

1. INTRODUCTION

E-Insurance is an application which is used tracking the detail of insurance activity of the customer details, insurance policy details. This project is useful for any kind of insurance company to manage all their insurance details. This project also helps in to find whether the customer is eligible of the insurance or not. With the help of this project all the activity is done by the E-Insurance application.

In this application the user login into application after the registration, it will show policy which customer are using at a time and show the amount he/she must pay for their next instalment and show the previous amount he/she paid. After the login he/she can also buy the new policy according to their needs present in the policy list.

In this online process the user enters the website it will show details about insurance and its types, also it will show the details about different duration schemes to the corresponding insurance type or insurance policy. In this process contains the user registration form which is used to apply for insurance policy through online. It also helps the customer to view their own insurance status information.

If the user registered insurance policy to this website, it will add into his profile and give the benefits to the user's. If the policy holder wants to view the information about their own policy details, he/she login into his account in the profile his current policy has been shown.

The existing system is the manual system. The manual system is prone to error. It is time consuming. It is very difficult for a person to produce the report. There are chances for changing the scheme report and do malpractice. This system involves a lot of manual entries with the applications to perform the desired task. Usage of papers in the payment process leads to less efficiency, less accuracy, and less productivity. It also increasing expenditure for papers shuffling and amount of storage. It may be delay in time between the payment and its receipt. In the existing system persons from different part of the world cannot transact data efficiently. Till now most companies has adopted the above manual system that produces lot of problems and at the same time the following disadvantages are there with the above system.

2. PROBLEM DEFINITION

PROJECT OBJECTIVE:

The following are the broad benefits of holding Insurance Policies in electronic form:

- The insurance policies are stored in electronic form. The risk of losing the policy bonds is avoided.
- The details of the insurance policies can be accessed from anywhere and anytime.
- As of now, Life insurance policies can be stored in electronic form. From December 2014, the insurance repositories will extend their services to store the health, motor, pension, and all type of insurance policies.
- Service requests like 'change of address' can be submitted on your e-insurance account. The IR will update the same on all the policies which are linked to the e-insurance account. There is no need to visit all the insurers.
- One time KYC (Know Your Customer). No KYC repetition when you buy new policies.
- You can get the statement of account about the details of the policies from IR.
- There is no cost involved for the policyholders to open an e-Insurance Account. It is free of cost. Insurance Repositories are paid by the insurers for their services.
- Premiums for all the policies can be paid online.
- Policy benefits like money back instalments /survival benefits/claims can be paid electronically through the registered bank a/c of the policyholder.
- An authorized representative can be appointed by the e-Insurance account holder to operate his/her account in case of unfortunate demise. Nominee and authorized representative can be the same person.
- the e-Insurance Account holder will have an option to shift from one Insurance Repository to the other.
- You can opt out of this service anytime.
- When you are buying a new insurance policy, quote your e-Insurance account number in the proposal form. Request the insurance company to issue the policy in electronic form.

PROPOSED METHODOLOGY:

There are various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as “Software Development Process Models” (e.g., Waterfall model, incremental model, V-model, iterative model, RAD model, Agile model, Spiral model, Prototype model etc.). Each process model follows a particular life cycle to ensure success in process of software development.

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design which is called development phase. After coding and development, the testing verifies the deliverable of the implementation phase against requirements.

PHASES OF SDLC

- Recognition of need/Initial Investigation
- Feasibility Study
- Analysis
- Design
- Coding
- Testing
- Maintenance

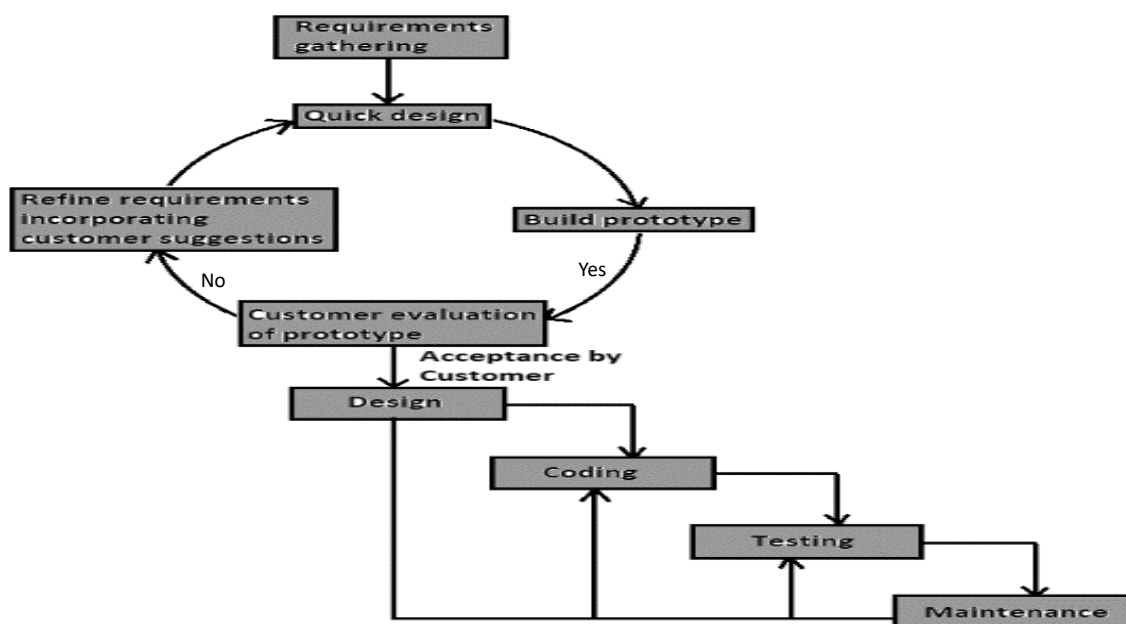


Fig 2.1

3. TECHNOLOGY USED

PHP:

PHP stands for "PHP: Hypertext Pre-processor". It is a widely used open-source general-purpose scripting language that is especially suited for web development and can be embedded into HTML². PHP is a server-side scripting language designed specifically for web development⁴. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites³. PHP can create, open, read, write, delete, and close files on the server. PHP can collect form data. PHP can send and receive cookies. PHP can add, delete, modify data in your database. PHP can be used to control user-access. PHP can encrypt data. With PHP you are not limited to output HTML. You can output images or PDF files.

HTML:

HTML stands for Hypertext Markup Language. It is a standardized system for tagging text files to achieve font, colour, graphic, and hyperlink effects on World Wide Web pages¹. HTML is the most basic building block of the Web. It defines the meaning and structure of web content². HTML is a markup language that defines the structure of your content⁴. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way or act a certain way.

CSS:

CSS stands for Cascading Style Sheets. CSS describes how HTML elements are to be displayed on screen, paper, or in other media¹. CSS is a rule-based language — you define the rules by specifying groups of styles that should be applied to elements or groups of elements on your web page². CSS saves a lot of work. It can control the layout of multiple web pages all at once¹. External stylesheets are stored in CSS files.

JAVASCRIPT:

JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else². JavaScript is a programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved². JavaScript can update and change both HTML and CSS³. JavaScript can calculate, manipulate, and validate data.

MySQL:

MySQL is an open-source relational database management system¹ that stores data in tables made up of rows and columns. Users can define, manipulate, control, and query data using Structured Query Language (SQL)¹. MySQL is often used for web applications and acts as the database component of the LAMP web application software stack.

BOOTSTRAP:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development². It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components². Bootstrap is used for creating front-end setups for responsive websites³. It is a collection of reusable HTML, CSS and JavaScript code that facilitates a faster workflow³.

4. SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Requirement

It is recommended that the minimum configuration for Software is as appended below:

- Processor : Pentium Series (1GHz or above)
- RAM : 1 GB
- Hard Disk : With 80GB or above
- Monitor : SVGA Monitor
- CD Drive : 52X or above
- Keyboard : Normal/Multimedia
- Mouse : Normal

Software Requirement

Use of Software

Operating System
Front End
Back End
For Documentation

Name of Software

Window seven, XP-SP3
PHP, HTML
MySQL 5.5
MS-Office 2007

Front End: -

- HTML
- CSS with Bootstrap
- JavaScript

Back End: -

- PHP

Database: -

- MySQL

5. MODULE DESCRIPTION

The modules involved are:

1. Administrator
2. Users
3. Reports

Administrator: -

The administrator is the super user of this application. Admin only access to enter this admin page. Admin had all privileges to give authenticate to user and gives privileges that which portion they can access. The administrator has all the information about all the users and about all products.

This module is divided into different sub-modules.

1. Manage Category
2. Manage Products
3. Manage Users
4. Manage Orders History
5. Reports

1. Manage Category:

a. Add Category:

The Online E – Insurance project contains different kind of products. The products can be classified into different categories. Admin can add new category into the existing system.

b. Edit Category:

Based on the market analysis admin can update a category. He can change the category of a particular product.

c. Delete Category:

Admin has privilege to delete a category from the existing system.

d. View Category:

All existing categories can view by the administrator. The administrator can keep track on categories.

2. Manage Products:

a. Add Products:

The E - Insurance project contains different kind of products. The products can be classified into different categories. Admin can add new products into the existing system, and he can map with the category.

b. Edit Products:

Based on the market position admin will modify the product details like price, quantity, packing etc.

c. Delete Products:

Administrator can delete the products based on the category.

d. View Products:

Administrator can view the product details. He can watch the products prices and contents of the products.

3. Manage Users:

a. View Users:

Admin can view the user details which are registered in his system because he must know how many users entered his site and purchase products. So, he can keep track on user's information.

b. Edit Users:

Admin has privileges to edit user information, which is requested by the users.

c. Delete Users:

Administrator has a right to delete user information without intimation.

4. Manage Order History:

a. View the Order History:

Administrator can view the Order History which is generated by the user. He can verify the products and their prices.

b. Generate Bill for History:

Administrator generates the bill for order list which is prepared by the users.

c. **Set Status:**

Admin can set the status for the Order list which is given by the user. The status can be different types as follows.

- Shifted
- Delivered
- Cancelled
- Rejected

USERS: -

1. View Products:

User can view the list of products based on their categories. A detailed description of a particular product can be displayed to the user, products details, by, price and date of manufacture etc...

2. Select Order Products:

After verifying the products details users select some required products.

3. Generate Order Products:

Whenever user completes his selection of product, he can add these selected products into E - Insurance, for this he must be an authenticated person. If he is not an authenticated person his selected products should not add into E - Insurance. Then he needs to register himself. If he is registered person, he can directly generate the Order Products.

4. Registration:

A registered user only can purchase Insurance from online. So that if a user wants to purchase products, he must be fill his registration form and must obtain a valid username and password which is provided by the administrator.

5 Billing:

After submitting his Policy user must pay his bill which is generated by the administrator. The mode of billing is Cash on delivery.

Reports: -

Reports can be generated by the admin for various purposes. Based on the reports admin take decisions of the system. Different kinds of reports can be generated according to the requirement.

6. DATABASE SCHEMA

CUSTOMER SIGNUP:

FIELD NAME	DATA TYPE	CONSTRAINT
FIRST NAME	VARCHAR2(10)	Not null
LAST NAME	VARCHAR2(10)	Not null
GENDER	Char (1)	Not null
DOB	DATE	Not null
ADDRESS	VARCHAR2(30)	Not null
PHONE NUMBER	NUMBER (10)	Not null
PIN CODE	NUMBER (6)	Not null
USERID	NUMBER (10)	Primary key
PASSWORD	VARCHAR2(20)	Not null
CITY	VARCHAR2(15)	Not null
STATE	VARCHAR2(15)	Not null
EMAIL	VARCHAR2(15)	null

LIFE INSURANCE:

FIELD NAME	DATA TYPE	CONSTRAINT
POLICY NO.	NUMBER (20)	Primary Key
POLICY NAME	VARCHAR2(20)	Not null
PRICE	NUMBER	Not null

HEALTH INSURANCE:

FIELD NAME	DATA TYPE	CONSTRAINT
POLICY NO.	NUMBER (20)	Primary key
POLICY NAME	VARCHAR2(20)	Not null
PRICE	NUMBER (10)	Not null

VEHICLE INSURANCE:

FIELD NAME	DATA TYPE	CONSTRAINT
POLICY NO.	NUMBER (20)	Primary Key
POLICY NAME	VARCHAR2(20)	Not Null
PRICE	NUMBER (10)	Not null

HOME INSURANCE:

FIELD NAME	DATA TYPE	CONSTRAINT
POLICY NO.	NUMBER (20)	Primary key
POLICY NAME	VARCHAR2(20)	Not Null
PRICE	NUMBER (10)	Not null

PAYMENT:

FIELD NAME	DATA TYPE	CONSTRAINT
USER ID	NUMBER (10)	Primary key
TOTAL PAYMENT	NUMBER (15)	Not null
PAYMENT DEW	NUMBER (15)	Not Null
POLICY NO	NUMBER (10)	Not null

ADMIN SIGNUP:

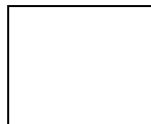
FIELD NAME	DATA TYPE	CONSTRAINT
FIRST NAME	VARCHAR2(10)	Not null
LAST NAME	VARCHAR2(10)	Not null
GENDER	CHAR (1)	Not null
DOB	DATE	Not null
ADDRESS	VARCHAR2(30)	Not null
PHONE NUMBER	NUMBER (10)	Not null
PIN CODE	NUMBER (6)	Not null
ADMIN ID	NUMBER (10)	Primary key
PASSWORD	VARCHAR2(20)	Not null
CITY	VARCHAR2(15)	Not null
STATE	VARCHAR2(15)	Not null
EMAIL	VARCHAR2(15)	Null

7. DATA FLOW DIAGRAM

Data flow diagrams (DFDs) reveal relationships among and between the various components in a program or system. DFDs are an important technique for modelling a System's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFDs consist of four major Components: entities, processes, data stores, and data flows. The symbols used to depict how these components interact in a system are simple and easy to understand; However, there are several DFD models to work from, each having its own symbology. DFD syntax does remain constant by using simple verb and noun constructs. Such a syntactical relationship of DFDs makes them ideal for object-oriented analysis and parsing functional specifications into precise DFDs for the systems analyst.

In the DFD, four symbols are used, and they are as follows.

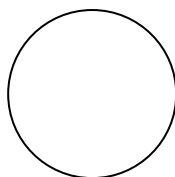
1. A square defines a source (originator) or destination of system data.



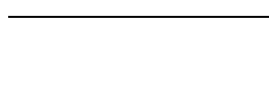
2. An arrow identifies data flow-data in motion. It is a pipeline through which information flows.



3. A circle or a "bubble" (Some people use an oval bubble) represents a process that transfers incoming data flows into outgoing data flows.



4. An open rectangle is a data store-data at rest, or a temporary Repository of data.



Context Level Data Flow Diagram: -

This level shows the overall context of the system and its operating environment and shows the entire system as just one process. Online bookstore is shown as one process in the context diagram, which is also known as zero level DFD, shown below. The context diagram plays a key role in understanding the system and determining the boundaries. The main process can be broken into sub-processes and the system can be studied with more detail; this is where the first level DFD comes into play.

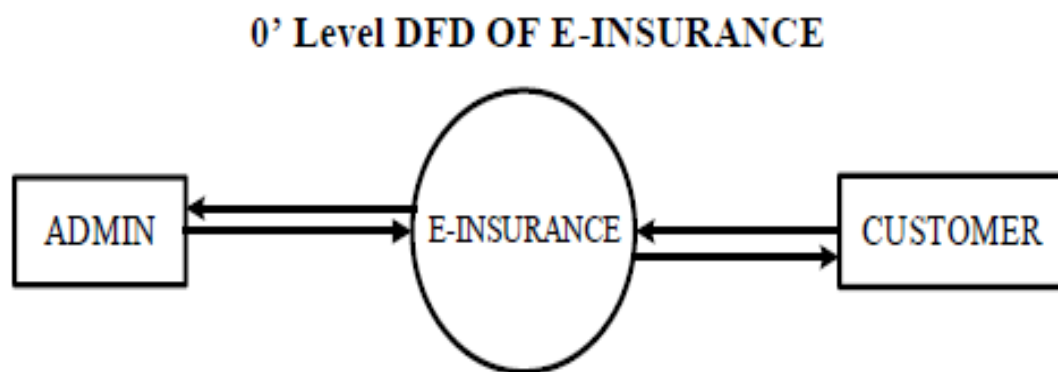


Fig 7.1

First Level DFD: -

This level (level 1) shows all processes at the first level of numbering, data stores, external entities, and the data flows between them. The purpose of this level is to show the major high-level processes of the system and their interrelation. A process model will have one, and only one, level-1 diagram. A level-1 diagram must be balanced with its parent context level diagram, i.e., there must be the same external entities and the same data flows, these can be broken down to more detail in level 1.

1' Level DFD OF E-INSURANCE

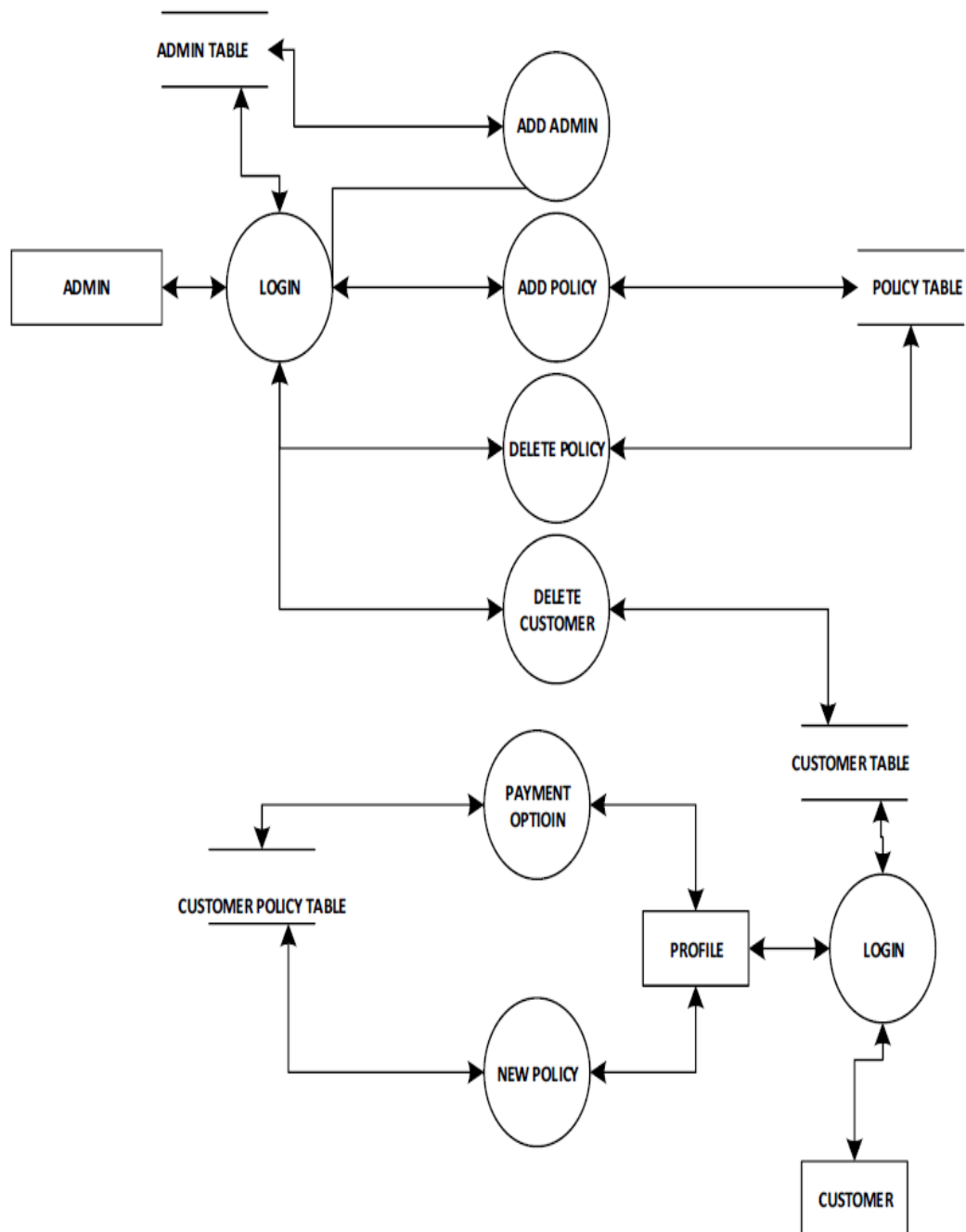


Fig 7.2

8. USE CASE DIAGRAM

A use case diagram is a graphical depiction of a user's possible interactions with a system. It shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses¹. Use case diagrams are typically developed in the early stage of development and people often apply use case modelling for the following purposes: specify the context of a system, capture the requirements of a system, validate a systems architecture, drive implementation, and generate test cases³.

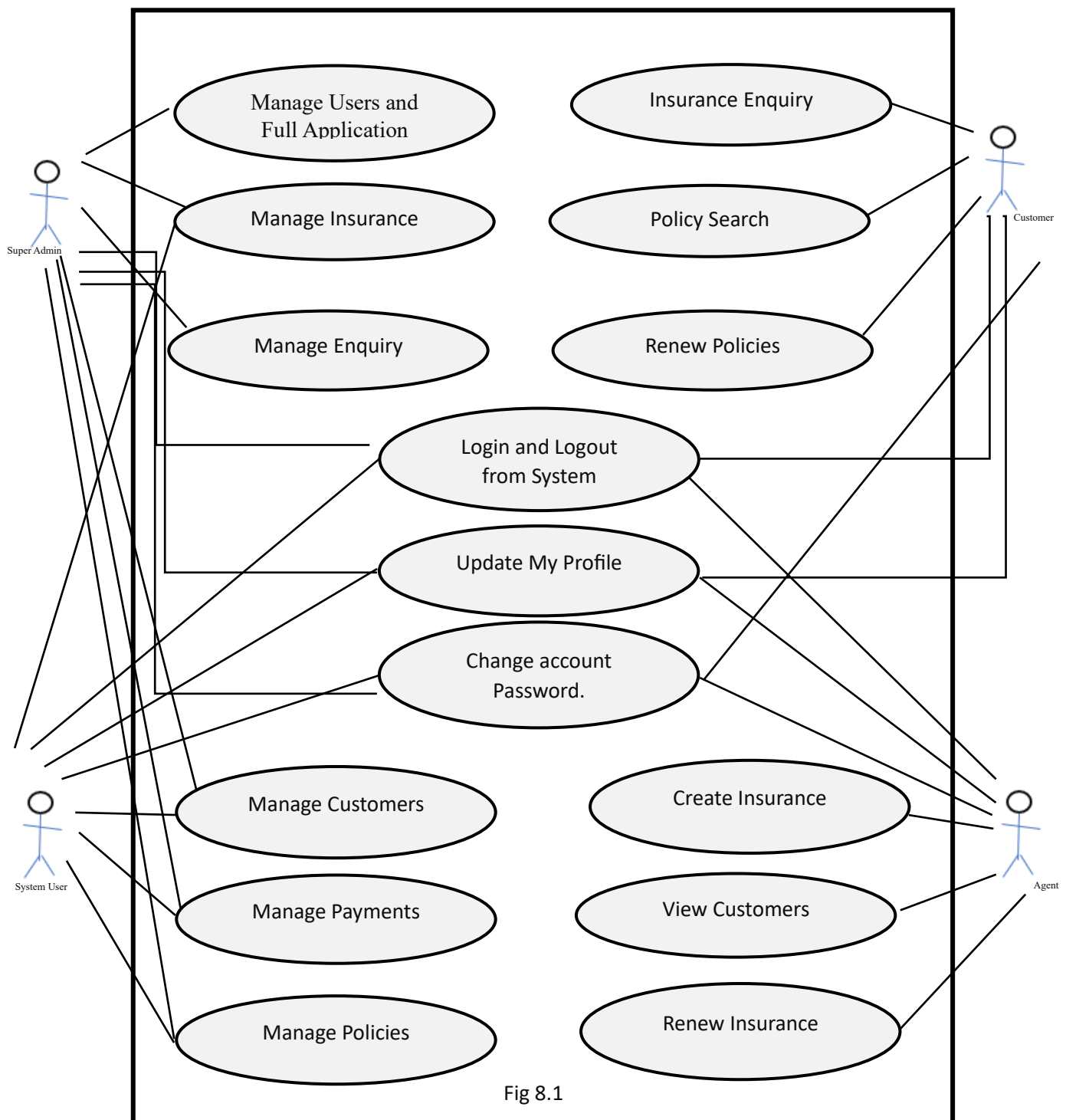


Fig 8.1

9. ER-DIAGRAM

ER-Modelling is a data modelling method used in software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modelling methods are called Entity-Relationship Diagrams or ER diagrams or ERDs.

An Entity-relationship model (ER model) describes the structure of database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of ER model are entity set and relationship set.

For example, an ER diagram representing the information system for a company's sales department might start with graphical representations of entities such as the sales representative, the customer, the customer's address, the customer's order, the product, and the warehouse. Finally, cardinality notations define the attributes of the relationship between the entities. The three main cardinal relationships are:

- **One-to-one (1:1).** For example, if each customer in a database is associated with one mailing address.
- **One-to-many (1:M).** For example, a single customer might place an order for multiple products. The customer is associated with multiple entities, but all those entities have a single connection back to the same customer.
- **Many-to-many (M: N).** For example, at a company where all call centre agents work with multiple customers, each agent is associated with multiple customers, and multiple customers might also be associated with multiple agents.

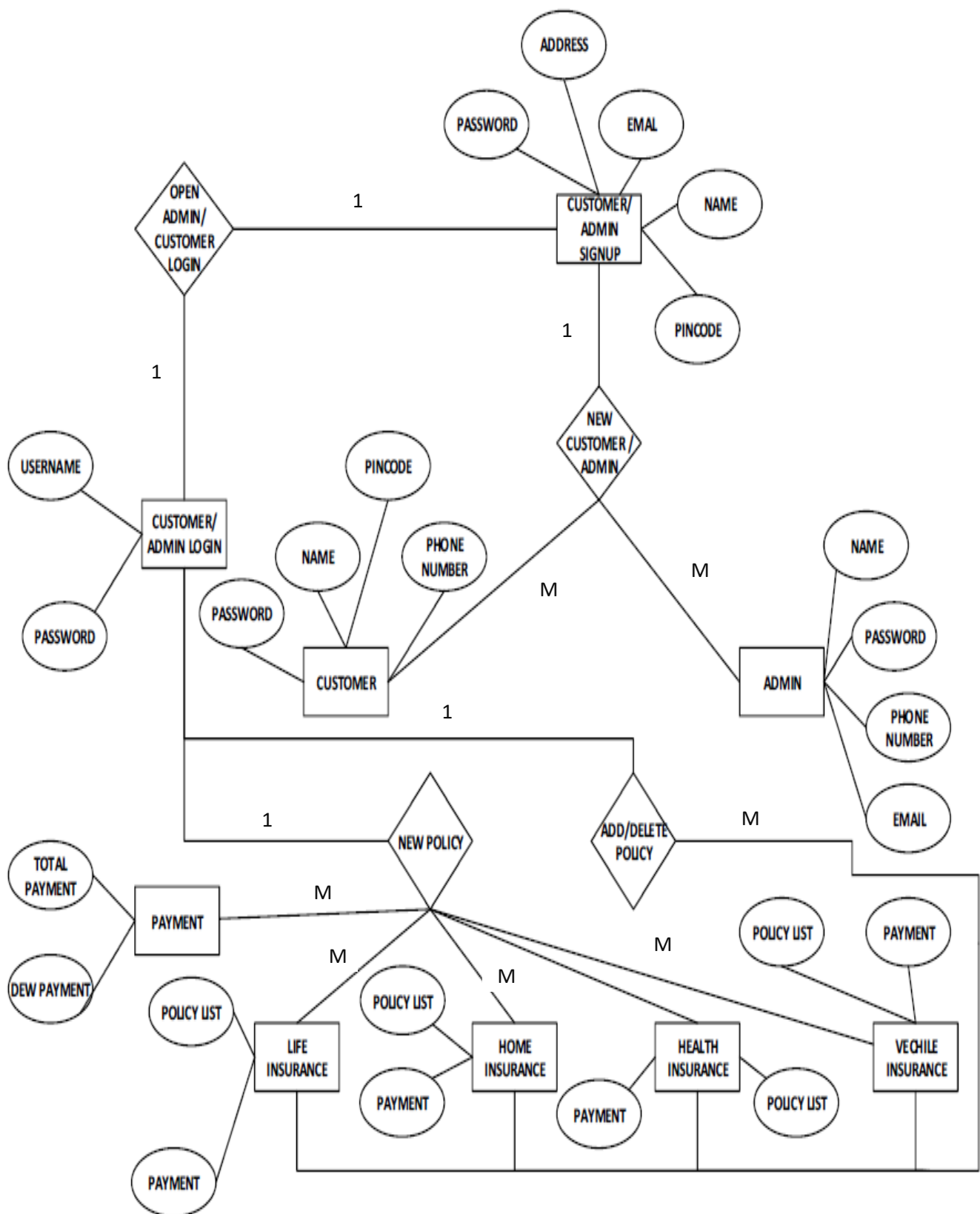


Fig 9.1

10. ADVANTAGES OF PROJECT

An insurance agency management system (AMS) can help insurance agencies streamline their operations and improve their customer service⁵. Here are some of the advantages of using an AMS:

- Prevent siloed data that is scattered around.
- Automate repetitive processes.
- Streamline system procedures.
- Ensure the holistic access and view of all workflows.
- Enhance agent productivity.
- Ensure total control over insurance policy.
- Effectively manage customers' claims
- Retain customers and provide a highly positive experience.

11. APPLICATION OF PROJECT

An insurance management system is a software that helps insurance companies, agents, brokers, and customers to manage various aspects of insurance policies and claims. It can have different features and benefits depending on the type and scope of the software. Some of the common types of insurance software are:

- **Document management software:** This software helps to store, organize, and access important documents related to insurance, such as policy applications, contracts, proofs, etc. It ensures security and compliance of the documents.
- **Customer relationship management software (CRM)/Insurance lead management software:** This software helps to manage the interactions with potential and existing customers, such as capturing leads, tracking sales activities, sending quotes, renewing policies, etc. It also helps to segment customers based on their needs and preferences. ¹
- **Insurance workflow automation software:** This software helps to automate various business processes that involve human resources, sales and marketing, lead management, etc. It reduces manual work and errors and improves efficiency and productivity. An example of workflow automation in insurance is STP (Straight-through Processing) or digital sales.
- **Claims management software:** This software helps to process and track claims from customers, such as verifying eligibility, assessing damages, calculating pay-outs, etc. It also helps to communicate with customers and vendors and resolve disputes.
- **Policy administration software:** This software helps to create, issue, and manage insurance policies for different products and services, such as life, health, property, etc. It also helps to update policy information, comply with regulations, and generate reports.

12. LIMITATION OF PROJECT

An insurance management system is a software that helps insurance companies, agents, brokers, and customers to manage various aspects of insurance policies and claims. However, like any other software, it also has some limitations that may affect its performance and usability. Some of the common limitations are:

- **Not suitable for all organizations:** Some insurance management systems may not be useful or compatible for all types of insurance companies or products. They may have different features, functions and requirements that may not match the needs and preferences of the users. Therefore, it is important to choose the right software that suits your business goals and requirements.
- **Expensive:** Insurance management systems can be quite expensive to acquire, install, maintain, and update. They may require a lot of hardware, software, and network resources to run smoothly and securely. They may also involve licensing fees, subscription fees or other charges that may add up to the cost. Therefore, it is important to consider the cost-benefit analysis of investing in an insurance management system.
- **Training costs:** Insurance management systems can be complicated to use and understand. They may require a lot of training and education for the users to learn how to operate them effectively and efficiently. They may also require constant updates and changes to keep up with the changing regulations, standards, and customer expectations. Therefore, it is important to provide adequate training and support for the users of an insurance management system.
- **Loss of focus due to automation:** Insurance management systems can automate a lot of tasks and processes that involve human resources, sales and marketing, lead management, etc. However, this may also result in a loss of focus or attention from the users. They may rely too much on the software and neglect other aspects of their business or customer service. They may also lose touch with their customers or market trends and miss out on opportunities or feedback. Therefore, it is important to balance automation and human intervention in an insurance management system.
- **Data security issue:** Insurance management systems store a lot of sensitive and confidential data related to insurance policies, claims, customers, etc. This data is vulnerable to cyberattacks, hacking, theft, or loss. If the data is compromised or

corrupted, it may cause a lot of damage and liability for the users and the customers. Therefore, it is important to ensure data security and protection in an insurance management system.

- **Policy limitations:** Insurance management systems are subject to the limitations of the insurance policies they manage. These limitations are the maximum amount of money that an insurance company will pay out for a claim in a policy period. If the losses exceed the limitations for a policy period and the policyholder has no additional insurance, then they themselves will be responsible for the excess losses. ⁵ These limitations are put in place because it would not be feasible for insurance companies to pay out an infinite amount of money for all the losses their policyholders experience.

13. REFERENCES

Designing and implementation phase: -

1. Software engineering: a practitioner's approach by roger s pressman.
2. System analysis and design by Elias m. Ewad.
3. DBMS: Bipin C Desai

Coding phase: -

1. PHP (BIBLE)
2. PHP (BLACK Book)
3. PHP (Complete Reference)

Referenced Sites:

www.w3school.com

www.php.net