2. Simplicity.
  3. Use the space wisely.

- **Project Report.**

  A full project report which enlightens the details of the project.

- **Working System.**

  A fully functional system at the end of the year.

2.3 Roles and Responsibilities

We are two project partners so divide works equally.

Role and Responsibilities of member-1 is:

- Database design and creation
- Implement registration and login functionality
- Checking the validation functionalities of Registration and Login
- Implement Manager module designing and its content
- Checking the validation functionalities of Manager module
- Testing of modules

Role and Responsibilities of member-2 is:

- UI Design
- Implement Admin module designing and its content
- Checking the validation requirements of Admin module
- Implement Person module designing and its content

- Checking the validation requirements of Person module
- Testing of modules

2.4 Project Scheduling

Project: Sales and Marketing Management System

Current Week															
Weeks **Tasks**	01/05 to 01/12	01/12 to 01/19	01/19 to 01/26	01/26 to 02/02	02/02 to 02/09	02/09 to 02/16	02/16 to 02/23	02/23 to 03/02	03/02 to 03/09	03/09 to 03/16	03/16 to 03/23	03/23 to 03/30	03/30 to 04/06	04/06 to 04/13	04/13 to 04/20
Database design and creation															
UI Design															
Implement registration and login functionality and validation															
Implement Admin module designing and its content															
Validation requirements of Admin module															
Implement Manager module designing and its content															
Checking the validation functionalities of Manager module															
Implement Person module designing and its content															
Checking the validation requirements of Person module															
Testing of modules															
Module Integration															
Tasks Weeks	01/05 to 01/12	01/12 to 01/19	01/19 to 01/26	01/26 to 02/02	02/02 to 02/09	02/09 to 02/16	02/16 to 02/23	02/23 to 03/02	03/02 to 03/09	03/09 to 03/16	03/16 to 03/23	03/23 to 03/30	03/30 to 04/06	04/06 to 04/13	04/13 to 04/20
Current Week															

Fig. 2(b): Timeline Chart

3. System Requirements Study

3.1 User Characteristics

There are mainly three users Administrator, Manager, Person in this system.

- Administrator has high end priorities and able to manage whole system. Such as administrator can add department in system, add managers of the system and manage products of the system.
- Manager has no administrative rights and manager can assign targets to their persons and can view previous sales. Manager also adds the stock of product.
- Person has low end priorities and not able to access administrative as well as managerial rights. Person has rights to manage activities and leads which are describe in functionality. Persons add selling details and completes their targets.

3.2 Hardware & Software Requirements

3.2.1 Hardware Interfaces

3.2.1.1 Server Environment

Table 1: Server Environment for Hardware Interfaces

Processors:	Intel Pentium 4, Intel Centrino, Intel Core Duo (or compatible) 1.8 GHz or Intel Xeon minimum Dual-Core Intel, PowerPC G5
Memory:	1 GB of RAM (2 GB recommended)
Disk Space:	256 MB of free disk space
Screen Resolution:	1024x768 pixels (1280x800 recommended) with 16-bit video card

3.2.1.2 Client Environment

Table 2: Client Environment for Hardware Interfaces

Processors:	Intel Pentium 4, Intel Centrino, Intel Core Duo (or compatible) 1.8 GHz or Intel Xeon minimum Dual-Core Intel, PowerPC G5

Memory:	512 MB of RAM (1 GB recommended)
Disk Space:	30 MB of free disk space
Screen Resolution:	1024x768 pixels (1280x800 recommended) with 16-bit video card

3.2.2 Software Interfaces

3.2.2.1 Server Environment

Table 3: Server Environment for Software Interfaces

Operating System:	Windows XP with Service Pack 3 or Windows Vista (Home Premium, Business, Ultimate, or Enterprise) or Windows 7 minimum, Mac 10.4 minimum, Ubuntu 10.6 minimum
JAVA Environment:	For Windows JDK 6 Update 13 minimum (JDK 6 Update 14 recommended) For Mac JDK 5 Update 16 minimum

3.2.2.2 Client Environment

Table 4: Client Environment for Software Interfaces

Operating System:	Windows XP with Service Pack 2 or Windows Vista (Home Premium, Business, Ultimate, or Enterprise) or Windows 7, Mac 10.4 minimum, Ubuntu 10.6 minimum
JAVA Environment:	For Windows JDK* 6 Update 13 minimum (JDK 6 Update 14 recommended) or JRE 1.7.0 Update 4 minimum For Mac JDK* 5 Update 16 minimum or JRE 1.7.0 Update 4 minimum * The JDK installation includes the Java Runtime Environment (JRE).

3.3 Constraints

- **Hardware or Software limitations:**

  Intel 1.8 GHz processor and JAVA environment required.

- **Language:**

  JavaFx language is used as implementation constraints.

- **Tools:**

  Developers can use their favourite Java development tools, such as the NetBeans (6.8 recommended) and Eclipse IDEs, for development of JavaFx applications.

- **Interfaces to other applications:**

  Application specific Plug-In is required to interface with other applications.

- **Database:**

  No restrictions are there because any type of database can be integrated with system and in this system Oracle database will be used.

4. System Analysis

4.1 Study of Current/Existing System

- Sales And Marketing staff has to rely mostly on the old systems or paper work to access information about the potential customer. Research has proved that timely and latest information about the potential customer is key of getting the business done (closing the deal).
- Sales & Marketing division has to be well connected with the other departments of the company. They need better systems to identify potential customer. They should be able to contact and perform regular follow-ups with the potential customer. Information about this customer should be available to other divisions of the company as soon as the order is received.
- Sales & Marketing staff cannot get all the required lead information as per their needs.
- Sales & Marketing through traditional channels is getting very expensive in the long run.
- Sales and Marketing staff cannot compare past records with the current one.
- Sales and Marketing staff does not have effective direct mechanism to access consumer data.
- Sales and Marketing information available is scattered everywhere in traditional system. There is no central place where one can go and access all the required records.

4.2 Problem and Weaknesses of Current System

- Time consuming
- Less accurate
- Less efficient
- Slow data processing
- Not user friendly environment
- Difficult to keep old records

4.3 Requirements of New System

4.3.1 User Requirements

- For every user of this system including administrator have to login into the system. After that, Administrator can manage department, designation and the whole system.
- Manager can manage targets, companies and can view activity and lead whereas Person can add, view or update activity and lead. Modules are abstracted uniquely.
- There will be user friendly error handling and validation.
- The design or layout of every form will be very clear and very interactive to the user.
- When the user open the software the welcome window will appear.
- This software will be easily understandable and operable by the user.
- All pages of the system are following a consistent theme and clear structure.
- The occurrence of errors should be minimized through the use of combo boxes, radio buttons and scroll down in order to reduce the amount of text input from user.
- After registration user will be able to login and then he/she would be able to view, edit, and update his /her own profile.

4.3.2 System Requirements

4.3.2.1 Hardware Requirements

Processor : Pentium 4 (P4) / AMD Athlon

RAM : 512 MB

Hard Disk	:	20 GB
FDD	:	1.44 MB
Keyboard	:	108 Keys
Mouse	:	3 Button scroll
Monitor	:	VGA/SVGA 14 inch (1024x768 pixels)
CD Drive	:	52 X

4.3.2.2 Software Requirements

OS	:	Any
Language	:	JavaFx
Front End	:	JavaFx
Back End	:	Oracle 10*g*
Platform	:	JavaFx 1.3

4.3.2.3 Communications Interfaces

Following communication interfaces are used in system.

System	:	Any Java supported System can run JavaFX system.

4.4 Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during

The feasibility study of this project such as technical, economic and operational feasibilities. The following are its features:

1. TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

2. ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

3. BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

5. System Design

5.1 Design Methodology

Here in this project, I used WATER FALL MODEL to complete this project. The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/ Implementation, and Maintenance.

It states that the phases are organized in a linear manner. There are variations of the waterfall model depending on the nature of activities and the flow of control between them. In a typical model, a project begins with feasibility analysis. On successfully demonstrating the feasibility of a project, the requirement analysis and project planning begins. The design starts after the requirements analysis is complete, and coding begins after the design is complete. Once the programming is completed, the code is installed.

After this, the regular operation and maintenance of the system takes place. With the waterfall model, the sequence of activities performed in a software development project is: requirement analysis, project planning, system design, detailed design, coding and unit testing, system integration and testing. There are two basic assumptions for justifying the linear ordering of phases in the manner proposed by the waterfall mode.

The various phases in this model are:

- Requirement analysis
- Design
- Implementation
- Testing
- Deployment
- Maintenance

5.2 Activity Diagram

5.2.1 Activity Diagram of Administrator

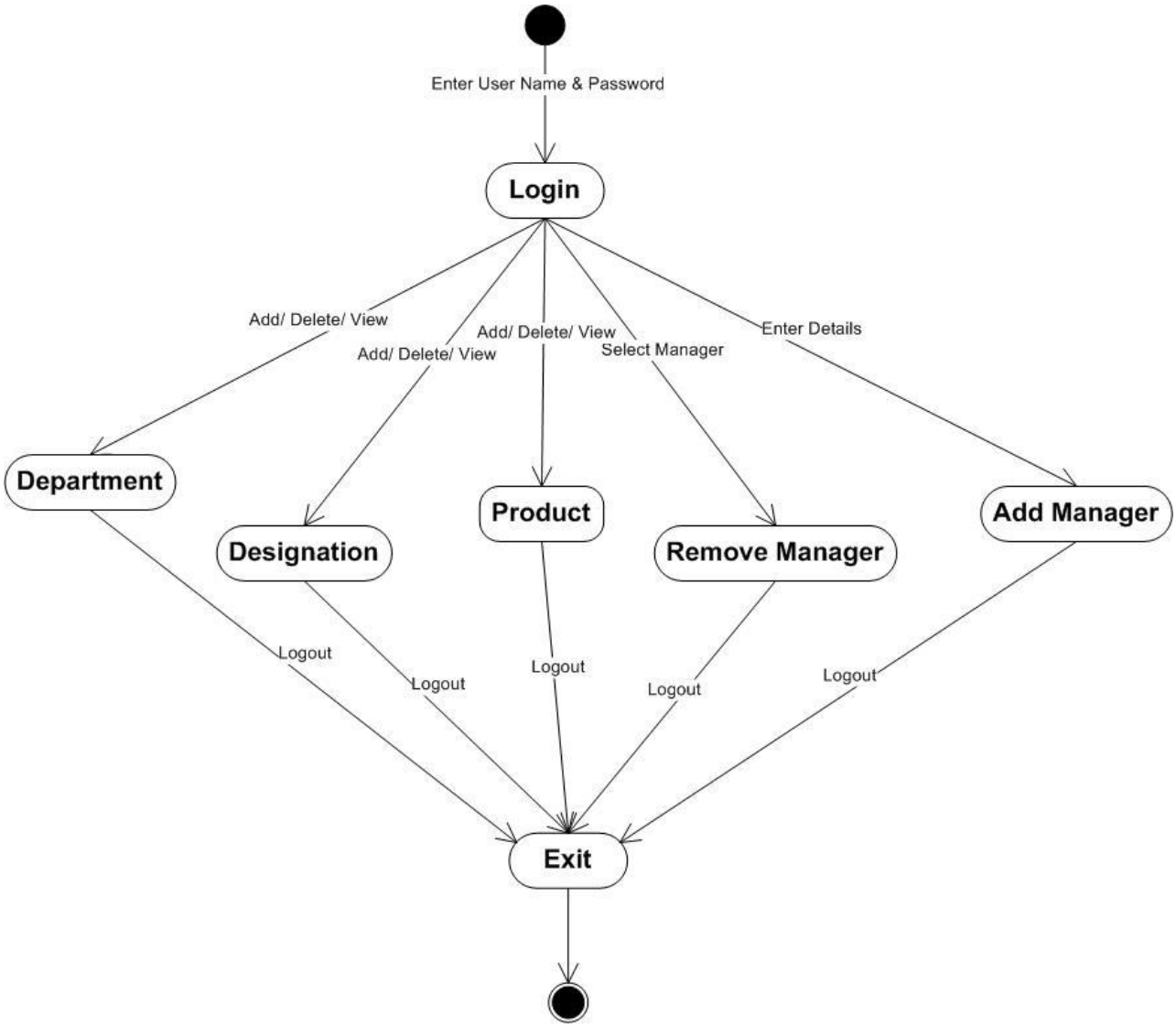

Fig 5(a): Activity diagram of Administrator module

5.2.2 Activity Diagram of Manager

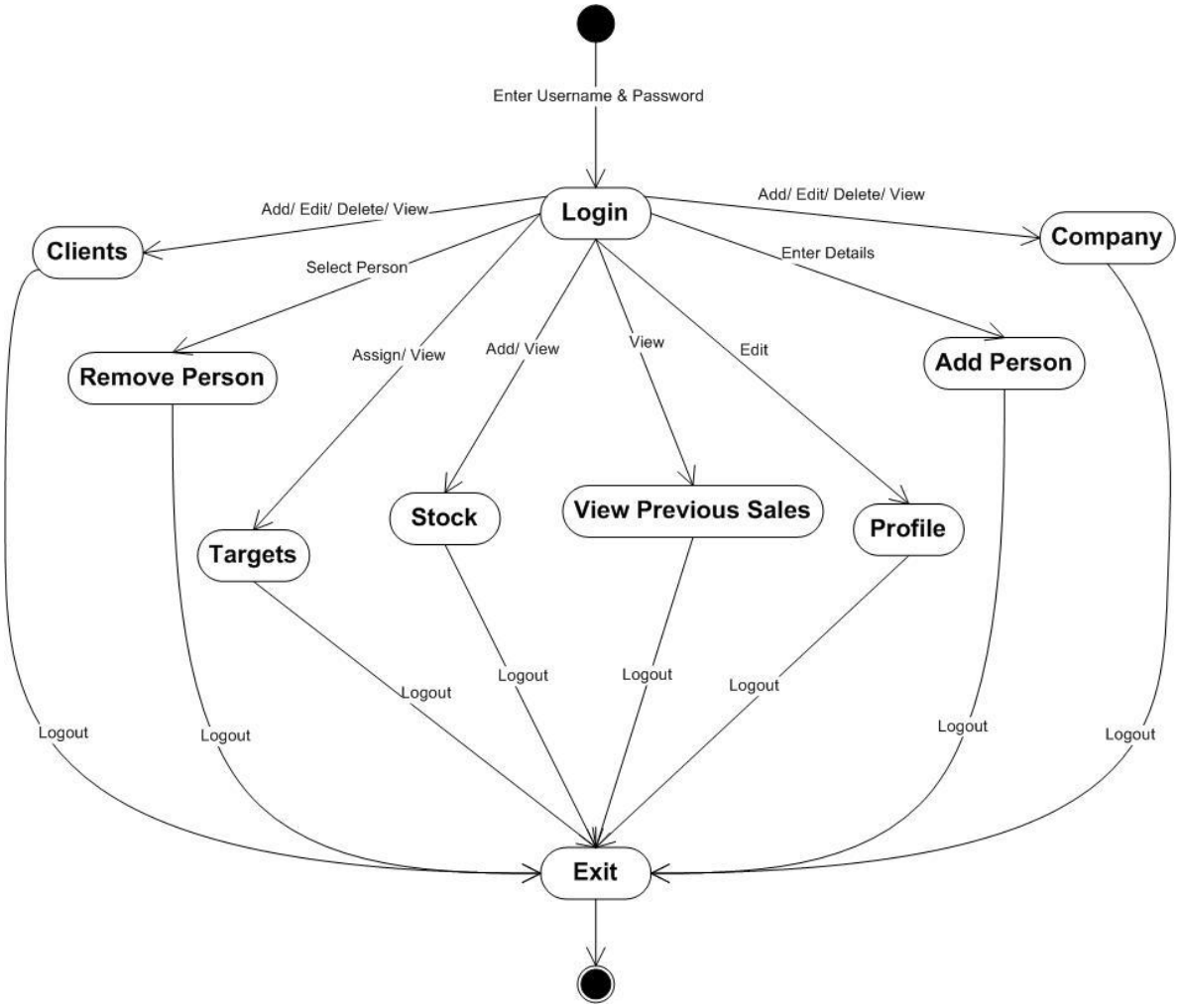

Fig 5(b): Activity diagram of Manager module

5.2.3 Activity Diagram of Person

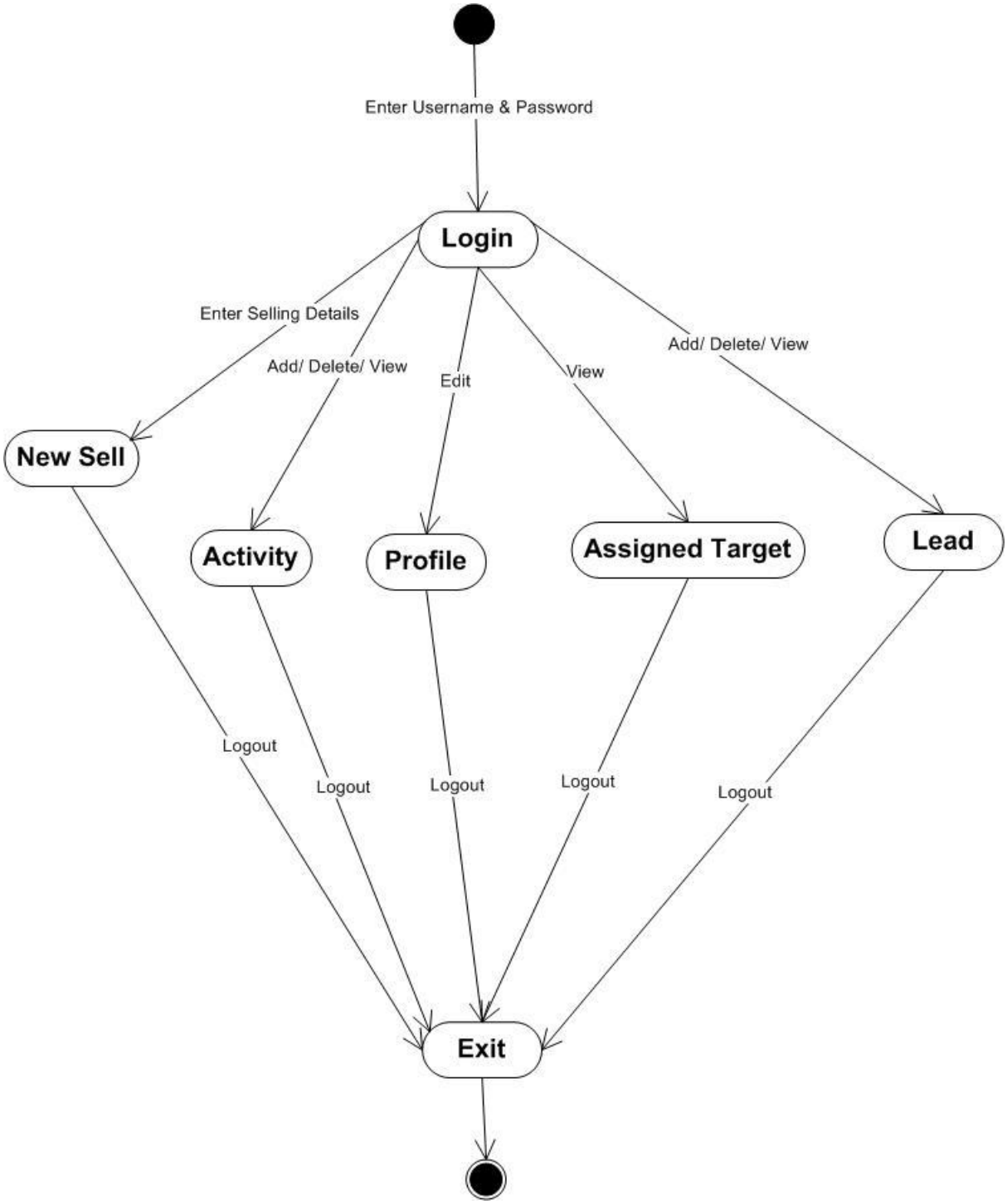

Fig 5(c): Activity diagram of Person module

5.3 Sequence Diagram

5.3.1 Sequence Diagram of Administrator

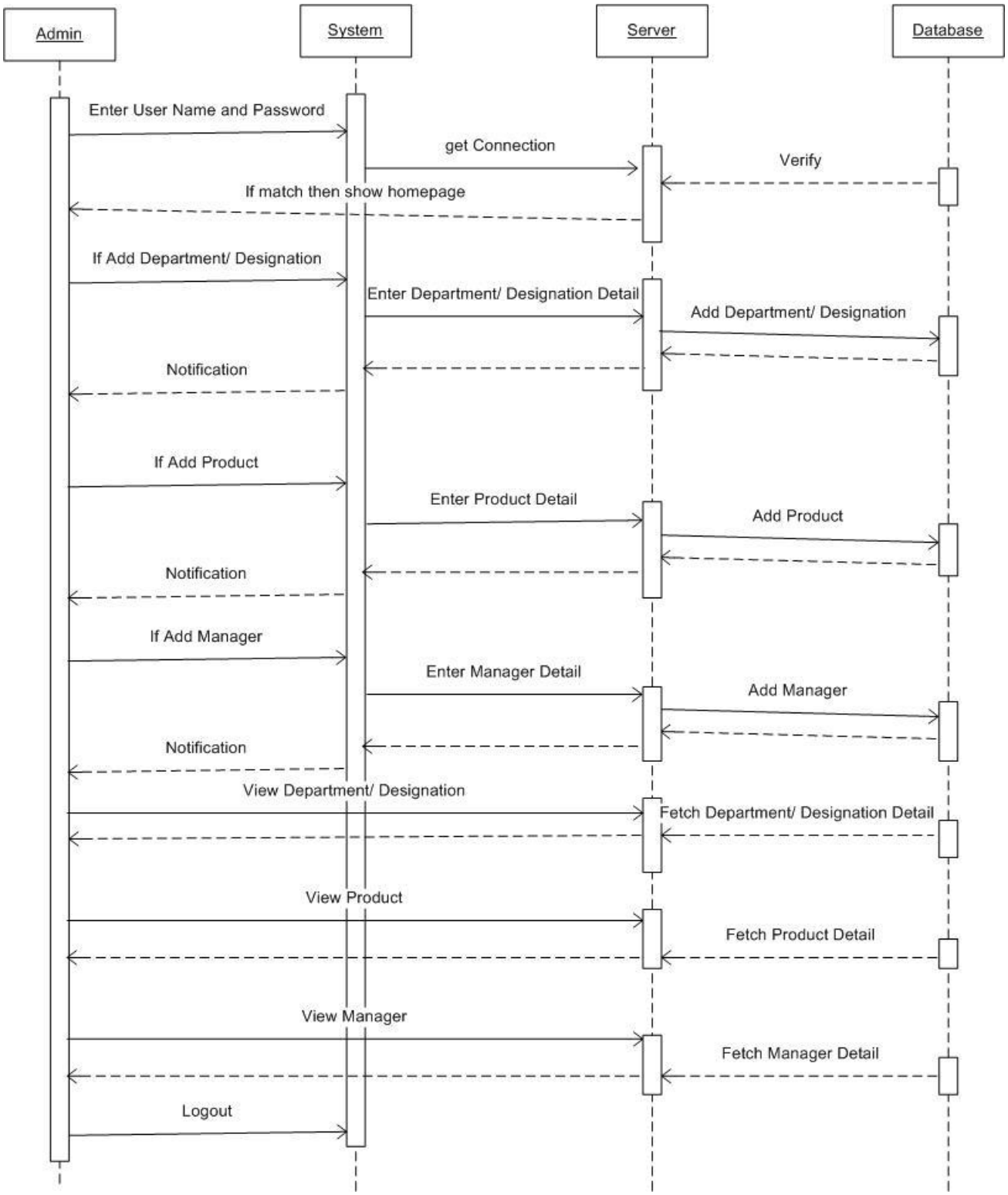

Fig 5(d): Sequence diagram of Administrator module

5.3.2 Sequence Diagram of Manager

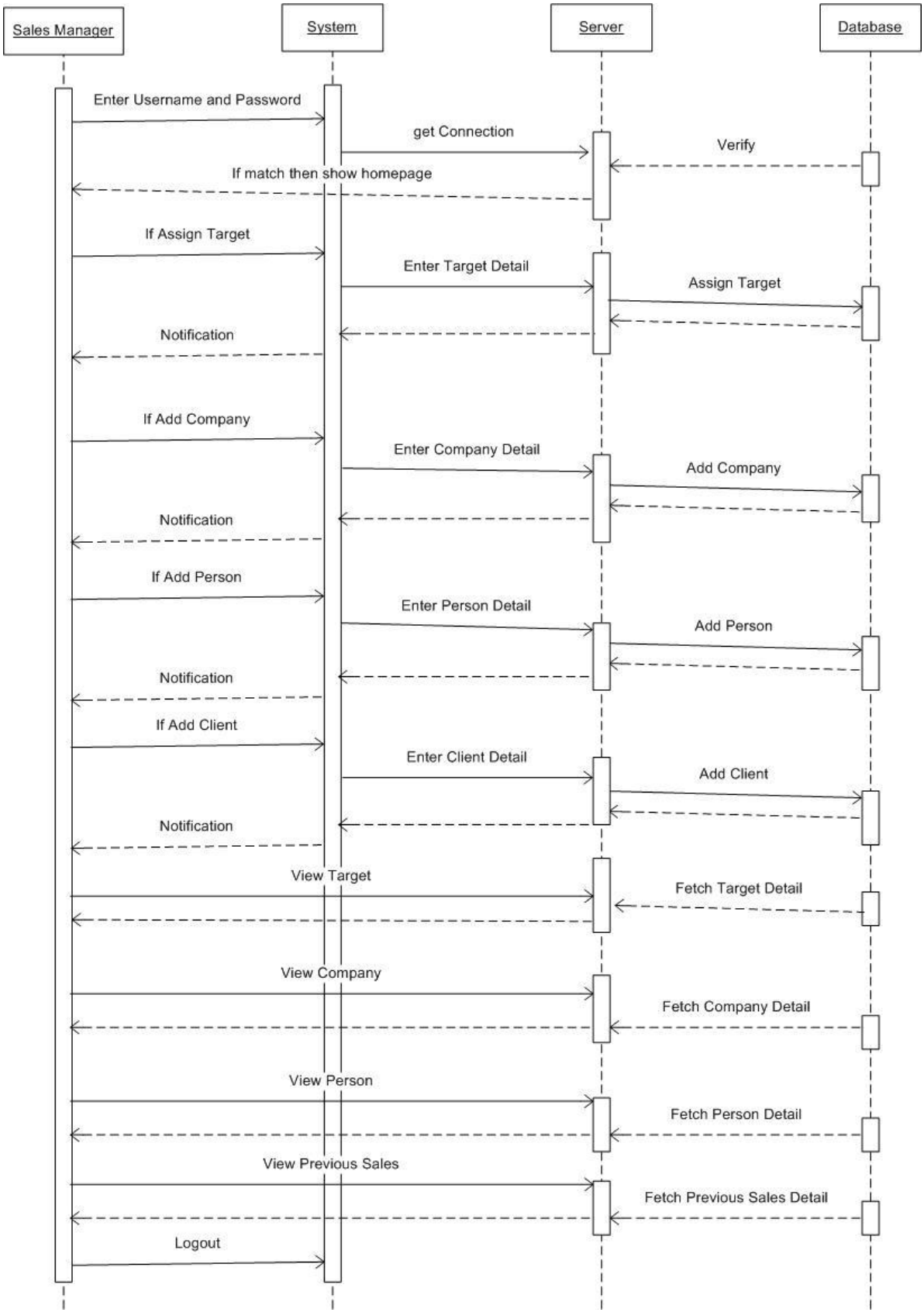

Fig 5(e): Sequence diagram of Manager module

5.3.3 Sequence Diagram of Person

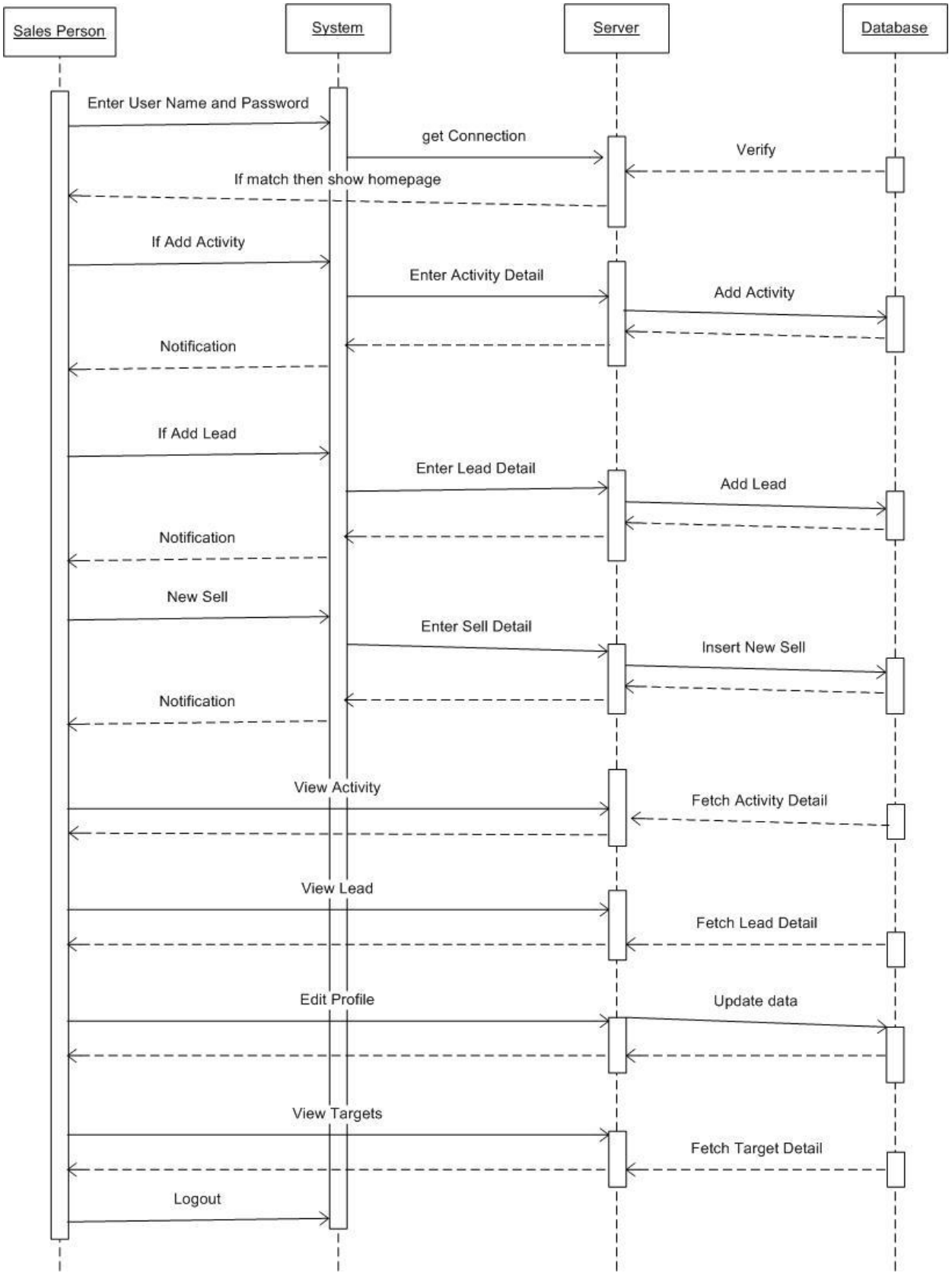

Fig 5(f): Sequence diagram of Person module

5.4 Class Diagram

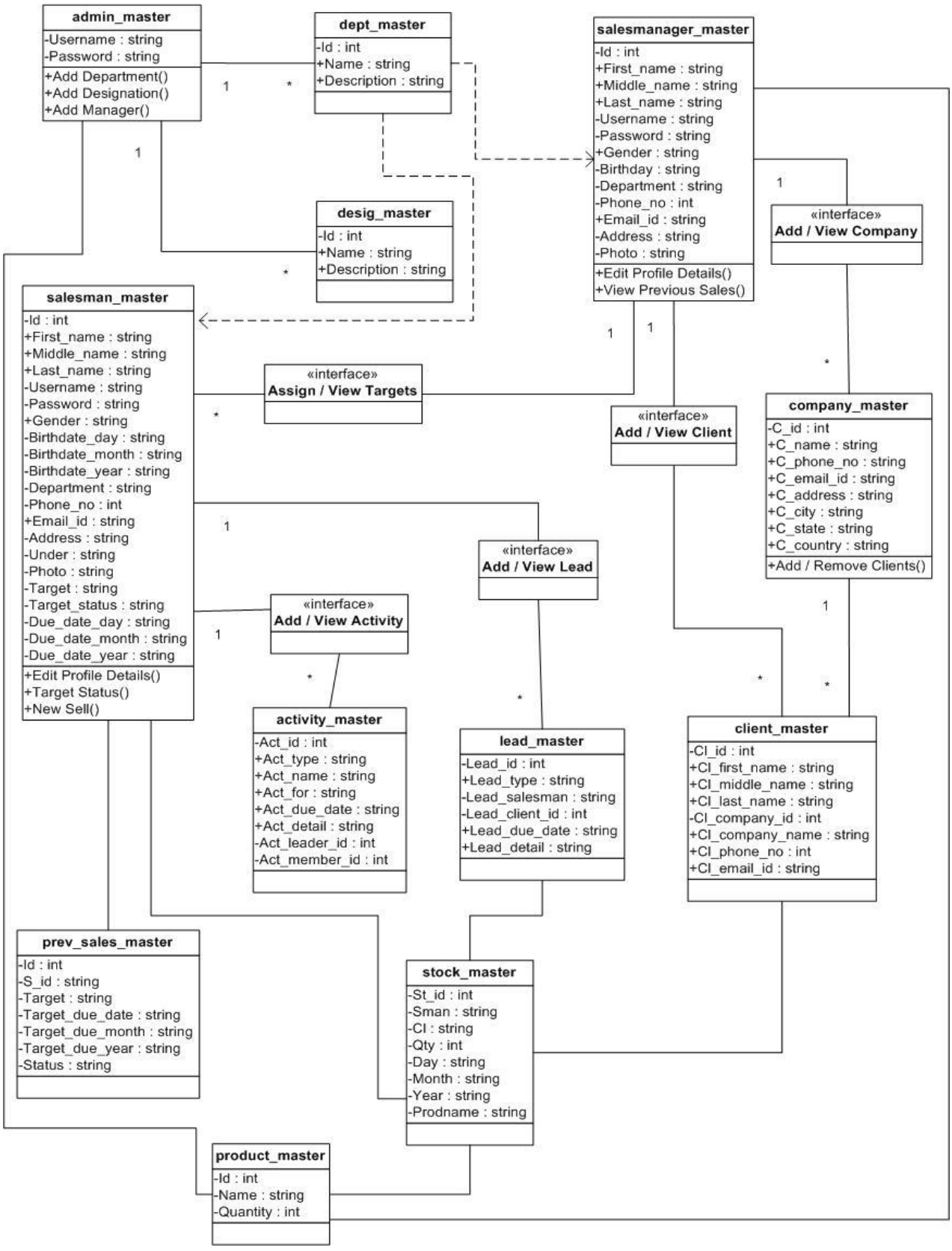

Fig 5(g): Class Diagram

5.5 Use Case Diagram

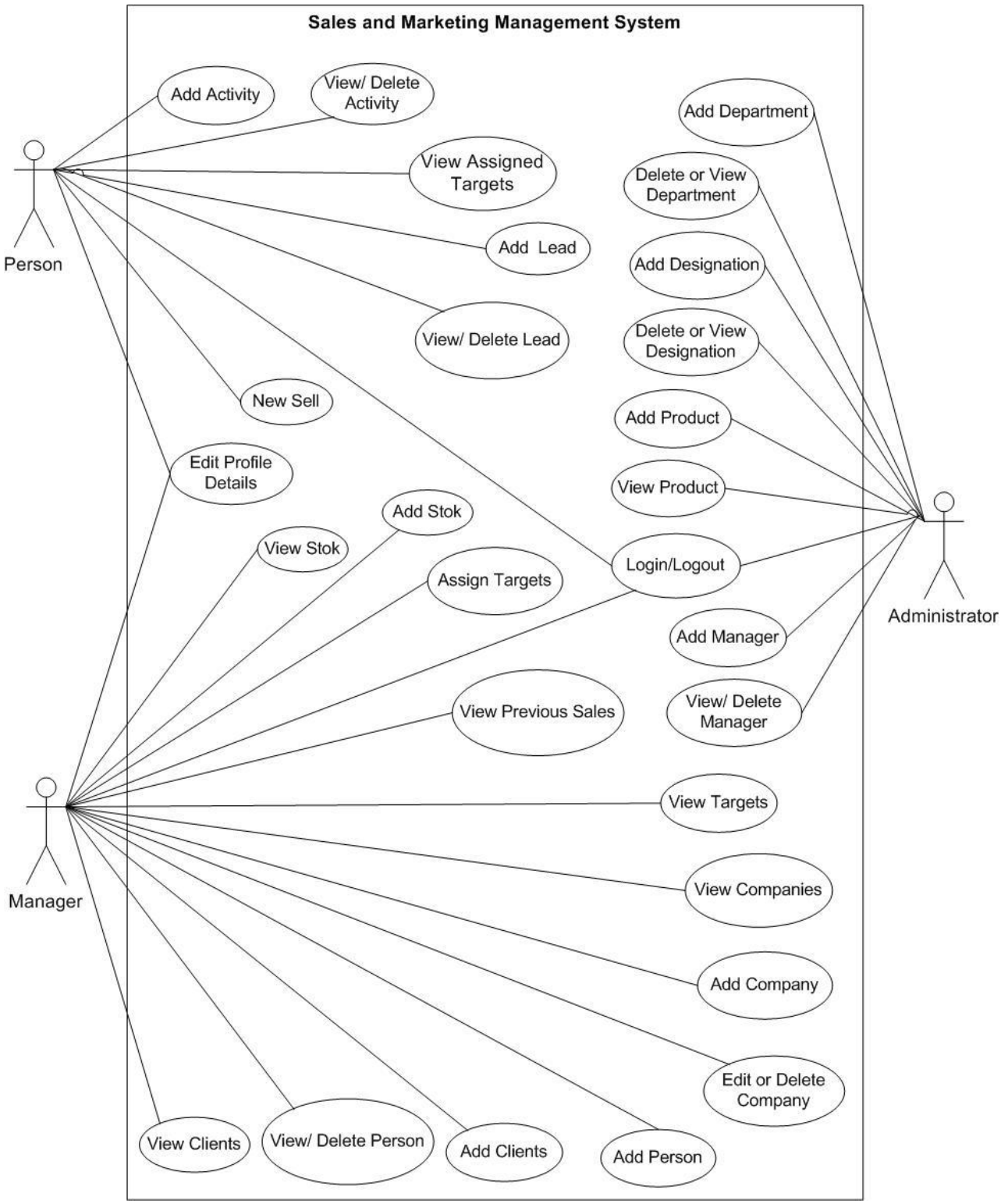

Fig 5(h): Use Case Diagram

6. System Architecture

6.1 Types of Architecture

JavaFX applications run under the control of the JavaFX runtime. At the time of this writing, there are three versions of the runtime: one for desktop environments, which runs on top of Java SE; another for mobile devices, which runs on Java ME; and a third version that runs on JavaTV. When you download JavaFX, all three versions are installed, and the correct version is automatically selected when you compile and test your applications in the NetBeans or Eclipse IDE.

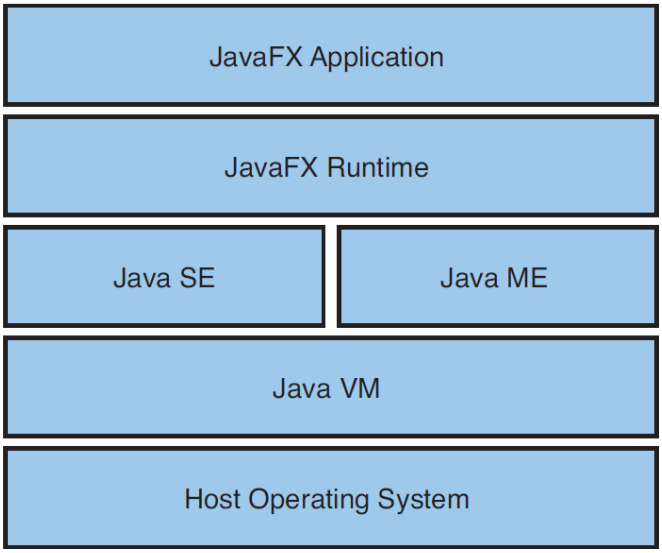

Fig 6(a): JavaFX Platform Architecture

The JavaFX runtime insulates the application from the details of the underlying Java platform, which means that it is possible, for the first time, to write a single application that can be deployed in multiple environments with little or no change. In other words, with proper use of the JavaFX APIs, you can write an application that will run on a desktop, in a browser (as an applet), on a cell phone (as a MIDlet), or on suitably equipped TVs.

JavaFX is extension of Java so the architecture of JavaFX is all same as Java platform. The JDBC architecture consists of two layers: first, the JDBC API, which supports Java application-to-JDBC Driver Manager communications and secondly the JDBC Driver API, which is handles Driver Manager-to-Database communications. The figure below shows the structure of the main interfaces and classes within JDBC and how JDBC programs interact with databases.

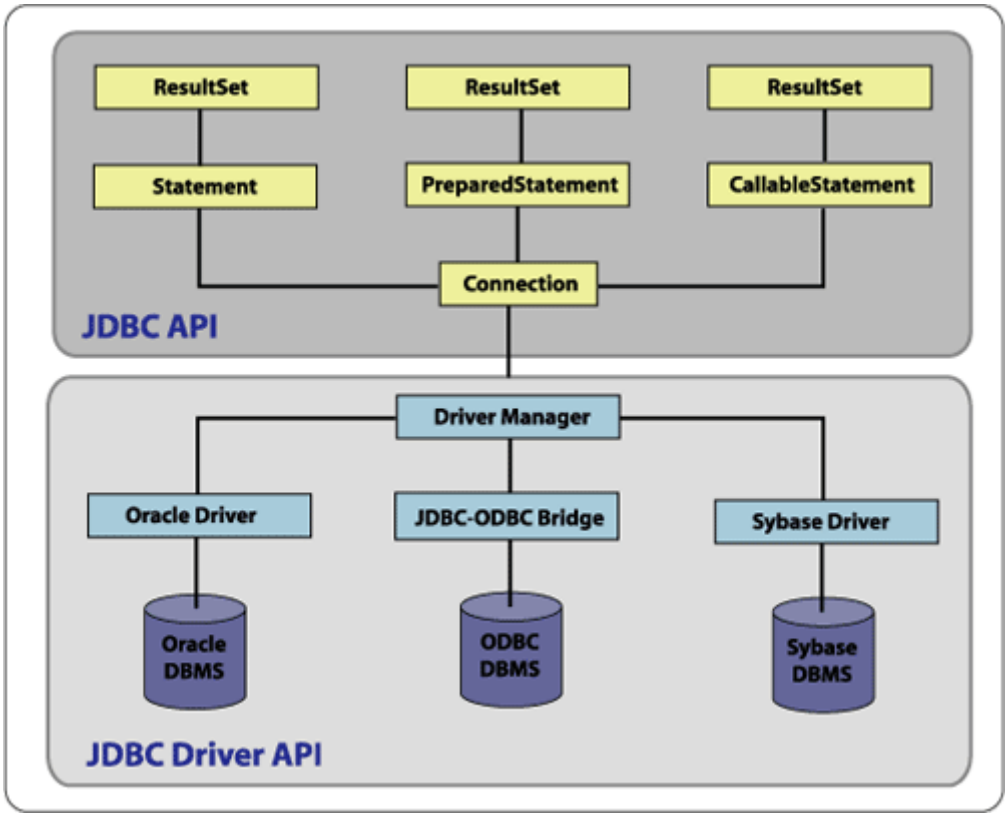

Fig 6(b): JDBC Architecture

As in ODBC, JDBC has an underlying driver manager that takes care of bridging the application with the DBMS. The driver manager supports multiple drivers connecting to many databases. The drivers themselves can be written either in Java or by using native methods. A driver written purely in Java can be used in any platform by downloading or being a part of an applet for example. A driver written using native methods gets tied up to the underlying platform in which the application runs or the applet runs.

JDBC works under two models, the two-tier model and the three-tier model.

6.2 Description of roles of each tier

Here in this system, I used "Three Tier Architecture". The client communicates with the business tier running on the server either directly.

- **Two Tier Model**

  In this model, the application communicates directly with the database as shown in the figure below.

  The client consists of the application and the JDBC driver(s). The client takes the responsibility of presentation, business logic, transaction management and resource management. The JDBC driver also resides on the client. This driver receives the request from the application and performs the necessary transformation (ie. transforms the request to vendor specific database calls). This transformed request is passed to the DBMS directly. Because the application and the driver reside on the same machine, the connection between the two is direct. The server is the DBMS, which receives the request from the client in a format it can understand.

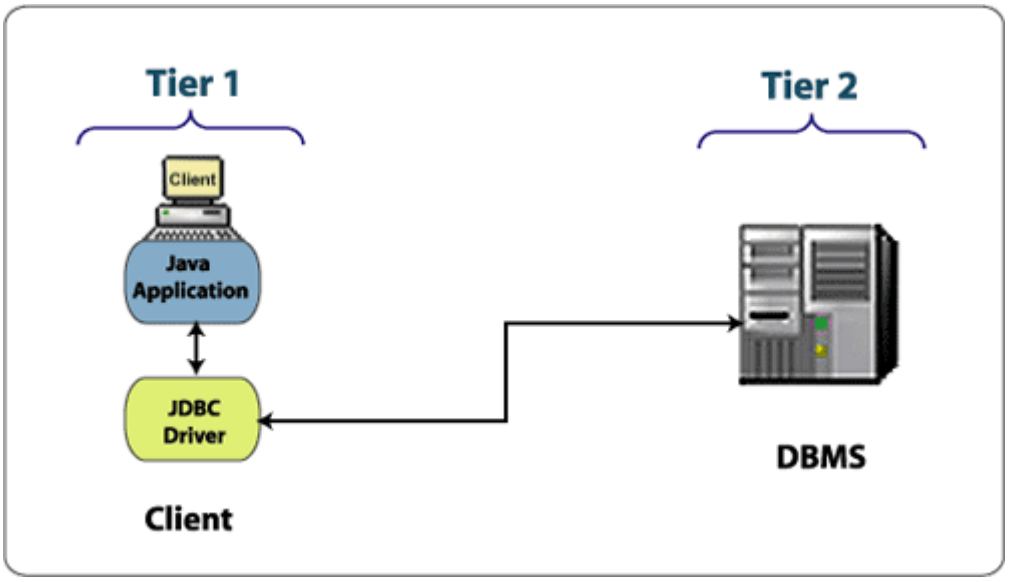

Fig 6(c): JDBC Two-Tier Model

The major disadvantage with this model is that there is a heavy burden on the client, as it takes care of both the presentation as well as the business logic. Also, the number of concurrent connections is limited because the connections are held by the application.

- **Three Tier Model**

  This model consists of a client, a middle tier, and a server. The client holds the application front-end, which concentrates on the presentation and communicates with the middle tier requesting data. The middle tier takes care of the business logic, communicating with the data source, handling transactions and connection pooling. The middle tier typically consists of an application, an application server and the JDBC driver. The application takes care of the business logic. The application server manages transactions, handling concurrent connections and identifying the drivers. The JDBC driver provides the connectivity to the data source and implemented the JDBC API, sending the request in the form that the data source can understand. The third layer is the actual data source, which is typically a DBMS (but can be any information system with JDBC driver support, such as a spreadsheet). The advantages in using a three-tier model have already previously been discussed, and won't be repeated here. Essentially, this architecture offers advantages in scalability, usability, maintenance and performance.

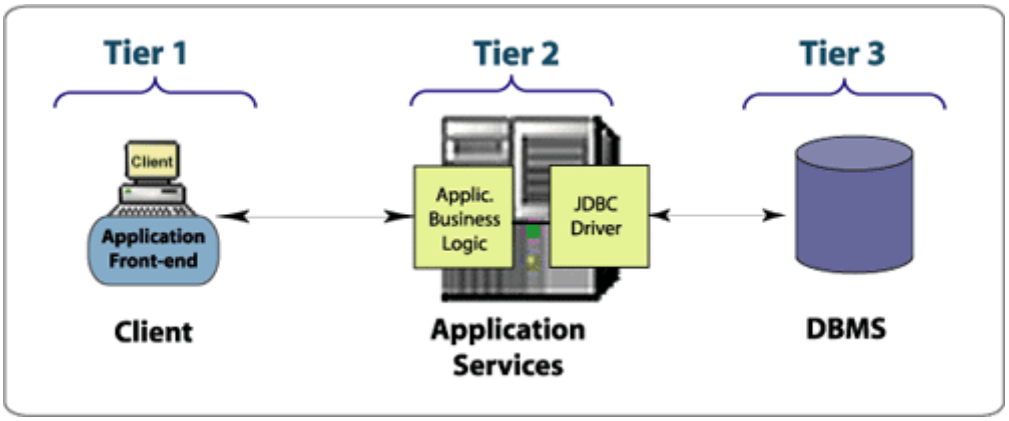

Fig 6(d): JDBC Three-Tier Model

7. Implementation

7.1 Security Features

As the system will run, the following security requirement will be fulfilling. Security Features of JavaFX is same as Java Platform.

- Safe from malevolent programs and should not be allowed to harm a user's computing environment.
- Non-intrusive as should be prevented from discovering private information on the host computer or the host computer's network.
- Authenticated as identity of parties involved in the program should be verified.
- Well-defined as security specification would be followed.
- Rules of operation should be set and verified.
- Well-behaved like programs should be prevented from consuming too many system resources.

In closure, provides a safe and secure platform for developing and running applications. Compile-time data type checking and automatic memory management leads to more robust code and reduces memory corruption and vulnerabilities. Bytecode verification ensures code conforms to the JVM specification and prevents hostile code from corrupting the runtime environment. Class loaders ensure that untrusted code cannot interfere with the running of other Java programs.

7.2 Coding Standards

The purpose of code is to facilitate the identification, retrieval of the items and information. A code is an oriented collection of symbols design to provide unique identification of an entry or attribute. Code is built with manually exclusive features. Codes in all cases specify object which are physical or on performance characteristics. They are used to give optimal distraction and other information. Codes are used for identifying, accessing, storing and matching records. The codes insure that only one value of the code with a single meaning

is correctly applied to give entity or attribute as described in various ways. Code can also be design in a manner easily understood and applied by the user.

This coding standard includes naming conversions of variables, constants and objects, standardized formats or labelling and commenting code, spacing, formatting and indenting.

7.2.1 Naming Convention

Classes' names and interface names are meaning full so anyone can easily understand this. The function names will start with small letters.

7.2.3 Labels and Comments

Sufficient labels and comments are included in the description of it for the benefits if the developer and other programmers who might examine or modified it later.

7.3 Functionalities

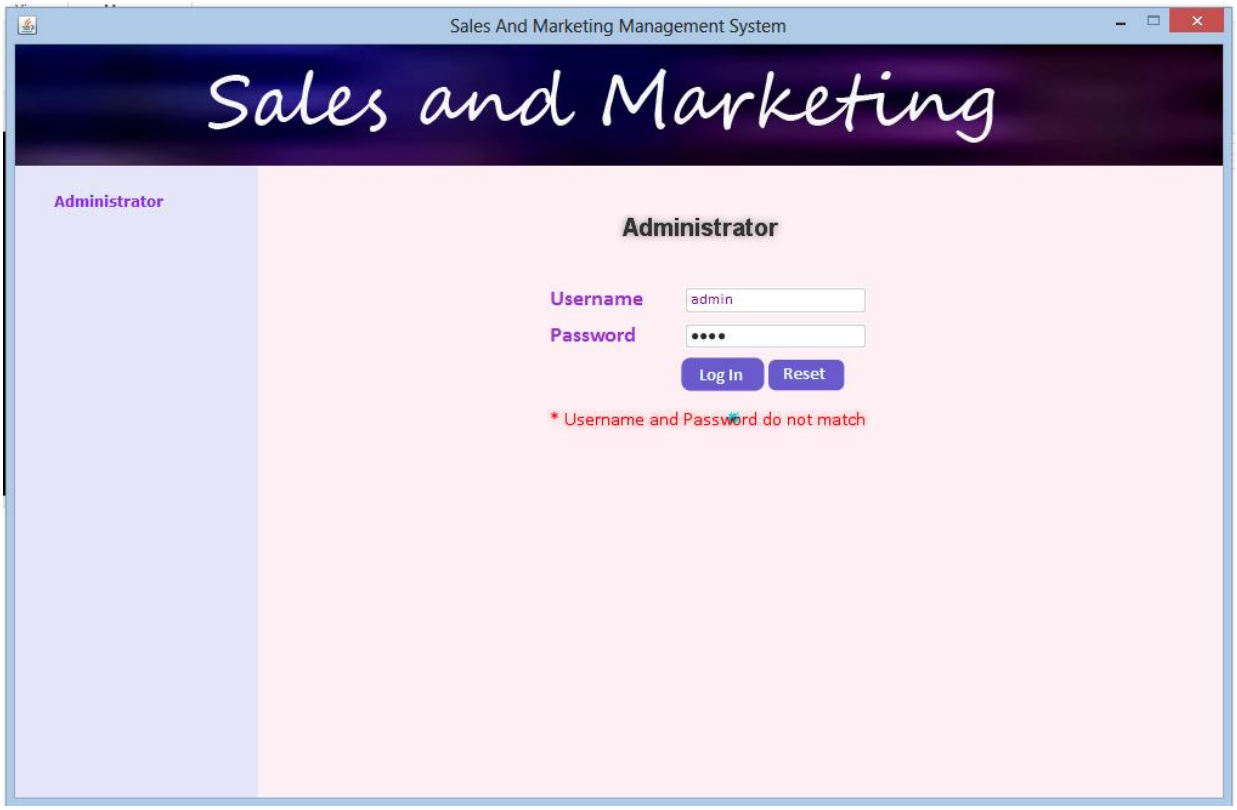

Fig 7(a): Admin Login

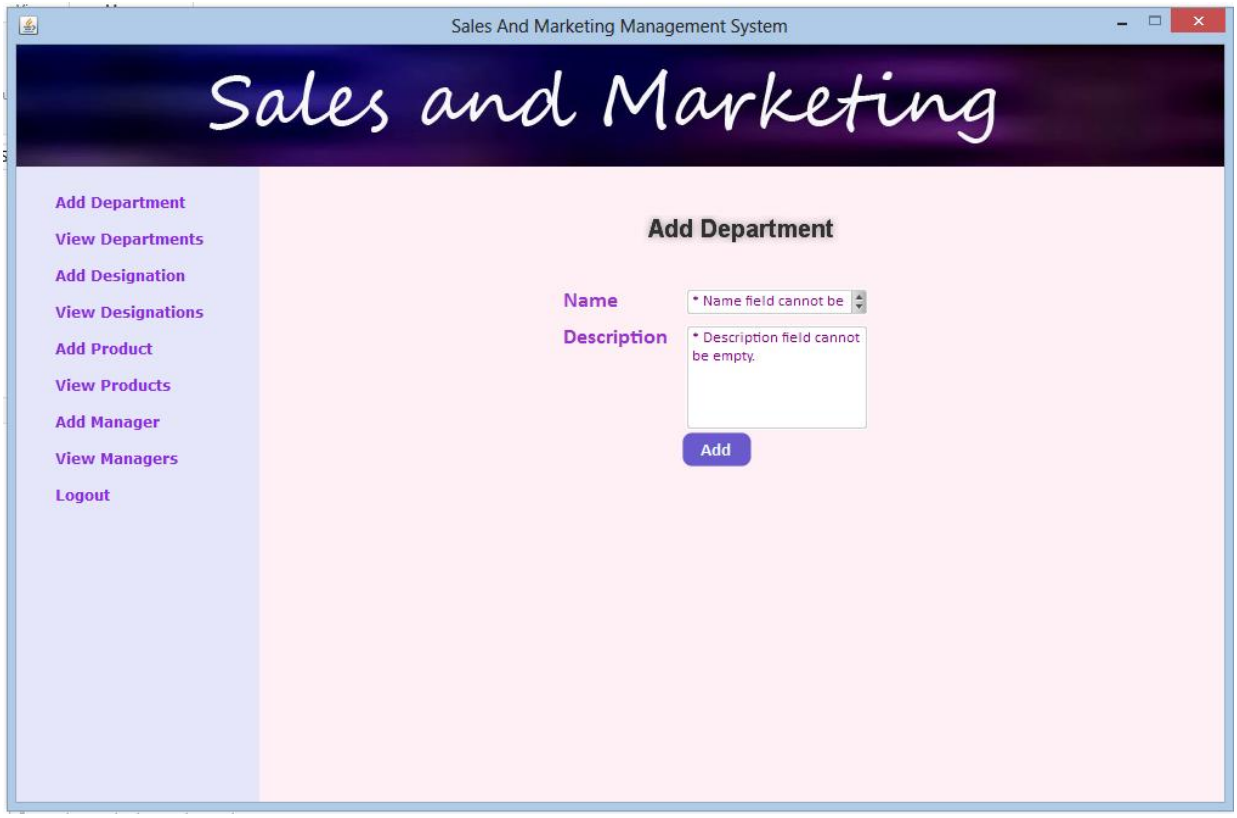

Fig 7(b): Add Department

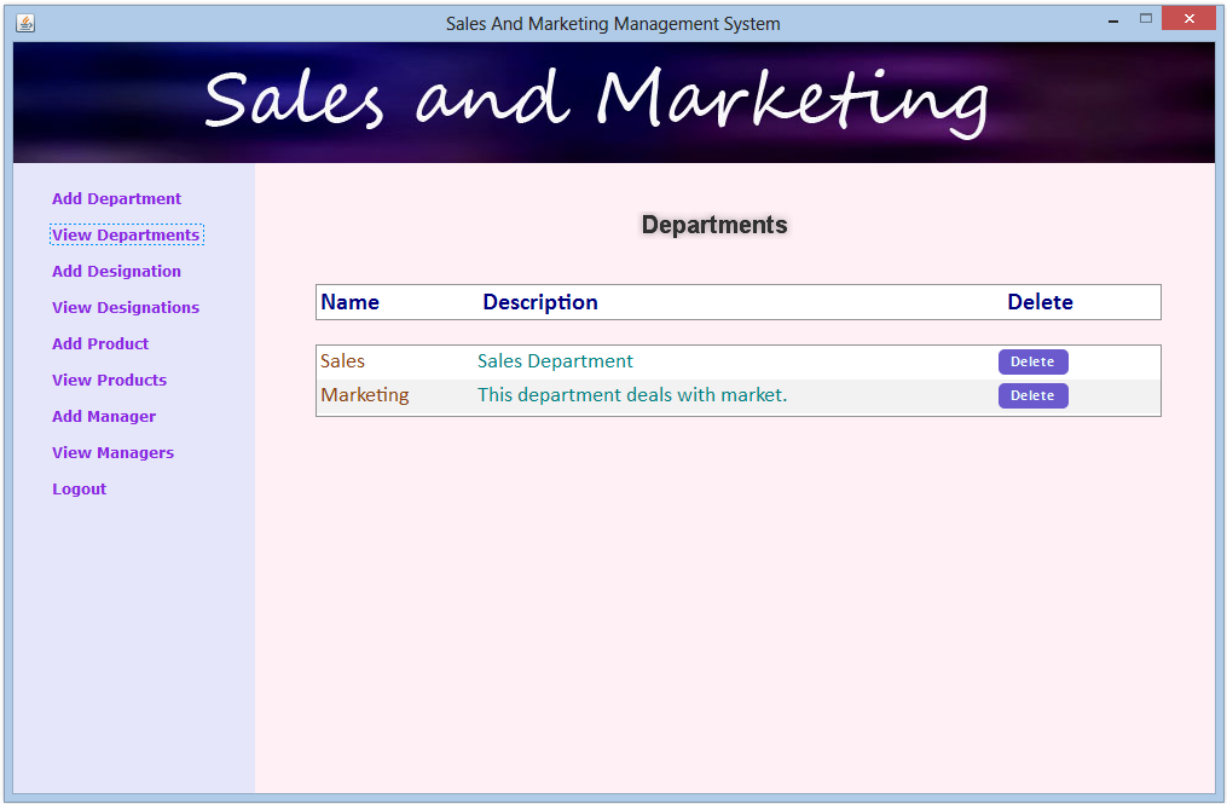

Fig 7(c): View Department

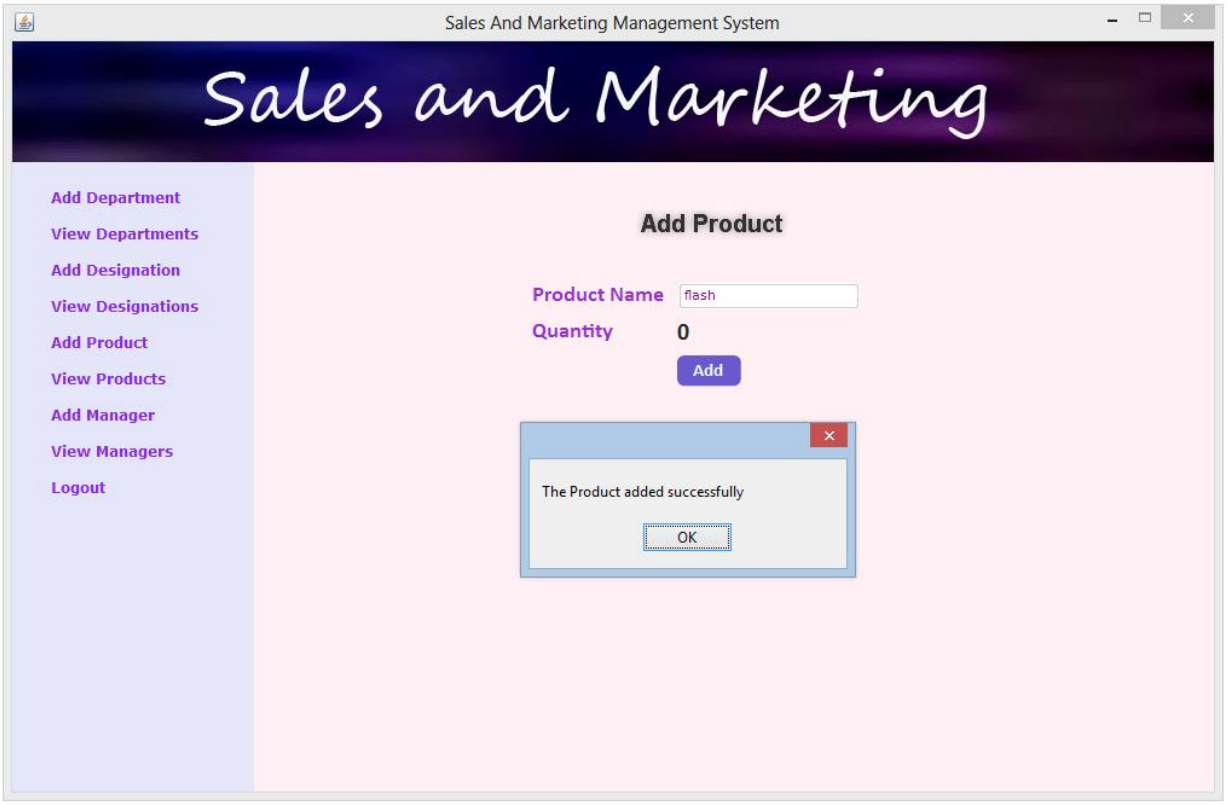

Fig 7(d): Add Product

Fig 7(e): View Product

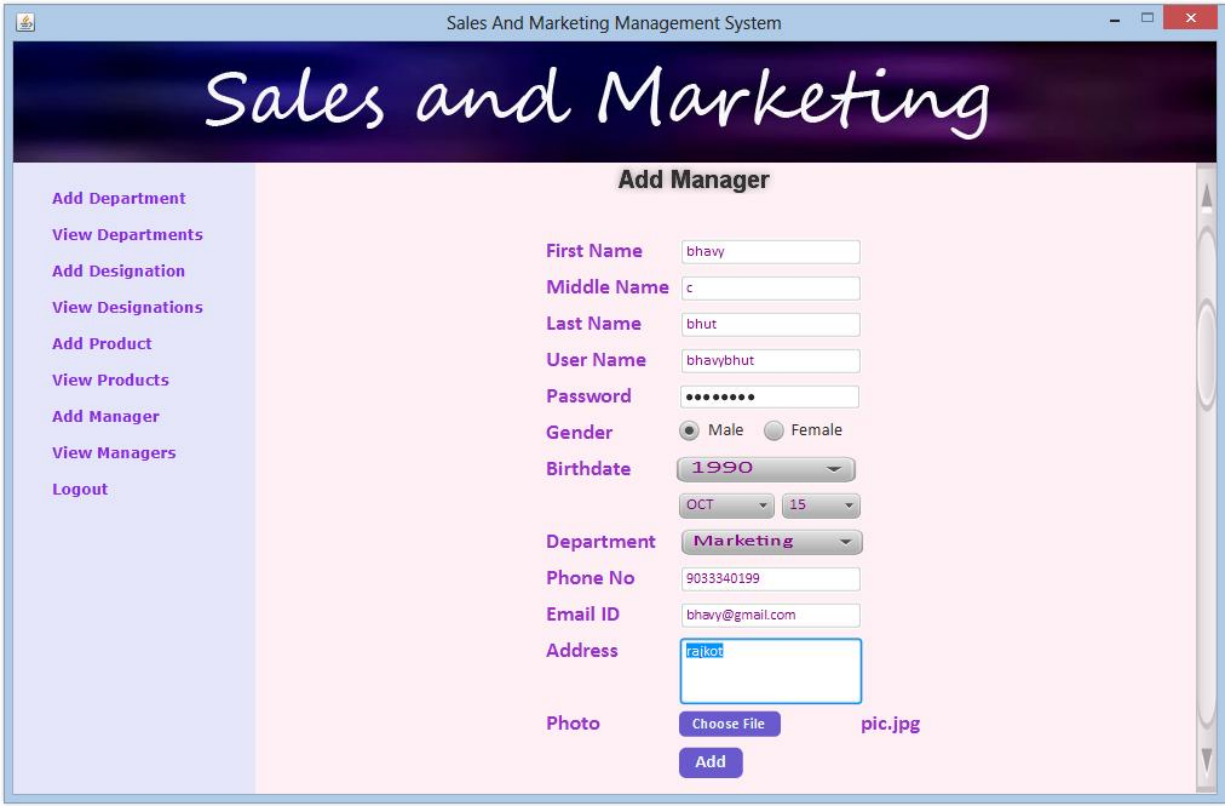

Fig 7(f): Add Manager

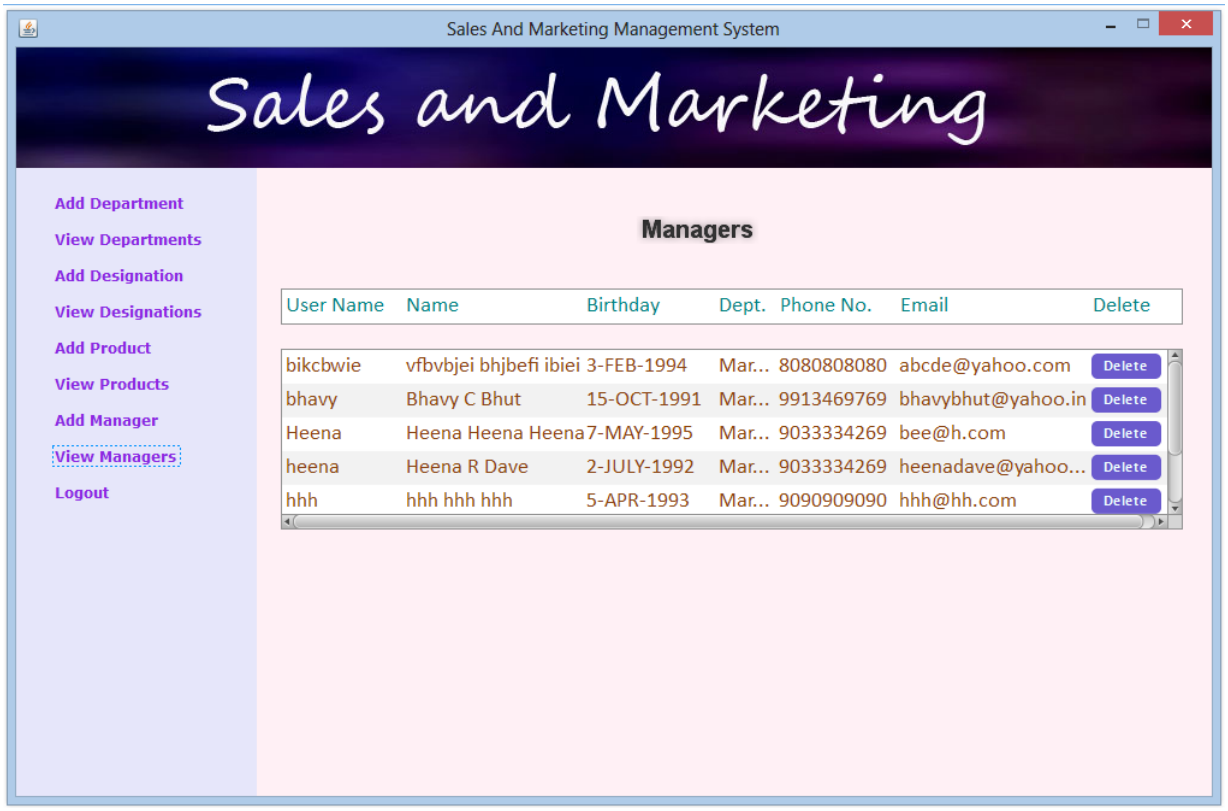

Fig 7(g): View Manager

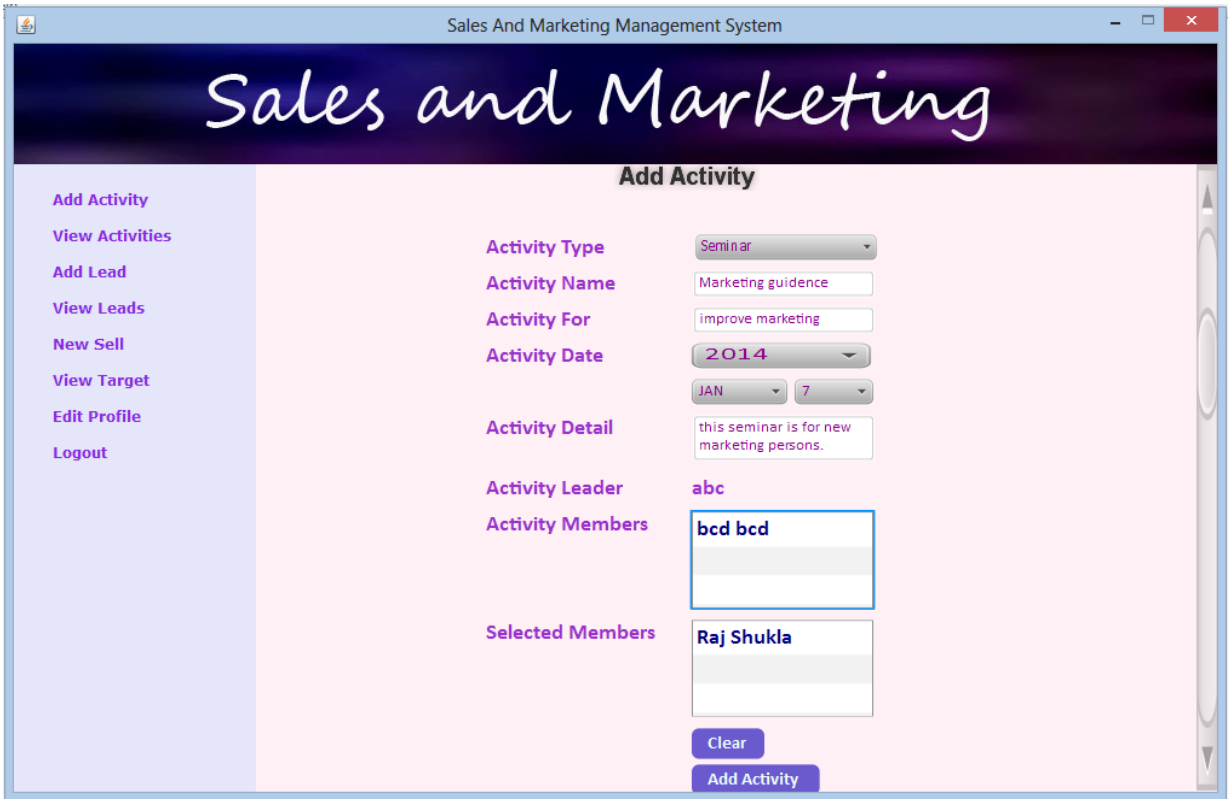

Fig 7(h): Add Activity

Fig 7(i): View Activities

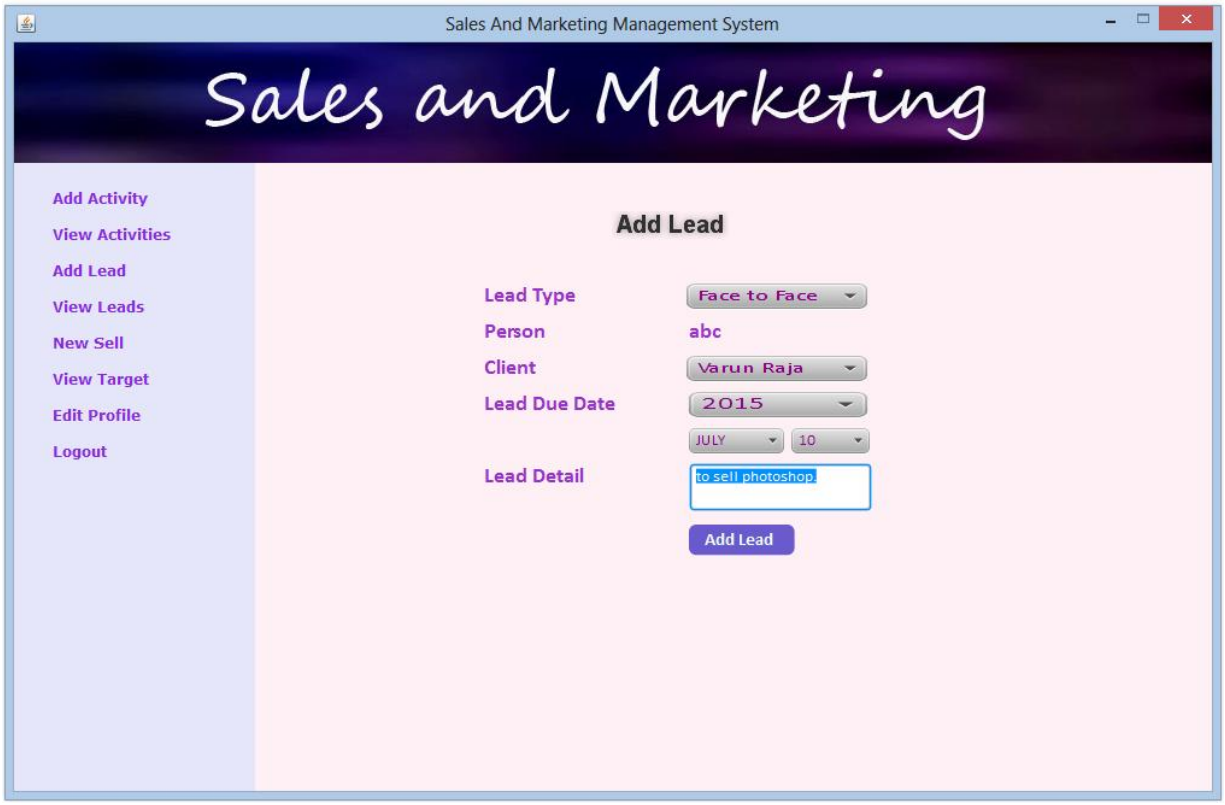

Fig 7(j): Add Lead

Fig 7(k): View Lead

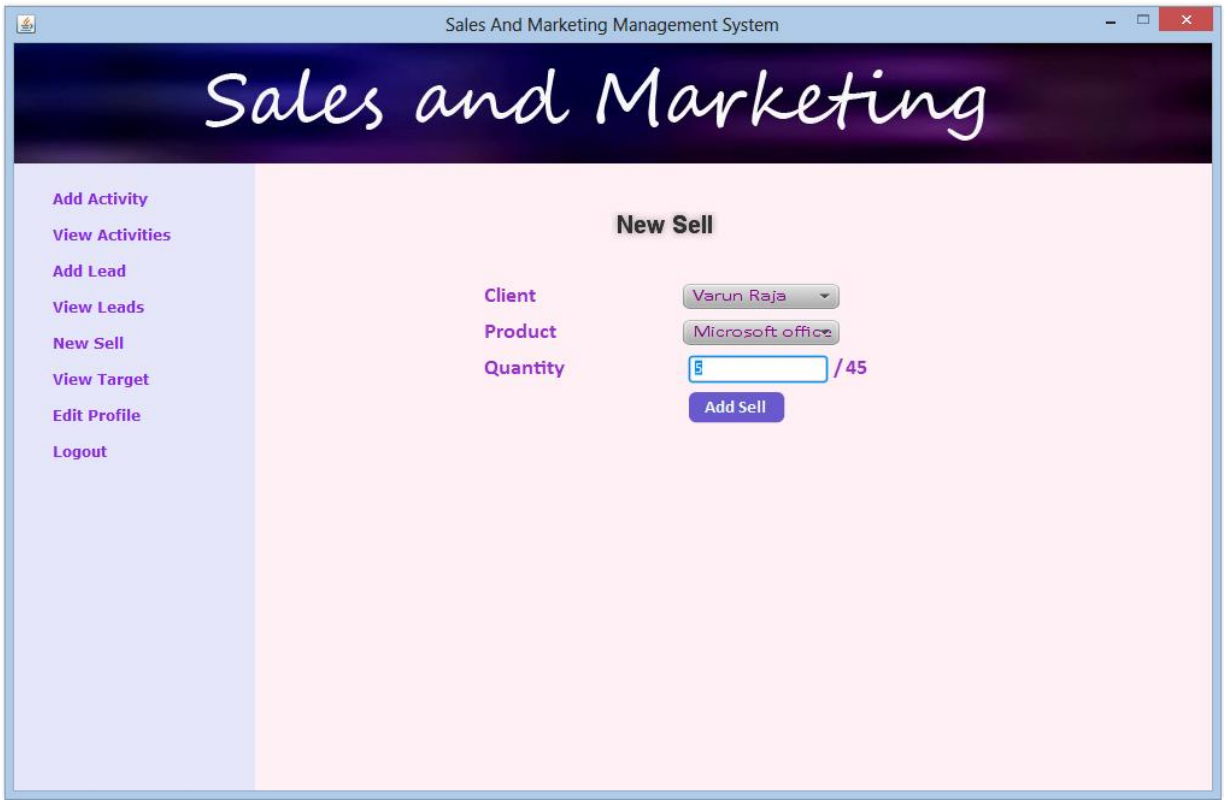

Fig 7(l): New Sell

Fig 7(m): View Target of Person

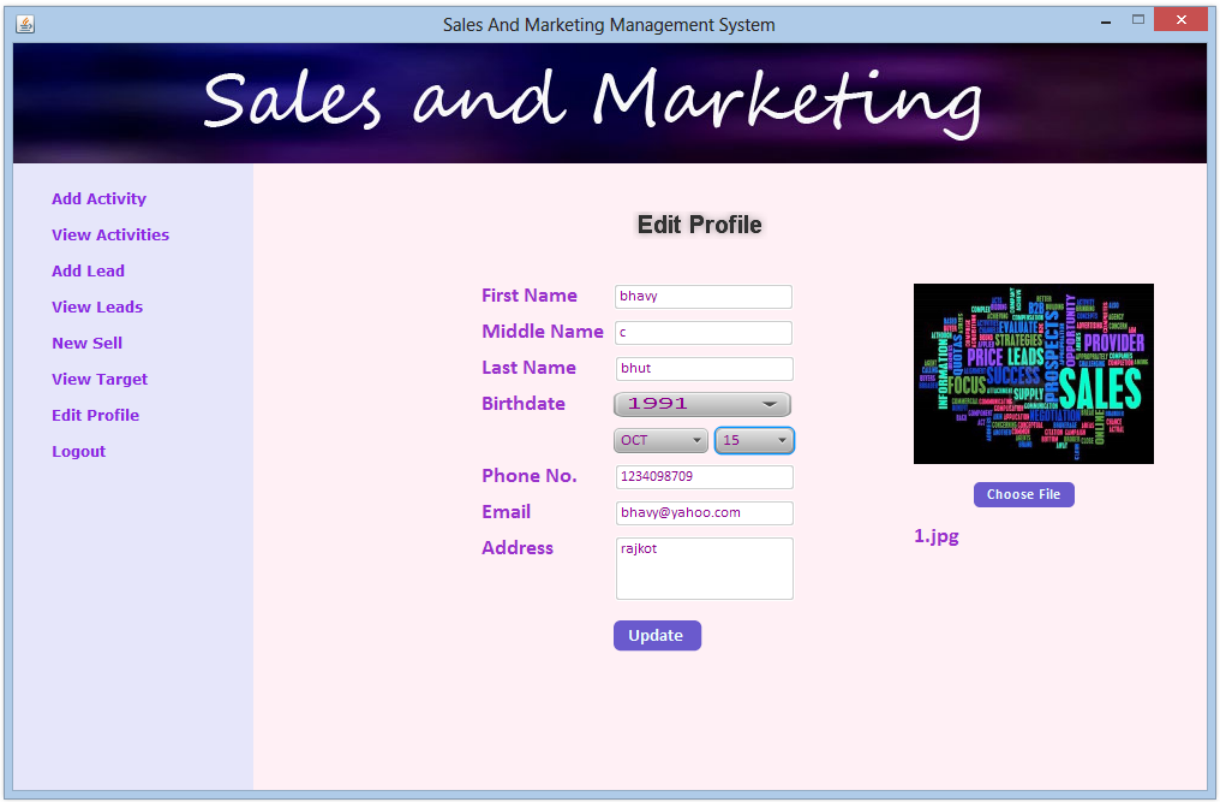

Fig 7(n): Edit Profile

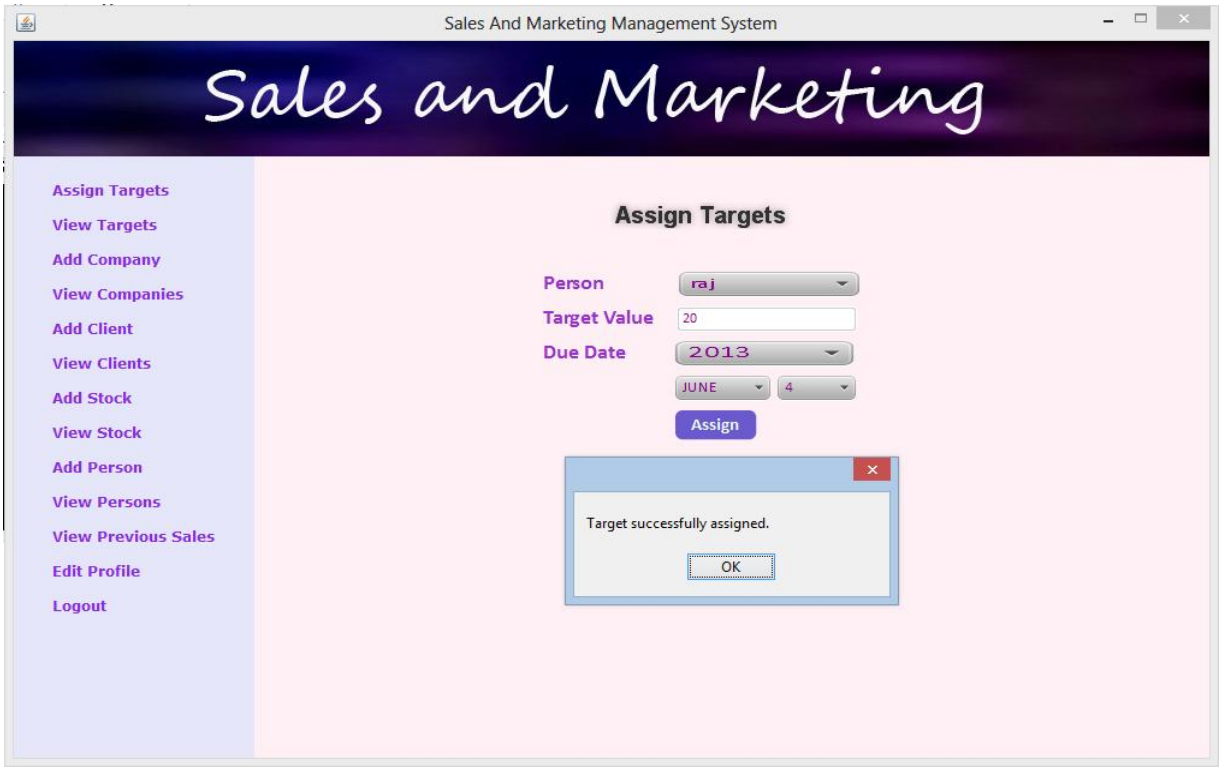

Fig 7(o): Assign Target

Fig 7(p): View Targets

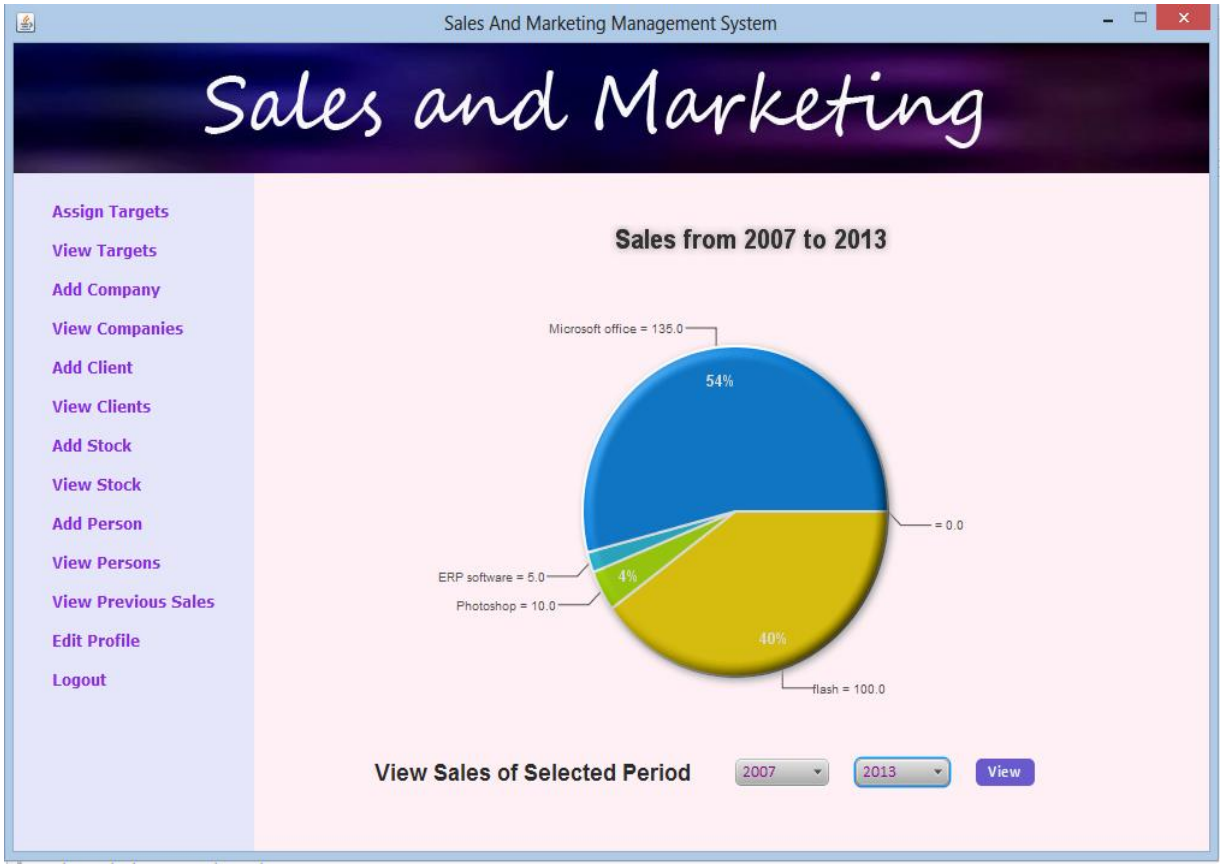

Fig 7(q): View Previous Sales

8. Testing

Testing is the process of executing the program to find if there are any errors. It is the final verification and validation activity. In testing phase we have tried to affirm the quality of the product. We have also tried to eliminate errors in the previous stages.

8.1 Testing Strategy

- **Unit Testing**

  Unit test comprises of a set tests performed by an individual program prior to the integration of the unit into large system. A program unit is usually the smallest free functioning part of the whole system. Module unit testing should be as exhaustive as possible to ensure that each representation handled by each module has been tested. All the units that makeup the system must be tested independently to ensure that they work as required. During unit testing some errors were raised and all of them were rectified and handled well. The result was quiet satisfactory and it worked well.

- **Integration Testing**

  Integration testing is a system technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. Bottom-up integration is the traditional strategy used to integrate the components of a software system into functioning whole. Bottom-up integration consists of unit test followed by testing of the entire system. A sub-system consists of several modules that communicated with other defined interface.

  The system was done the integration testing. All the modules were tested for their compatibility with other modules .They test was almost successful. All the modules coexisted very well, with almost no bugs. All the modules were encapsulated very well so as to not hamper the execution of other modules.

- **Validation Testing**

  After validation testing, software is completely assembled as a package, interfacing errors that have been uncovered and corrected and the final series of software test; the validation test begins. Steps taken during software design and testing can greatly improve the probability of successful integration in the larger system. System testing is actually a series of different tests whose primary purpose is to fully exercise the compute –based system.

- **Recovery Testing**

  It is a system that forces the software to fail in a variety of ways and verifies that the recovery is properly performed.

- **Security Testing**

  It attempts to verify that protection mechanisms built into a system will in fact protect it from improper penetration. The system's security must of course be tested from in vulnerability form frontal attack.

- **Black Box Testing**

  Black box testing is done to find out the following information as shown in below:

  - Incorrect or missing functions.
  - Interface errors.
  - Errors or database access.
  - Performance error.
  - Termination error.

  The mentioned testing is carried out successfully for this application according to the user's requirement specification.

- **Test Data Output**

  After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

8.2 Testing Methods

I used two types of testing methods:

1) Unit testing &

2) Integration Testing.

8.2.1.1 Unit Testing

- **Admin Module**

Table 5: Unit test at Admin side

No	Unit	Input	Expected Output	Obtained Output	Status	Remedies
1	Login	User Name and Password	Check with database for their relevance and redirect to the home page.	Database is verified and application redirected to admin home page	Success	
2	Add Department	Department Name	Insert into database	Data inserted into database	Success	
3	Delete Department	Select Department to delete	Delete from database	Data deleted from database	Success	Cannot edit Department
4	Add Designation	Designation Name	Insert into database	Data inserted into database	Success	
5	Delete Designation	Select Designation to delete	Delete from database	Data deleted from database	Success	Cannot edit Designation
6	Add Product	Product Name	Insert product into database	Data inserted into database	Success	
7	Delete Product	Select Product to delete	Delete from database	Data deleted from database	Success	Cannot edit Product
8	Add Manager	Give all the necessary details	Check with database for duplication of record and proper validation	Data validated and Detail inserted	Success	
9	View Manager	Not necessary	All managers will show	Managers detail showed	Success	
10	Delete Manager	Select Manager to delete	Delete from database	Data deleted from database	Success	

11	Logout	Not necessary	Will be Redirect to main page and clear user credentials	Redirected to main page and cleared user credentials	Success	

- **Manager Module**

Table 6: Unit test at Manager side

No	Unit	Input	Expected Output	Obtained Output	Status	Remedies
1	Login	User Name and Password	Check with database for their relevance and redirect to the home page.	Database is verified and application redirected to admin home page	Success	
2	Assign Target	Give all the necessary details	Insert into database	Data inserted into database	Success	
3	View Target	Not necessary	All assigned target will show	Target detail showed	Success	Cannot edit,update, delete Target
4	Add Company	Give all the necessary details	Insert into database	Data inserted into database	Success	
5	Edit Company	Select Company to edit	Update database	Data updated into database	Success	
6	Add Client	Give all the necessary details	Insert into database	Data inserted into database	Success	
7	Edit Client	Select Company to edit	Update database	Data updated into database	Success	
8	Add Stock	Add the quantity of product into stock	Insert into database	Data inserted into database	Success	
9	View Stock	Not necessary	All product stock will be shown	Stock detail showed	Success	Cannot edit,update, delete Stock

10	Add Person	Give all the necessary details	Check with database for duplication of record and proper validation	Data validated and Detail inserted	Success	
11	Delete Person	Select Manager to delete	Delete from database	Data deleted from database	Success	
12	View Previous Sales	Select time interval to view previous sales	Will Fetch details of sales from database and will display in graphical format	Fetched details of sales from database and displayed in graphical format	Success	
13	Edit Profile	Give all the update details	Update Database	Data updated into database	Success	Cannot update assigned username
14	Logout	Not necessary	Will be Redirect to main page and clear user credentials	Redirected to main page and cleared user credentials	Success	

- **Person Module**

Table 7: Unit test at Person Module

No	Unit	Input	Expected Output	Obtained Output	Status	Remedies
1	Login	User Name and Password	Check with database for their relevance and redirect to the home page.	Database is verified and application redirected to admin home page	Success	
2	Add Activity	Activity Name and other related info.	Insert into database	Data inserted into database	Success	
3	Delete Activity	Select Activity to delete	Delete from database	Data deleted from database	Success	Cannot edit Activity
4	Add Lead	Lead and other related info.	Insert into database	Data inserted into database	Success	

5	Delete Lead	Select Lead to delete	Delete from database	Data deleted from database	Success	Cannot edit Lead
6	New Sell	Selling info.	Insert product sell into database	Data inserted into database	Success	
7	View Target	Not necessary	Fetch data from database	Data displayed from database	Success	Cannot edit, delete assigned target
8	Edit Profile	Give all the update details	Update Database	Data updated into database	Success	
12	Logout	Not necessary	Will be Redirect to main page and clear user credentials	Redirected to main page and cleared user credentials	Success	

8.2.1.2 Integration Testing

Table 8: Integration test of all three modules

No	Unit	Input	Expected Output	Obtained Output	Status	Remedies
1	Admin Module	Register Manager	Insert Manager	Manager inserted	Success	
2	Manager Module	Assign Target	Assign Target	Target Assigned	Success	
3	Person Module	Add new sell	Insert sell	Sell inserted to database	Success	

8.2.2 Test Summary

Developer will test all the above tests after every 7-10 days at starting of this system. Here after 1 year, developer will test the entire above test once in a month because after 1 year system will be stable and easily handled and more error free.

9. Limitation and Future Enhancement

- **Limitation**
  - The size of the database increases day-by-day, increasing the load on the database.
  - Training for simple computer operations is necessary for the users working on the system.
  - At present, this system is not runnable into browsers.
  - System doesn't have executable file support.

- **Future Enhancement**

Any system starts with providing necessary functionalities first. Then, it is possible to add new modules and functionalities according to the requirement. Our system supports major functionalities used by the Sales and Marketing people. But to make the system more efficient than traditional channel, some future enhancements of our system are:

  - We look forward in working with the impact of lead into the Sales. This will help system to track more accurately about the Marketing impact on Sales.
  - Assign targets to Persons by product. Now, in this system, it is possible to assign overall target but product wise target assignment is not possible.
  - The method of messaging between Manager and Person can be added.
  - Persons can view their past sales and can compare with the current one.
  - By the future technology, it will be possible to make this system to be run into browsers as well without changing much of the code.
  - Back up will automatically generated every 24 hours.

10. Conclusion

The project titled as “Sales and Marketing Management System” is a desktop based application. This system provides facility for assign targets, add lead, add product, add manager or person, add activity, view previous sales etc. This system is developed with scalability in mind. Additional modules can be easily added when necessary. The system is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The project has been completed successfully with the maximum satisfaction of the organization. The constraints are met and overcome successfully. The system is designed as like it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. This system has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the system. The application has been tested with live data and has provided a successful result. Hence the system has proved to work efficiently.

Appendix 1: Data Dictionary

- **admin_master**

Table 9: Admin Table

Column Name	Column Type	Attribute	Extra
Username	Varchar2 (20)	Primary Key	-
Password	Varchar2 (100)	-	-

- **dept_master**

Table 10: Department table

Column Name	Column Type	Attribute	Extra
Id	Number	Primary Key	Auto Increment
Name	Varchar2 (20)	-	-
Description	Varchar2 (100)	-	-

- **desig_master**

Table 11: Designation table

Column Name	Column Type	Attribute	Extra
Id	Number	Primary Key	Auto Increment
Name	Varchar2 (20)	-	-
Description	Varchar2 (100)	-	-

- **prev_sales_master**

Table 12: Previous Sales Record table

Column Name	Column Type	Attribute	Extra
Id	Number	Primary Key	Auto Increment
S_id	Number	-	Person Id
Target	Varchar2 (15)	-	-
Target_due_date	Varchar2 (2)	-	-
Target _due_month	Varchar2 (4)	-	-
Target _due_year	Varchar2 (4)	-	-
Status	Varchar2 (15)	-	Target Status

- **salesmanager_master**

Table 13: Manager table

Column Name	Column Type	Attribute	Extra
Id	Number	Primary Key	Auto Increment
First_name	Varchar2 (50)	-	-
Middle_name	Varchar2 (50)	-	-
Last_name	Varchar2 (50)	-	-
Username	Varchar2 (50)	-	-
Password	Varchar2 (100)	-	-
Gender	Varchar2 (6)	-	-
Birthdate_day	Varchar2 (2)	-	-
Birthdate_month	Varchar2 (4)	-	-
Birthdate_year	Varchar2 (4)	-	-
Department	Varchar2 (30)	-	-
Phone_no	Number	-	-
Email_id	Varchar2 (40)	-	-
Address	Varchar2 (100)	-	-
Photo	Varchar2 (40)	-	-

- **stock_master**

Table 14: Stock table

Column Name	Column Type	Attribute	Extra
St_id	Number	Primary Key	Auto Increment
Sman	Varchar2 (20)	-	Person Id
Cl	Varchar2 (20)	-	Client Name
Prodname	Varchar2 (20)	-	Product Name
Qty	Number	-	Product Quantity
Day	Varchar2 (2)	-	-
Month	Varchar2 (4)	-	-
Year	Varchar2 (4)	-	-

- **lead_master**

Table 15: Lead table

Column Name	Column Type	Attribute	Extra
Lead_id	Number	Primary Key	Auto Increment
Lead_type	Varchar2 (50)	-	-
Lead_salesman	Varchar2 (20)	-	-
Lead_client	Varchar2 (20)	-	-
Lead_due_date	Varchar2 (2)	-	-
Lead_due_month	Varchar2 (4)	-	-
Lead_due_year	Varchar2 (4)	-	-
Lead_detail	Varchar2 (200)	-	-

- **salesman_master**

Table 16: Person table

Column Name	Column Type	Attribute	Extra
Id	Number	Primary Key	Auto Increment
First_name	Varchar2 (50)	-	-
Middle_name	Varchar2 (50)	-	-
Last_name	Varchar2 (50)	-	-
Username	Varchar2 (50)	-	-
Password	Varchar2 (100)	-	-
Gender	Varchar2 (6)	-	-
Birthdate_day	Varchar2 (2)	-	-
Birthdate_month	Varchar2 (4)	-	-
Birthdate_year	Varchar2 (4)	-	-
Department	Varchar2 (30)	-	-
Phone_no	Number	-	-
Email_id	Varchar2 (40)	-	-
Address	Varchar2 (100)	-	-
Under	Varchar2 (30)	-	-
Photo	Varchar2 (40)	-	-
Target	Varchar2 (15)	-	-
Target_status	Varchar2 (15)	-	-
Due_date_day	Varchar2 (2)	-	-
Due_date_month	Varchar2 (4)	-	-
Due_date_year	Varchar2 (4)	-	-

- **company_master**

Table 17: Company table

Column Name	Column Type	Attribute	Extra
C_id	Number	Primary Key	Auto Increment
C_name	Varchar2 (200)	-	-
C_phone_no	Varchar2 (10)	-	-
C_email_id	Varchar2 (50)	-	-
C_address	Varchar2 (50)	-	-
C_city	Varchar2 (50)	-	-
C_state	Varchar2 (50)	-	-
C_country	Varchar2 (50)	-	-

- **client_master**

Table 18: Client table

Column Name	Column Type	Attribute	Extra
Cl_id	Number	Primary Key	Auto Increment
Cl_first_name	Varchar2 (50)	-	-
Cl_middle_name	Varchar2 (50)	-	-
Cl_last_name	Varchar2 (50)	-	-
Cl_company_id	Number	-	-
Cl_company_name	Varchar2 (200)	-	-
Cl_phone_no	Number	-	-
Cl_email_id	Varchar2 (50)	-	-

- **product_master**

Table 19: Product table

Column Name	Column Type	Attribute	Extra
Id	Number	Primary Key	Auto Increment
Name	Varchar2 (20)	-	-
Quantity	Number	-	-

- **activity_master**

Table 20: Activity table

Column Name	Column Type	Attribute	Extra
Act_id	Number	Primary Key	Auto Increment
Act_type	Varchar2 (50)	-	-
Act_name	Varchar2 (50)	-	-
Act_for	Varchar2 (50)	-	-
Act_due_date	Varchar2 (2)	-	-
Act_due_month	Varchar2 (4)	-	-
Act_due_year	Varchar2 (4)	-	-
Act_detail	Varchar2 (200)	-	-
Act_leader	Varchar2 (20)	-	-
Act_member	Varchar2 (200)	-	-

REFERENCES

- **BOOKS:**
  - Gail and Paul Anderson - 'Essential JavaFX' - Prentice Hall
  - James L. Weaver - 'JavaFX Script' - Apress
  - J. F. DiMarzio - 'JavaFX A Beginner's Guide' - Oracle Press
  - Kim Topley – 'JavaFX Developer's Guide' - Addision Wesley
  - Lucas L. Jordan - 'JavaFX Special Effects' - Apress

- **ONLINE REFERENCE:**

  - http://docs.oracle.com/cd/B12037_01/server.101/b10739/create.htm
  - http://docs.oracle.com/javafx/index.html
  - http://stackoverflow.com/questions/15482623/javafx-1-3-listview
  - http://stackoverflow.com/questions/8392119/javafx-1-3-crud-using-listview
  - http://docs.oracle.com/cd/E17802_01/javafx/javafx/1.3/docs/api