



الجامعة السورية الخاصة  
SYRIAN PRIVATE UNIVERSITY

الجمهورية العربية السورية  
الجامعة السورية الخاصة  
كلية الهندسة المعلوماتية  
قسم الهندسة المعلوماتية -  
هندسة البرمجيات ونظم المعلومات

## نظام إدارة شركات التأمين

مشروع (تخرج 2) - قدم إكمالاً لمتطلبات الحصول على درجة البكالوريوس في هندسة  
المعلوماتية- هندسة البرمجيات ونظم المعلومات

إعداد

زين العابدين روماني

بشار شيخاني

زيرفا شيخو

إشراف

الدكتور أكرم مسوح

المهندس شادي بليدي

السنة الأكاديمية

2024/2023

**Syrian Arab Republic  
Syrian Private University  
Faculty of Computer &  
Information Engineering.  
Department of Software &  
Information Systems Engineering.**



## **Management System Insurance – Phase III**

**A Report Submitted in Partial Fulfillment of The Requirements For The Degree  
of Bachelor's Degree in Software and System Engineering Information For The  
Undergraduate Students**

### **Prepare By**

**Zain Alabdeen Roumani**

**Bashar Shiakhani**

**Zerva Shikho**

### **Supervised by**

**Dr. Akram Massouh**

**Eng. Shadi Blidi**

**Academic Year**

**2024/2023**

# Supervisor Certification

.....

prepared by

.....

was made under my supervision at Department of Computer and Information Engineering – Faculty of Computer & Communication Engineering in partial fulfillment of the Requirements for the Degree of Bachelors of Software and information System Engineering.

Academic Rank:..... Name:..... Signature:

Date:

# إهداء Dedication

إلهي لا يطيب الليل إلا بشكرك ولا يطيب النهار إلا بطاعتك ولا تطيب اللحظات إلا بذكرك، إلى نور الكون العظيم.  
إليك ربي

قيل عن الأب هو ذاك الشخص الذي تطلب منه نجمة فيعود لك حاملاً السماء.  
إلى من حصد الأشواك عن دربي ليمهد لي الطريق، إلى من كان لي خير سندٍ وصديق، إلى من كلله الله بالوقار وأحمل اسمه بكل افتخار.

والذي العزيز

إلى من كانت سماءً أنعمت عليّ بفيضها ورمزاً للعطاء والتضحية، إلى من كان لدعائها المبارك الأثر العظيم في تسيير سفينة العلم حتى ترسي على هذه الصورة، إلى من الجنة تحت قدميها.

أمي الحبيبة

إلى من يداها خطت لي درب الوصول وكانت بعطفها عنواناً للتفاؤل غير المعقول حملت لي مع كل كأسٍ من الشاي سكرًا كان محتواه أخي عليك بالعبور أنا بقربك لا تخف كُن دائماً مسرور.

أختي يارا

إلى رفيقة الاختصاص، إلى من بقلبيها العطوف ساندت كلَّ ملهوف، إلى المحبوبة القوية وصاحبة المواقف العفوية، رفيقة الخطوة الأخيرة وما قبل الأخيرة.

أختي المهندسة لين

إلى من كان لحنها دمها دوراً في تسهيل الصعب، إلى الطفلة ذات الثمانية عشر عاماً، إلى من حملت من الأزهار شداها ومن الشمس دفء قلبها إلى آخر من أنجبت أمي وختامها مسك.

أختي ليلاس

إلى السر الخفي وراء كل نجاح عظيم، إلى اليد المعطاءة التي كانت ومازالت تربت بحنانها على كتفي، إلى أمي التي لم تلدني، إلى صاحبة القلب الحنون المعطاء.

خالتي علا

إلى الرجل الذي كان أخي في كلّ المواقف، إلى من ساندني في دربي وخفف عني مرارة الطريق وكان الأمل  
لحناً دائماً في حديثه يتلوه على مسامعي بين حين وآخر.

صهري الغالي حسين

إلى من اتخذوا من القلب مقعداً، وجودي بقربهم هو معنى الأمان سندهم ودعمهم لم ولن ينسى ما حييت أبداً.  
أخوالي الأعزاء (حسن، محمد)

إلى من كان عطفهما كعطف أمي تخرج الكلمات من فمهما لتكون بلسماً لجراحي، أما راحة منزلهما كعطر  
النرجس التي تداعب روعي بطمأنينة لا توصف، حفظكما الله ودمتما لي.

جدي، جدتي

إلى زميلة المحادثات والمناقشات، إلى من أخذت من السماء وسع قلبها ومن حملت معاني اسمها قلباً وقالباً،  
شكراً لإخلاصك الصادق دوماً.

المهندسة سلام الداهاوك

إلى أصدقائي الذي تسكن صورهم وأصواتهم أجمل اللحظات والأيام، إلى شعلة الأمل والإصرار.

أصدقائي جميعاً

المهندس زين العابدين محمد سامر روماني

بسم الله الرحمن الرحيم

(يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ)

أود أن أهدي تخرجي إلى من أوصانا بهم الرحمن و إلى من زرعوا في قلبي بذور حب العلم والسعي نحو النجاح و إلى من سهروا ليالٍ طويلة من أجل راحتي، ومن استيقظوا فجرأ من أجل الدعاء لي و إلى جنة الله على الأرض و إلى من لا أسعى في هذه الدنيا إلا لنيل رضاهم و إلى من علموني و إلى صنعوا مني رجلاً قادراً على مواجهة الحياة

والدي ووالدتي الغاليين

إلى من الذي مهدت الطريق أمامي كي أحقق هدفي المنشود و إلى بسمة الحياة و نبع الأمل الذي يفيض على قلبي بالتفاؤل دوماً و إلى صاحبة الخلق الرفيع والذوق العالي و إلى من أتشارك معها الدم والصدق والحنان و إلى الوردة التي تنير حياتي

أختي العزيزة

إلى من شاركوني الأمل والأمل والنجاح والفشل و إلى المساند والداعم والكتف الذي أتكى عليه عندما تقرر الحياة أن تميل بي و إلى شركاء الدرب الطويل والطموح البعيد و إلى من فتحوا أفئدتهم لي عندما ضاقت بي دروب الحياة رغم اتساعها

أصدقائي الأحباء

فرحة التخرج وذلك الحلم الذي اتعيني، وسُقت عمري في سبيل ان احققه، ها هو قد تحقق واختلطت الدموع مع فرحة كنت أنتظرها لعمرى الذي مضى، اليوم ينتهي المشوار الذي كابدنا فيه وبعد فترة سنقول يا لها من ذكريات و الحمد لله دائماً وإلى الأبد.

المهندس بشار عدنان شيخاني

أهدي تخرجي للغالين للقلبي حفظكم الله ورعاكم وجزاكم عنا خيرا بحثت كثيرا في كتب ومجلات وغيرها لأهديكم تخرجي وأعبر لكم بمشاعري تجاهكم بانتقاء أعذب الكلمات وأرقى العبارات لتليق بمقامكم فما وجدت سوى كلماتي البسيطة من مشاعر صادقة من القلب للقلب أشكركم من أعماق قلبي ساندتموني ووقفتم بجانبتي طيلة حياتي حتى وصلت بفضل الله ثم انتم إلى إتمام دراستي الجامعية و إلى بحر الخُب والحنان والنبض الساكن في عروقي.

أبي الغالي وأمي الغالية

إلى نجوم سمائي المتألئة وسندي في الحياة و إلى شمعة العطاء وأمل الغد المُشرق و إلى البيت الدافئ ونهر الحب الذي يفيض على الصفاء الذي أنعم عليّ به الله ألتفت إليها في جميع الأوقات والعين الساهرة، والعطاء المستمر دون مقابل ودون انقطاع.

أخوتي الأعزاء

إلى الذين كان لهم الدور البارز فيما وصلت إليه الآن و إلى من ساندني وشجعني و إلى من كنتم مصدر القوة والعزيمة و إلى من تحلوا بالإخاء وتميزوا بالوفاء والعطاء و إلى من بررقتهم في دروب الحياة السعيدة والحزينة سرت و إلى من كانوا معي على طريق النجاح والخير.

أصدقائي المقربين

لم أمتلك من شرف التخرج شيء لولا وجود والدي ووالدتي بحياتي، ففرحتي وتخرجي اليوم أهداها إليكم يا من وقفتم بجانبتي، وزرعتم فيني المثابرة والإصرار الذي أنا من خلاله هنا.

المهندسة زيرفا خوشمان شيخو

## شكر وتقدير Acknowledgements

إلى تلك المنارة التي كانت أول محطة لنا على طريق الحلم، فأهدتنا بعظمتها علما وأيام ستبقى محفورة في الذاكرة.

إلى من كان لعلمهم وحكمتهم الفضل في وصولنا لنيل إجازة في الهندسة المعلوماتية، إلى الدكاترة الأكارم في الجامعة السورية الخاصة- كلية الهندسة المعلوماتية، لكم منّا العرفان بالجميل وخالص المحبة والتقدير.

لجهودكم المبذولة ودعمكم المستمر لنا في سبيل إتمام هذا العمل، لكم منا جزيل الشكر والامتنان:

الدكتور مهيب النقري

الدكتور أكرم مسوح

الدكتور وسام الخطيب

الدكتور فادي ابراهيم

المهندس شادي بليدي

المهندسة ثراء أصلان



# Abstract

The idea of insurance is based on cooperation between members of the same society to bear the harm that befalls one of them and to distribute it over the whole.

As for the insurance mechanism, it is carried out by insurance companies or their complements by reinsurance companies, which organize the process by collecting premiums after determining them from each according to the chances and quantity of what is expected to be incurred in damage as a result of the realization of the insured risk.

And then redistributing it to those who were harmed from the total insured them according to the terms and conditions of the insurance contract.

We learned that there are many operations that they have to do manually, it takes a lot of time and causes many errors .

As a result of this, sometimes a lot of problems happen and they face many conflicts with customers.

To solve the above problem, and continue to keep records of insurance, customer seat availability, insurance price, order creation, customer registration and other things, we make this insurance system proposal.

By using this program, we can insure from any customer in an easy and simple way without the need to visit the insurance company for insurance subscriptions and electronic payments.

Our platform also allows the customer to subscribe for the first time through the website, and this is what distinguishes our project from other projects related to insurance companies, is the customer's ability to access the site without visiting the company.

This project provides all of the above effectively, with protection, performance, and high privacy, in the right way.

## المخلص

تستند لفكرة التأمين الى التعاون بين افراد المجتمع الواحد على تحمل الضرر الذي يصيب احدهم و توزيعه على مجموع .

واما عن الية التأمين يتم بواسطة شركات التأمين او مكملاتها شركات اعادة التأمين , وهي التي تنظم العملية عن طريق تحصيل الاقساط بعد تحديدها من كل حسب فرص وكمية ما متوقع ان يصيبه من ضرر نتيجة لتحقق الخطر المؤمن منه .

ومن ثم اعاده توزيعها على من تضرر من مجموع المؤمن لهم وفقا لشروط و احكام عقد التأمين .

علمنا أن هناك العديد من العمليات التي يتعين عليهم القيام بها يدويًا، فهي تستغرق الكثير من الوقت وتتسبب في العديد من الأخطاء. نتيجة لهذا، في بعض الأحيان تحدث الكثير من المشاكل وكانوا يواجهون العديد من النزاعات مع العملاء.

لحل المشكلة المذكورة أعلاه، ومواصلة الاحتفاظ بسجلات التأمين ، وتوافر المقاعد للعملاء، وسعر التأمين، وإنشاء الطلبات ، تسجيل العملاء وأشياء أخرى، نقدم هذا الاقتراح الخاص بنظام التأمين. باستخدام هذا البرنامج، يمكننا تأمين من أي عميل بشكل سهل و بسيط دون الحاجة الى زيارة الشركة المؤمنة من عمليات الاشتراك في التأمين والدفع الالكتروني .

كما توفر منصتنا هذه للعميل الاشتراك لأول مرة عن طريق الموقع وهذا ما يميز مشروعنا عن باقي المشاريع التي تخص شركات التأمين هي قدرة العميل الوصول الى الموقع دون زيارة الشركة. يوفر هذا المشروع كل ما ذكر اعلاه بشكل فعال وبحمائية وأداء وخصوصية عالية و بطريقة الصحيحة .

## Table of Contents

Supervisor Certification .....	3
Dedication إهداء .....	4
Acknowledgements شكر وتقدير .....	8
Abstract .....	9
المخلص .....	10
Index of shape : .....	16
Table Index : .....	18
Chapter One .....	20
Introduction and Definition of the Project .....	20
1 Introduction and Definition of the Project .....	21
1.1 Bringing The Problem:[1] .....	21
1.2 Project Concept : .....	21
1.3 Aim of the Project: [3].....	21
1.4 Documentation Description:.....	21
1.4.1 Project Study: .....	21
1.4.2 Theoretical study of the project: .....	22
1.5 Analytical and Design Study of the Project:.....	22
1.6 Conclusion and Recommendations : .....	22
1.7 Diagram System: .....	22
Chapter Two .....	23
Project topic and reference study .....	23
2 Project topic and reference study .....	24
2.1 Introduction: .....	24
2.2 the basic Concepts: .....	24
2.3 Reference study: .....	24
2.4 Importance of Services: .....	25
2.5 Type of Insurance:.....	25
2.6 Similar Application : .....	26

2.6.1	Arope Syria :	26
2.6.2	BindQH App:	26
2.6.3	TLCDRM App: .....	27
2.7	Comparison of Similar Applications .....	27
<b>Chapter Three .....</b>		
	<b>Analytical Study of the Proposed System .....</b>	<b>28</b>
<b>3</b>	<b>Analytical Study of the Proposed System .....</b>	<b>29</b>
3.1	Requirements Document :	29
3.1.1	Documentation and Auxiliary Reports:	29
3.2	Feasibility: .....	29
3.3	SRS Document: .....	31
3.3.1	Introduction:.....	31
	This platform aims to provide a comprehensive and integrated experience via the Internet for customers in various insurances. ....	31
3.3.2	System description: .....	31
3.3.3	Project Requirement: .....	31
3.3.4	Performance Requirement:.....	34
3.3.5	Design Requirements:.....	35
3.3.6	Project Limitations: .....	36
3.4	Gantt Chart: .....	37
3.5	Analysis chart of project.....	40
3.5.1	Use-Case Diagram :	40
3.5.2	Activity Diagram:.....	41
3.5.3	Sequence Diagram (Admin)(Log-In): .....	43
3.5.4	Sequence Diagram (Customer)(Sign-Up):.....	44
3.5.5	Sequence Diagram (Admin)(Add-Department):.....	45
3.5.6	Sequence Diagram (Admin)(Edit-Department):.....	46
3.5.7	Sequence Diagram (Admin)(Add-Employee): .....	47
3.5.8	Sequence Diagram (Admin)(Add-Task):.....	48
3.5.9	Sequence Diagram (Admin)(Edit-Profile):.....	49

3.5.10	Sequence Diagram (Customer)(Electronic-Payment):.....	50
3.5.11	Sequence Diagram (Customer-Affairs)(Check-Information-Customer): .....	51
3.5.12	Sequence Diagram (Insurance-Affairs)(Add-Insurance): .....	52
3.5.13	Sequence Diagram (Associates)(Enter Customer Visit Information): 53	
3.5.14	Use-Case Narrative :.....	54
3.6	Entity Relationship Diagram (ERD) : .....	67
3.7	Data Base Table :.....	68
3.7.1	Table Structure for admin : .....	68
3.7.2	Table Structure for Employee : .....	68
3.7.3	Table Structure for Tasks: .....	68
3.7.4	Table Structure for Customer : .....	69
3.7.5	Table Structure for Doctors:.....	69
3.7.6	Table Structure for New Customer Check: .....	69
3.7.7	Table Structure for Department: .....	70
3.7.8	Table Structure for Blog: .....	70
3.7.9	Table Structure for Company Account: .....	70
3.7.10	Table Structure for Payment:.....	70
3.7.11	Table Structure for Associates: .....	71
3.7.12	Table Structure for Visit: .....	71
3.7.13	Table Structure for Note: .....	71
3.7.14	Table Structure for Insurance:.....	71
3.7.15	Table Structure for Insurance Request: .....	72
3.8	Unit Testing: .....	72
3.8.1	Unit Testing of the Add Employee: .....	72
3.8.2	Unit Testing of the Login: .....	73
3.8.3	Unit Testing of the Check Customer Information: .....	74
3.9	Requirements Tracking Matrix:.....	75
3.10	.....System Architecture:	76

3.10.1	Client-Server Architecture: .....	77
3.10.2	Features of this Architecture: .....	77
3.10.3	Why to Use? .....	77
3.10.4	Client-Server Diagram: .....	78
Chapter Four	.....	79
The Design Phase of the Proposed System	.....	79
4	Chapter Four-Design of the Proposed Project.....	80
4.1	Class Diagram: .....	80
4.2	Design Models at the Physical Level: .....	81
4.3	Web Page Site: .....	82
4.4	Update Requirements Tracking Matrix: .....	83
5	Chapter Five Implementation of Project.....	85
5.1	Programming Resource: .....	86
5.1.1	BootStrap5 (Front-End-Framework): .....	86
5.2	Programming Tools: .....	86
5.2.1	XAMPP Local Host: .....	86
5.3	Technologies Used: .....	87
5.3.1	Html5:.....	87
5.3.2	CSS3: .....	87
5.3.3	JAVASCRIPT: .....	87
5.3.4	PHP: .....	87
5.3.5	SMTP: .....	88
5.4	Interfaces Pages: .....	88
5.4.1	(Admin) Log-In Page:.....	88
5.4.2	(Admin) Dashboard: .....	88
5.4.3	(Customer Affairs) View Customer Details: .....	88
5.4.4	(Admin) Add Task to Employee:.....	88
5.4.5	(Admin) Profile: .....	89
5.4.6	(Admin) Mange Employee: .....	89

5.4.7	(Customer) Electronic Payment: .....	89
5.4.8	(Customer) Register a New Insurance: .....	89
5.4.9	(Finance Officer) Payment of Customer Insurance: .....	89
5.4.10	(Customer Affairs) Check Customer Information: .....	90
5.4.11	(Associates) Enter Customer Visit Information: .....	90
5.4.12	(Insurance Affairs) Mange Insurance: .....	90
5.5	Interfaces Screen: .....	91
Figure (5.35):	Insurance Affairs – Edit Insurance .....	96
6	Chapter Six Test and Result .....	97
6.1	Introduction: .....	98
6.1.1	Tests During Development: .....	98
6.1.2	Tests After Development: .....	98
6.1.3	Testing Result (In Performance): .....	99
Chapter Seven	.....	102
Consultation and Future Prospects	.....	102
7	Conclusion and Future Prospects.....	103
7.1	Conclusion :.....	103
7.2	Future prospects .....	103
8	Chapter Eight References .....	104

## Index of shape :

Figure (3.1): Gantt Chart(1) .....	37
Figure (3.2): Gantt Chart(2) .....	38
Figure (3.3): Gantt Chart(3) .....	39
Figure 4 :Use-Case Diagram .....	40
Figure (3.5): Activity-Diagram (Check Customer Information).....	41
Figure (3.6): Activity-Diagram (Delete News).....	41
Figure (3.7): Activity-Diagram (Electronic Payment).....	42
Figure (3.8): Activity-Diagram (Login Admin).....	42
Figure (3.9): Sequence-Diagram (Admin)(Login) .....	43
Figure (3.10): Sequence-Diagram (Customer)(Sign-Up) .....	44
Figure (3.11): Sequence-Diagram (Admin)(Add-Department) .....	45
Figure (3.12): Sequence-Diagram (Admin)(Edit-Department) .....	46
Figure (3.13): Sequence-Diagram (Admin)(Add-Employee).....	47
Figure (3.14): Sequence-Diagram (Admin)(Add-Task) .....	48
Figure (3.15): Sequence-Diagram (Admin)(Edit-Profile) .....	49
Figure (3.16): Sequence-Diagram (Customer)(Payment-Electronic) .....	50
Figure (3.17): Sequence-Diagram (Customer-Affairs)(Check Information Customer) .....	51
Figure (3.18): Sequence Diagram (Insurance-Affairs)(Add-Insurance) .....	52
Figure (3.19): Sequence Diagram (Associates)(Enter Customer Visit Information) .....	53
Figure (3.20): Entity Relationship Diagram.....	67
Figure (3.21): Client-Server Diagram.....	78
Figure (4.22): Class Diagram.....	80
Figure (4.23): Web Page Site.....	82
Figure (5.24): Admin – Login Page .....	91
Figure (5.25): Admin – Dashboard Page .....	91
Figure (5.26): Admin – Add Task to Employee Page .....	92
Figure (5.27): Admin – Profile Page.....	92
Figure (5.28): Admin – Add New Employee Page .....	93
Figure (5.29): Customer – Electronic Payment Page .....	93
Figure (5.30): Customer – Register A New Insurance Page .....	94
Figure (5.31): Customer – Register A New Insurance Page .....	94
Figure (5.32): Customer Affairs – Check Customer Information Page .....	95
Figure (5.33): Associates - Enter Customer Visit Information Page .....	95
Figure (5.34): Insurance Affairs – Add Insurance Page .....	96



<b>Figure (5.35): Insurance Affairs – Edit Insurance .....</b>	<b>96</b>
<b>Figure (6.36): Testing Result-Arope App .....</b>	<b>99</b>
<b>Figure (6.37): Testing Result - BINDQH App.....</b>	<b>100</b>
<b>Figure (6.38): Testing Result – TLDCRM.....</b>	<b>100</b>
<b>Figure (6.39): Testing Result – Management Insurance System.....</b>	<b>101</b>

## Table Index :

<b>Table (1.1): Comparison of Similar Applications .....</b>	<b>27</b>
<b>Table (3.2): Use-Case Narrative Log-In .....</b>	<b>54</b>
<b>Table (3.3): Use-Case Narrative Payment of Client Insurance .....</b>	<b>54</b>
<b>Table (3.4): Use-Case Narrative Add Employee .....</b>	<b>55</b>
<b>Table (3.5): Use-Case Narrative Delete Employee.....</b>	<b>56</b>
<b>Table (3.6): Use-Case Narrative Edit Employee .....</b>	<b>56</b>
<b>Table (3.7): Use-Case Narrative Add Department .....</b>	<b>57</b>
<b>Table (3.8): Use-Case Narrative Edit Department .....</b>	<b>58</b>
<b>Table (3.9): Use-Case Narrative Delete Department.....</b>	<b>58</b>
<b>Table (3.10): Use-Case Narrative Edit Profile .....</b>	<b>59</b>
<b>Table (3.11): Use-Case Narrative Account Statement.....</b>	<b>60</b>
<b>Table (3.12): Use-Case Narrative Register a New Insurance .....</b>	<b>60</b>
<b>Table (3.13): Use-Case Narrative Electronic Payment .....</b>	<b>61</b>
<b>Table (3.14): Use-Case Narrative Edit Profile .....</b>	<b>61</b>
<b>Table (3.15): Use-Case Narrative View Insurance Details.....</b>	<b>62</b>
<b>Table (3.16): Use-Case Narrative Delete Blog.....</b>	<b>63</b>
<b>Table (3.17): Use-Case Narrative Associates-Enter Customer Visit Information .....</b>	<b>63</b>
<b>Table (3.18): Use-Case Narrative Check Customer Information .....</b>	<b>64</b>
<b>Table (3.19): Use-Case Narrative Delete Insurance .....</b>	<b>65</b>
<b>Table (3.20): Use-Case Narrative Payment of Client Insurance .....</b>	<b>65</b>
<b>Table (3.21): Use-Case Narrative Add Customer .....</b>	<b>66</b>
<b>Table (3.22): DataBase Tables(Admin Table) .....</b>	<b>68</b>
<b>Table (3.23): DataBase Tables(Employee Table).....</b>	<b>68</b>
<b>Table (3.24): DataBase Tables(Task Table) .....</b>	<b>68</b>
<b>Table (3.25): DataBase Tables(Customer Table).....</b>	<b>69</b>
<b>Table (3.26): DataBase Tables(New Customer Check Table) .....</b>	<b>69</b>
<b>Table (3.27): DataBase Tables(New Customer Check Table) .....</b>	<b>69</b>
<b>Table (3.28): DataBase Tables(Department Table) .....</b>	<b>70</b>
<b>Table (3.29): DataBase Tabels(Blog Table).....</b>	<b>70</b>
<b>Table (3.30): DataBase Tabels(Company Account) .....</b>	<b>70</b>
<b>Table (3.31): DataBase Tabels(Payment Table) .....</b>	<b>70</b>
<b>Table (3.32): DataBase Tables (Associates) .....</b>	<b>71</b>
<b>Table (3.33): DataBase Tables (Visit) .....</b>	<b>71</b>
<b>Table (3.34): DataBase Tables (Note) .....</b>	<b>71</b>
<b>Table (3.35): DataBase Tables (Insurance) .....</b>	<b>71</b>

<b>Table (3.36): DataBase Tables (Insurance Request) .....</b>	<b>72</b>
<b>Table (3.37): Unit Testing Add Employee .....</b>	<b>72</b>
<b>Table (3.38) Unit Testing Login.....</b>	<b>73</b>
<b>Table (3.39): Unit Testing Check Customer Information.....</b>	<b>74</b>
<b>Table (3.40): Requirements Tracking Matrix.....</b>	<b>75</b>
<b>Table (4.41): Update Requirements Tracking Matrix .....</b>	<b>83</b>

**Chapter One**

**Introduction and Definition of  
the Project**

## **Chapter One- Introduction and Definition of the Project**

### **1 Introduction and Definition of the Project**

#### **1.1 Bringing The Problem:[1]**

In light of the development witnessed by the world, each company has its own system .

And here lies the problem that we want to solve, which is to secure a system for insurance companies that is an alternative system for traditional papers and processes that take time and effort on the part of customers and employees.

A system that the client, employee and manager can deal with until the system is automated smoothly and completely and with the correct information.

#### **1.2 Project Concept :**

Is a social system that aims to create a reserve to face the uncertain losses that individuals and institutions are exposed to by transferring the burden of risk from one person to several people or a group of people. That is, it is a system designed to reduce the uncertainty of financial losses by shifting the burden of risk.

#### **1.3 Aim of the Project: [3]**

The project aims to provide a suitable management platform for the insurance company that has the availability, where the administrator and employees can deal with the system without being restricted by the factor of time and place.

It also has ease of use and flexibility, as it provides good and appropriate interactive interfaces for the customer and the person responsible for him.

The interfaces are also presented in a coordinated and well-thought-out design.

#### **1.4 Documentation Description:**

This document provides a detailed overview of the stages of project completion, as well as: studying the project, defining the objectives of the project and the method of its development. The project was studied theoretically and then tried to project it onto the practical side of the project ..

##### **1.4.1 Project Study:**

Define the objectives of the project and the way to develop it

### **1.4.2 Theoretical study of the project:**

The project was studied in theory and then tried to project it onto the practical side of the project.

### **1.5 Analytical and Design Study of the Project:**

Include a detailed description of project components and how these components are analyzed, designed, investigated, and tested

### **1.6 Conclusion and Recommendations :**

This chapter presents the lessons learned from the project and the development team's suggestions for modification and development.

### **1.7 Diagram System:**

The system diagram for an insurance company management system project is a comprehensive system design that includes all the components and elements required for the effective development and operation of the system. This diagram includes a set of drawings and diagrams that accurately describe each part of the system, from the database to the user interface.

The system diagram includes a set of components, including:

1. Database: Contains all important information about partners, customers, premiums and the company's financial performance.
2. User Interface: It makes it easier for individuals to use the program, as they can add new installments or update information about their partners.
3. File Management System: Helps to organize the company's important files and documents.
4. Reporting System: Helps generate financial and statistical reports on the company's performance.
5. Security System: Ensures that company data is protected and secured from unauthorized access.
6. Access Control System: It helps define the level of access for users and define their permissions.
7. Storage system: It helps to store all information in an organized and secure manner.
8. Task management system: helps keep track of the tasks and tasks that need to be completed in the company.

## **Chapter Two**

### **Project topic and reference study**

## Chapter two - Project topic and reference study

### 2 Project topic and reference study

#### 2.1 Introduction:

Introduction: In this chapter, we will study the topic of the project and the reference study for this health project

#### 2.2 the basic Concepts:

Insurance or the system of insurance or guarantee is a means to face the risks that a person is exposed to in his entity or money during his life in order to mitigate its impact. The essence of this method is the cooperation that is achieved with the participation of people exposed to the same danger in facing the effects that result from its achievement in relation to some of them, by paying each of them a subscription or an installment, and the collected amounts are collected and then distributed to those who are affected by the disaster. Thus, the effects of the disaster on those involved in achieving this cooperation will be achieved.

Insurance is a practical reality, and it is one of the best means that enables a person to mitigate the effects of disasters, whether these disasters occurred by the person himself, by his negligence or negligence, or by the actions of others. It is a means of safety that is consistent with the spirit of the modern era, in which the requirements of life have increased, and the danger of machinery has increased, and the dangers of development in it have become clear.

The effectiveness of insurance, as the modern means of confronting risks and its effects, was the most prominent reason that led to its prosperity, the diversity of its fields, its development, and its extension to different fields, to insure individuals from every risk they are exposed to, whether with their money or their people. The effectiveness of insurance also led to some countries imposing some types of insurance to ensure that certain groups of people would receive compensation for a specific accident.

The insurance system assumes the existence of a legal instrument that regulates the relationship between the insurer and the insured, which is the insurance contract. However, insurance is not limited to this aspect. In addition, insurance is a technical process in which insurance companies use technical means in order to be able to achieve their goals in covering the risks that occur. They use the technical elements necessary to manage insurance operations, such as the rules derived from the science of statistics and the clearing system between Risks and the application of the law of abundance, in addition to the rules of financial management. Thus, it will be able to achieve the goal of covering the risks that befall the insured, and at the same time the insurer - the insurance company - will be able to manage the insurance project in a way that achieves its investment purposes, to reap some profit and contribute to achieving the purposes of the national economy.

#### 2.3 Reference study:

Electronic insurance is an insurance system that is managed and implemented electronically, using modern technology and fast communications. This system aims to facilitate the process of buying and managing insurance, improving the quality of services provided to customers, and reducing costs and time spent in issuing policies and other documents. Electronic insurance also provides innovative



solutions to the challenges facing the insurance industry, such as high rates of fraud and financial losses, and changes in consumer and market behavior.

Electronic insurance differs from traditional paper insurance in many aspects, including:

1. **Management:** Electronic insurance is managed electronically, while traditional paper insurance is managed manually.
2. **Speed:** Policies and other documents are issued faster in electronic insurance due to the use of modern technology.
3. **Cost:** Traditional paper insurance requires higher costs for issuing policies and other documents, while costs are reduced in electronic insurance.
4. **Access:** Customers can easily access their data and insurance details in the electronic insurance, while they need to visit the offices in the traditional paper insurance.
5. **Security:** The level of security in electronic insurance is improved due to the use of encryption and special programs to protect data, while traditional paper insurance can be exposed to theft and loss.

#### **2.4 Importance of Services:**

1. **Saving time and effort:** E-insurance facilitates the process of managing and issuing policies and other documents faster and more effectively, which saves time and effort for companies and customers.
2. **Reducing costs:** Costs are reduced in electronic insurance because there is no need to use paper, ink and other materials needed in traditional paper insurance.
3. **Increased security:** Electronic insurance helps to increase the level of security due to the use of encryption and special programs for data protection, which protects personal information of customers.
4. **Improving customer service:** E-insurance facilitates access to insurance data and details, which improves customer service and helps solve problems faster.
5. **Environmental Sustainability:** Electronic insurance helps reduce the use of paper and materials needed to manage traditional paper insurance, which contributes to preserving the environment and improving sustainability.

#### **2.5 Type of Insurance:**

##### **1) Medical Insurance:**

is obtaining insurance coverage for the costs of treating medical conditions that the insured may be exposed to, simply because if he bears the cost himself without insurance, he can reach to bear unexpected large sums or resort to advances to cover the costs of treatment (in case of exposure to diseases whose treatment is expensive).

##### **2) Life Insurance:**

Life Insurance: Here the issue is a little different, because the person asks for insurance for his life for the benefit of the beneficiaries whose names are specified in the insurance policy, and

the duration of the insurance remains for a certain period, for example, for the age of 65 or 70, and it does not differ if the death occurred due to an accident or a natural one.

### 3) **Property Insurance:**

is one of the types of insurance that includes insurance on shops or private properties such as homes, offices and buildings, and the benefits of property insurance vary according to the type of things you want to insure.

### 4) **Car Insurance:**

It is the insurance that provides you with protection in the event of a traffic accident that causes injury to a person or damages his property

## 2.6 **Similar Application :**

### 2.6.1 **Arope Syria :**

We visited the Arope company located in Damascus and contacted the Human Resources Manager to allow us to take a look at the internal system of this system.

he noticed the following that their system is an internal system only, meaning that this system does not work with registration via the website.

Also, the system does not allow the customer to obtain his data or any data related to him electronically, but the employee is dealt with directly and face to face to obtain the information that the customer wants.

### 2.6.2 **BindQH App:**

Lightning fast, the agency's modern operating system enables BindHQs, MGAs, MGUs and wholesalers to connect more businesses, expand their profit margins and become more valuable businesses.

BindHQ's constantly improving cloud-based platform reduces total cost of ownership; Nothing needs to be maintained, lower upfront capital expenditures, and once you're live, product upgrades are continuous and automatic.

Pricing:

1. BindHQ Light : Starting at \$275 per month
2. BindHQ Pro Starting at \$375 per month
3. Enterprise: starting at \$475 per month

### 2.6.3 TLCDRM App:

It is specifically designed for the insurance sales industry, starting with health and life.

Performs sales or call centre operations based on multiple call centre workflows and we continue to add key features and amenities to the system weekly. The purpose of a CRM is to help the agency achieve the following goals: sales, communications, retention, commissions, and customer service.

Pricing:

1. Agency :Starting at \$75 per month

## 2.7 Comparison of Similar Applications

**Table (1.1): Comparison of Similar Applications**

المقارنة بين النظام الحالي و أنظمة المشابهة				
	Management System Insurance	Arope Syria System	BindHQ	TLCDRM
Easy to use				
Client has access				
Expensive				
Attractive Interface				

**Chapter Three**  
**Analytical Study of the**  
**Proposed System**

## Chapter Three- Analytical Study of the Proposed System

### 3 Analytical Study of the Proposed System

This chapter presents all aspects of the project in a practical way, where the functional and non-functional requirements of the project will be detailed, the mechanism for achieving them and the cases of their use, and then testing their validity and achieving the desired goals. Considering the analysis stage, the basis of the project development stages, any error in this stage leads to an error in all the following stages and can lead to the failure of the entire project, or at the very least the failure of one of the project's goals, defining concepts, objectives and the framework of the problem, deleting illogical concepts and understanding the project more.

#### 3.1 Requirements Document :

This chapter presents all aspects of the project in a practical way, as it will detail the functional and non-functional requirements of the project, the mechanism for achieving them and their use cases, then testing their validity and achieving the desired goals.

Given the analysis stage, and the basis of the project development stages, any error in this stage leads to an error in all the following stages and can lead to the failure of the entire project, or at least the failure of one of the projects.

In this project, information was collected in order to conduct an analytical study of the system in order to obtain documented and correct information, which leads to the success of the project

##### 3.1.1 Documentation and Auxiliary Reports:

Some references on the Internet were relied upon, in addition to benefiting from some previous reports and projects, as their importance was to give a broader and more comprehensive perception of the idea, while giving perceptions of the non-functional services that the system should provide.

#### 3.2 Feasibility:

The economic feasibility of implementing a distance education project is the process of evaluating and analysing the financial and economic aspects of the project. Economic feasibility aims to determine whether the project brings expected financial benefits and is worth investing in. The economic viability assessment includes several important factors, including:

1. Implementation cost: The total cost of developing and operating the project must be determined, including the cost of developing the educational platform, the cost of human resources, the cost of marketing, and any other related costs.

2. Projected revenue: Projected revenue should be estimated, such as subscription fees or monthly subscriptions for students and teachers, and sales through courses or ancillary services.
3. Maintenance and operation cost: The cost of maintaining and operating the project must be taken into account, including hosting and storage costs, updates and upgrades, and technical support.
4. Benefit Analysis: The expected benefits of the project should be identified, such as increasing the number of students enrolled, improving the quality of education, expanding access to education, and saving traditional travel and accommodation costs.
5. Cost-Benefit Analysis: A detailed cost-benefit analysis must be performed to assess whether expected benefits exceed expected costs and at what rate.
6. Risk Assessment: The potential risks associated with the implementation of the distance education project must be analyzed and their impact on the economic and financial aspects assessed.

Using this information and financial analyses, a comprehensive economic feasibility study can be prepared to evaluate the implementation of the distance education project and take the appropriate decision based on the results.

The project is to create a platform for managing insurance companies. Here is an example of a feasibility analysis for this project: (Please note that these figures are fictitious and are for illustrative purposes only and may vary depending on actual conditions and economic fluctuations):

1. Implementation cost:
  1. Development and design of the platform: 250,000,000 SP.
  2. Annual marketing costs: 20,500,000 SP.
  3. The total costs of the project in the first year: 270,500,000 SP.
2. Expected Revenue:
  1. Annual subscription fees for clients: 1,000,000 SYP to the client.
  2. The goal is to get 500 clients in the first year.
  3. Expected revenues in the first year: 500,000,000 SP.
3. Maintenances and Operation Cost:
  1. Annual hosting and operating costs: 50,000,000 SP.
  2. Annual technical support and updates costs: 20,500,000 SYP.
  3. Total costs of maintenance and operation in the first year: 70,500,000 SP.
4. Benefits Analysis:
  1. Provide remote subscription to customers in remote areas or not available for traditional subscription.
  2. Reducing costs for customers and insurance companies.
  3. Increase customer satisfaction.
  4. Work on the level of service improvement for them.
5. Cost Benefits Analysis:

The expected net return in the first year: Project revenues - implementation, maintenance and operation costs = 500,000,000 - 350,000,000 = 150,000,000 SP.

6. Risk Assessment:

1. Competition in the electronic insurance market.
2. Challenges of expansion and increasing the number of joint customers.
3. The impact of technology and technical dependence on project sustainability.

Based on this analysis of economic feasibility, we can make an informed decision regarding the implementation of a project to manage insurance companies, according to the expected return, implementation cost and potential risks. These numbers can be optimized and adjusted according to the circumstances and requirements of your individual project.

### 3.3 SRS Document:

A document used to describe the software requirements for a specific project. The document aims to document and define the functional and non-functional requirements for the system to be developed.

An SRS document usually contains the following components:

#### 3.3.1 Introduction:

This platform aims to provide a comprehensive and integrated experience via the Internet for customers in various insurances.

#### 3.3.2 System description:

1. The system is available to everyone at any time and anywhere.
2. The customer can subscribe to the site for the first time electronically without the need to visit the company.
3. The customer can pay online.
4. The system includes management of users, employees, departments, blogs, and insurances.

#### 3.3.3 Project Requirement:

##### 3.3.3.1 Functional Requirement:

Functional Requirements of the Admin:

1. Review of Completed and In Complete Problems on the System.
2. Review of Information on Each Problem.
3. Review of Employees in the system.
4. Review Dashboard For All Details.
5. Add Employee.
6. Edit Employee.
7. Delete Employee.
8. Send Message to Employee.

9. Add Department.
10. Edit Department.
11. Delete Department.
12. Edit Profile.

Functional Requirements of the Finance Officer:

1. Payment of Client Dues.
2. Edit Profile.

Functional Requirements of the Insurance Affairs:

1. Edit Profile.
2. Add Insurance.
3. Edit Insurance.
4. Delete Insurance.

Functional Requirements of the Customer:

1. Account Statement.
2. Electronic Payment.
3. View Insurance Details.
4. Register a New Insurance.

Functional Requirements of the Blogger Employee :

1. Edit Profile.
2. Add News.
3. Edit News.
4. Delete News.

Functional Requirements of the Customer Affairs :

1. Edit Profile.
2. Add Customer.
3. Modify Customer Information.
4. Delete Customer.
5. Check Customer Information.



Functional Requirements of the Associates:

1. Enter Customer Visits Information.
2. Send Notes.
3. Edit Profile.

### **3.3.3.2 Non-Functional Requirement:**

#### **3.3.3.2.1 Availability:**

System managers wishing to access the system can use it anywhere .

#### **3.3.3.2.2 Performance :**

1. Good performance is one of the main pillars of any successful work in the digital environment.
2. Ensure the web application are user-friendly and run smoothly.
3. This application works in the same way regardless of operating system environments.
4. The time taken for import file and display multimedia should be minimal
5. ( Ex : No more than 60 second )

#### **3.3.3.2.3 Reliability :**

- 1.1 Application of the authority system so that the system ensures that no user enters data or pages without its validity .
- 1.2 The system provides confirmation of entry and exit from the system .
- 1.3 The system preserves the data of system manage Ex :Preserve the privacy of the system participants through a private password

#### **3.3.3.2.4 Usability :**

Interfaces between the system and the system manager are user-friendly, clear and easy helping the user to handle the system well ) Easy to use by users in general by 80%)

#### **3.3.3.2.5 Security:**

1. **Digital Signature:** Which a type of cryptography used to simulate the security properties of its digital form, and not the ordinary, the digital has been added to each payment receipt process in which system performs.
2. **MD5 Algorithm:** is a cryptographic hash function algorithm that takes the message as input of any length and changes it into a fixed-length message of 16 bytes. We applied it to our project to encrypt some confidential information such as passwords and usernames

3. **Anti-SQL Injection:** Is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind web application. we prevent SQL injection attacks in many ways such as input validation and parametrized queries including prepared statements and turn off the visibility of database errors on your production sites and more ways.
4. **Generate Password:** we have created a function to generate strong password, then they are encrypted by md5, so it is impossible to decrypt them in any way. It is used when adding a new employee or customer to the system.
5. **Powerful Identity Verification and Call Session Management :** Strong set of authentication and control modules Manage sessions by way of using strong passwords, connection sessions with expiration times, and avoid using single sign on and put the hash variable in another hash variable.
6. **Secure Direct Object References :** Use indirect references to the private objects of each user or each communication session.
7. **Security configuration :** Using the latest versions of tools such as php 8.1 and MYSQL 8.0 and more , and a test application via PHP-Unit.
8. **Do not Exposure sensitive information :** Encrypt all this sensitive data in the case of storage and transmission in a way that protects it from these threats and disable the auto-complete feature when filling sensitive data and disable the “caching” feature for pages that contain sensitive data.

#### 3.3.4 Performance Requirement:

1. **Response Time:** Website Response time should be less than 2 Seconds to achieve a smooth and fast user experience, and images and content must be loaded quickly and efficiently to avoid delays in appearing to users.
2. **Resource Usage:** The RAM usage of the host server should be moderate and not exceed the specified limit to achieve stable and reliable performance, and Database utilization and efficient query execution should be optimized to reduce the time it takes to retrieve data and improve data read and write performance.
3. **Loading Capacity:** The site must withstand a high load of users and requests during peak periods, such as promotions or large sales, and the performance and endurance of

the site must be optimized when dealing with a large volume of data, such as product listings and orders.

4. **Application Response:** The application must have fast performance in carrying out calculations and operations related to money and inventory, such as adding products to the shopping cart or making payment operations, and the performance and speed of executing search and browsing operations must be improved to provide an interactive and smooth experience for users.
5. **Expandability:** The application must be expandable to accommodate increasing user volume and future requirements. The application architecture must be scalable and additional modules and components can be easily added without affecting the overall performance.

### 3.3.5 Design Requirements:

1. **User Interface:** The user interface must be simple and easy to use to enable customers and employees to navigate and interact easily. The customer interface must be attractive and attractive to customers by using appropriate graphic design and appropriate colors. The customer interface must be compatible with the various devices and screens used, including desktop computers, and tablets and smartphones.
2. **Page layout and structure:** The design of the pages should be a tidy and organized structure to enable users to find content and functionality easily. Each page should have a responsive design based on variable elements such as text size, images, and buttons based on the screen size and type of device used. Pages should be designed in a way that allows for quick and flexible navigation between The different sections of the platform.
3. **Database design:** The database must be designed effectively to store and manage customer data and system content. The database must be reliable and secure to protect sensitive data and prevent data loss or corruption.
4. **System security:** The system must be designed with high security standards to protect users' data and ensure the confidentiality of personal information and test results. Mechanisms must be implemented to combat hacking and protect the system from electronic attacks.

5. **Platform Compatibility:** The platform must be compatible with popular web browsers and operating systems to ensure accessibility and ease of use for customers.

### 3.3.6 Project Limitations:

1. **Technology Limitations:** There may be technical limitations related to programming requirements or the tools and technologies used in developing the Platform. We may have certain limitations on the scope of technical support, computer hardware and internet connection.
2. **Time Constraints:** We may experience time constraints in connection with scheduling and project execution. We must keep in mind the set schedule for project delivery and achieve the set targets in a timely manner.
3. **Resource Constraints:** We may have limitations on the financial, human and material resources available for the project. We must control the effective use of resources and achieve project objectives with the best use of available resources.
4. **Security and Privacy Restrictions:** We may experience security and privacy restrictions on the Learning Platform. We must follow data security practices, protect users' privacy, and provide a safe and secure environment for distance learning.
5. **Compatibility and responsiveness limitations:** We may experience limitations regarding the platform's compatibility with different devices, browsers and operating systems. We must ensure that the platform is compatible with a wide variety of devices and provide a prompt and appropriate response to the user experience.

### 3.4 Gantt Chart:

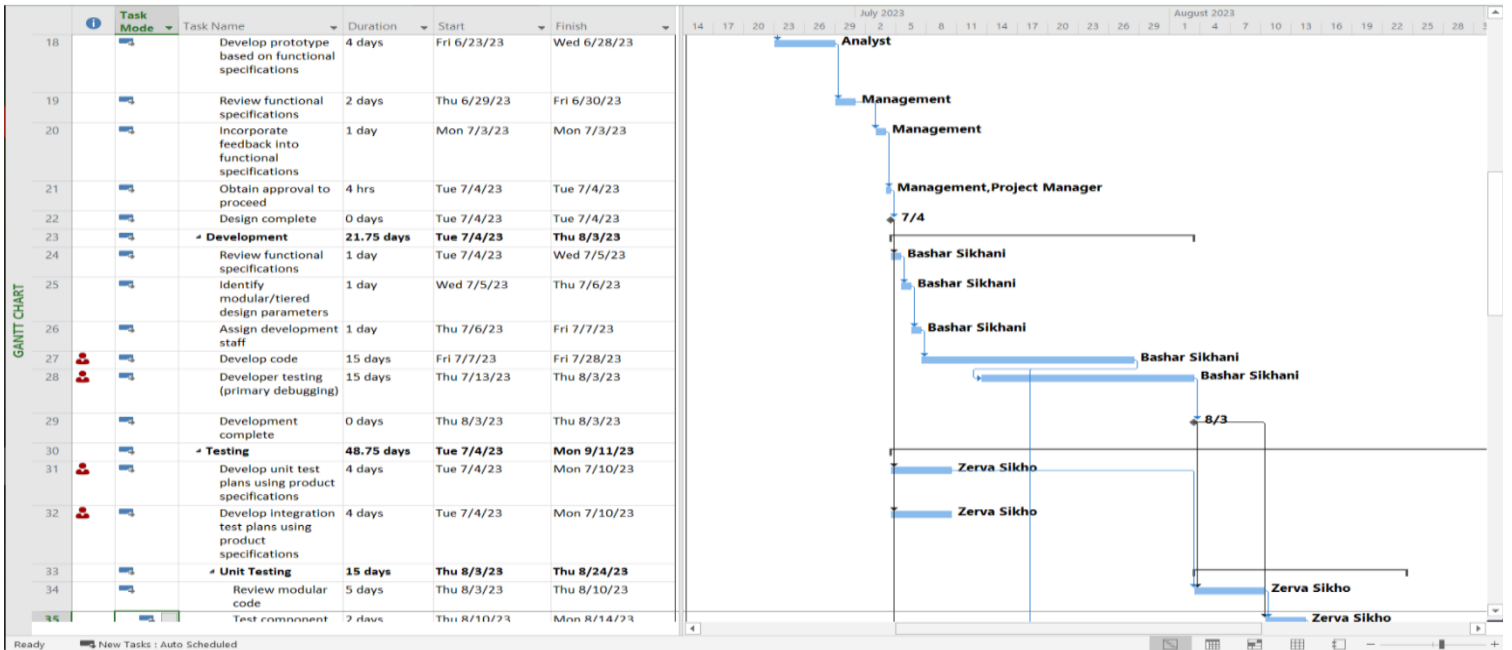
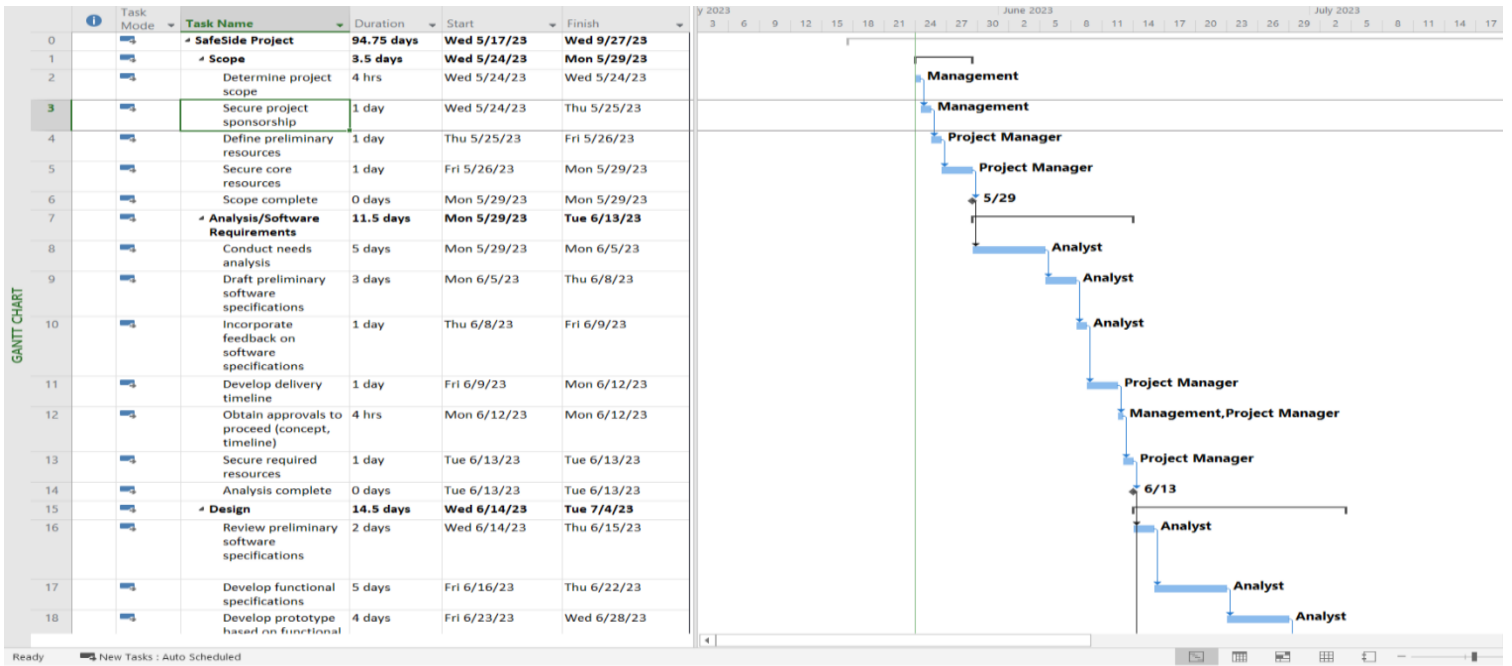


Figure (3.1): Gantt Chart(1)

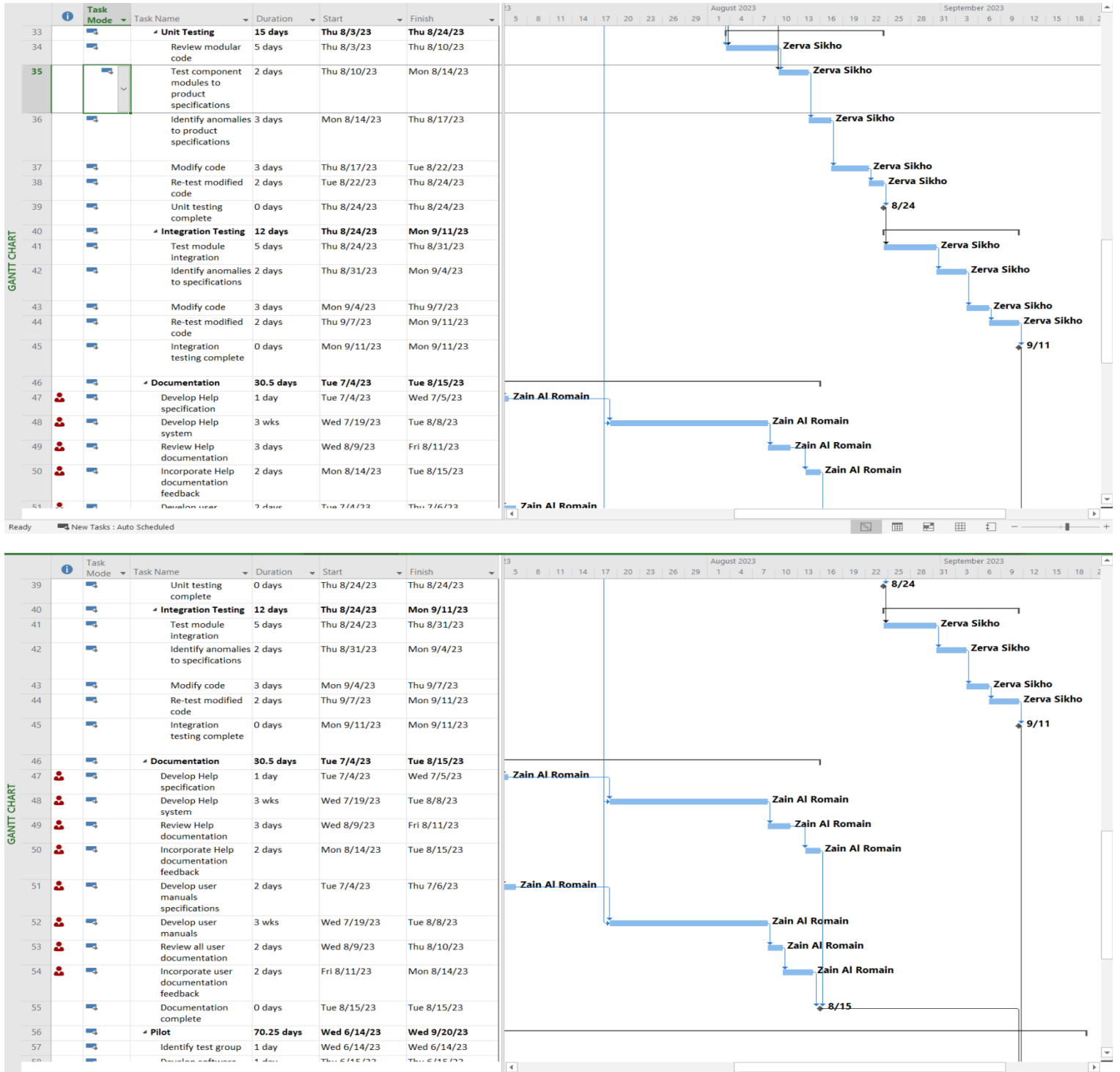


Figure (3.2): Gantt Chart(2)

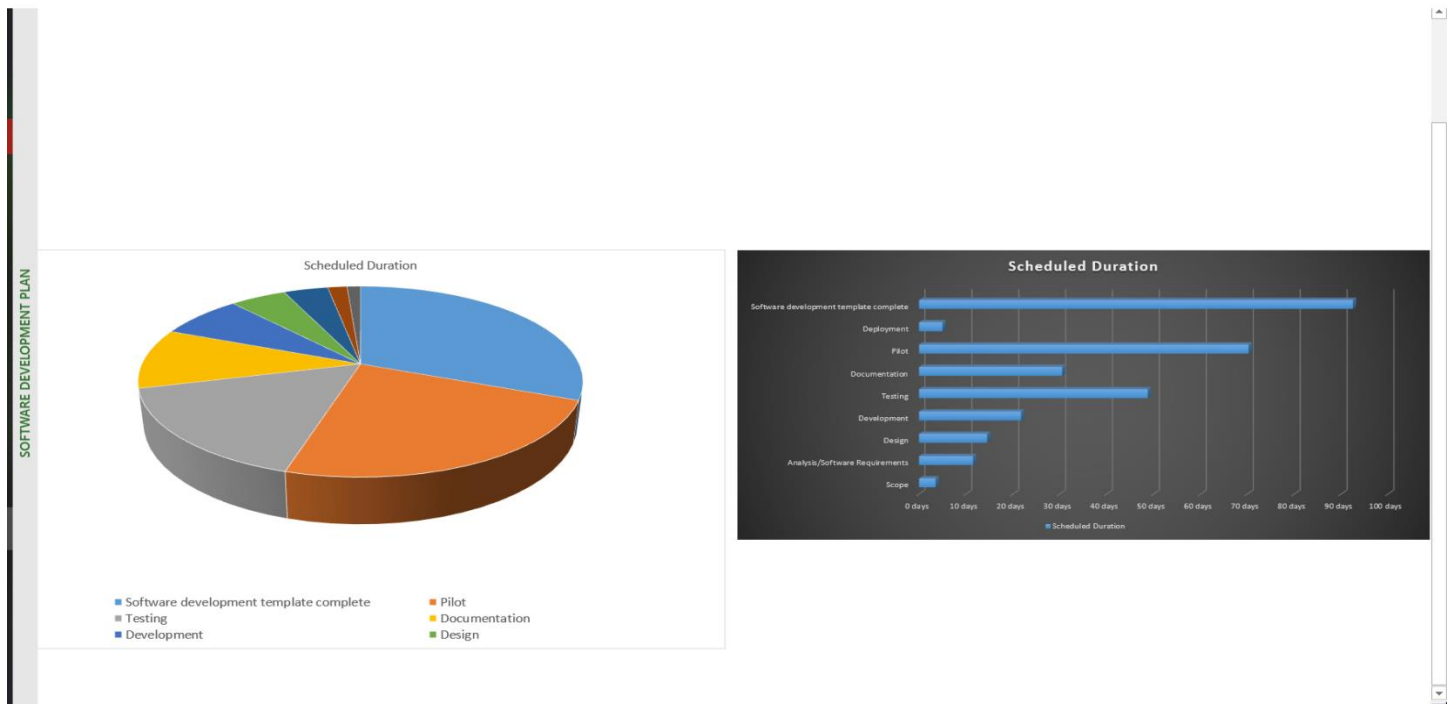
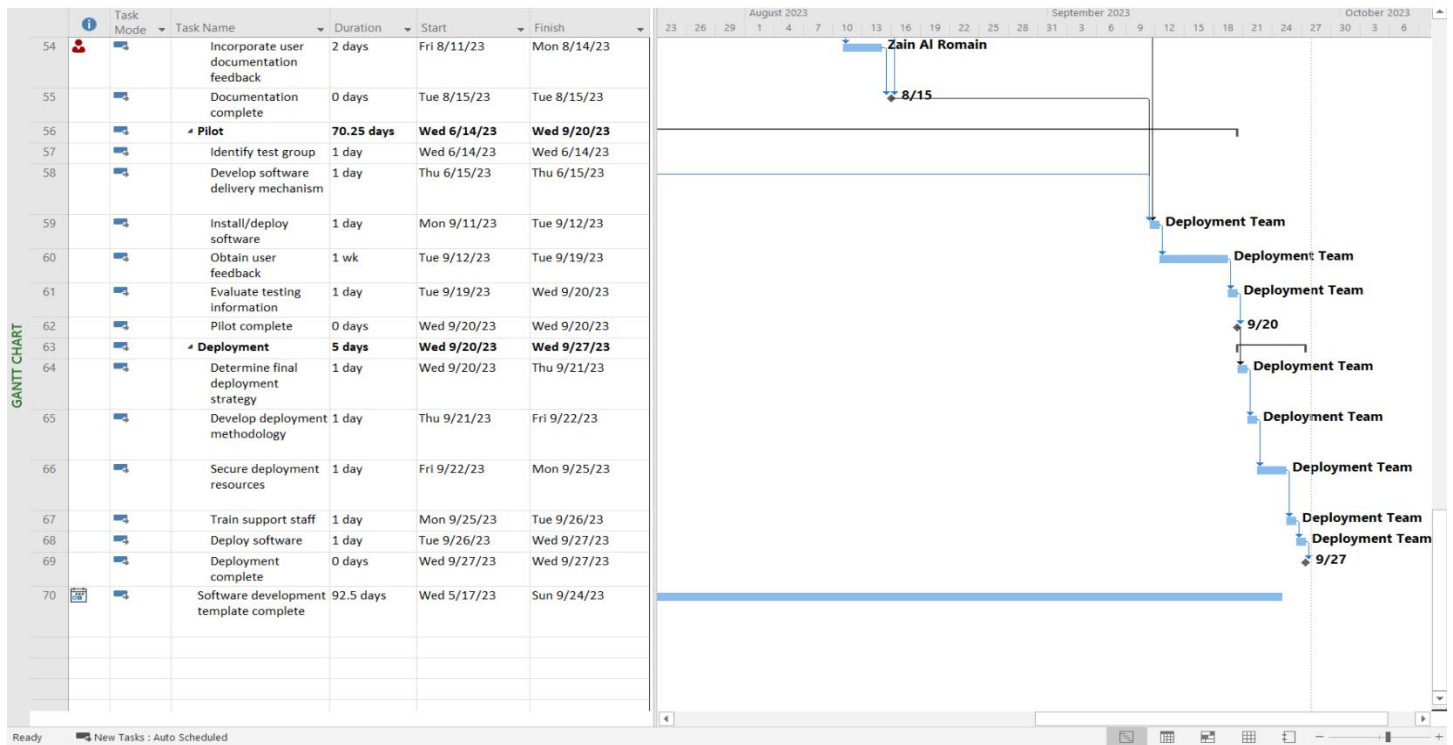


Figure (3.3): Gantt Chart(3)

### 3.5 Analysis chart of project

#### 3.5.1 Use-Case Diagram :

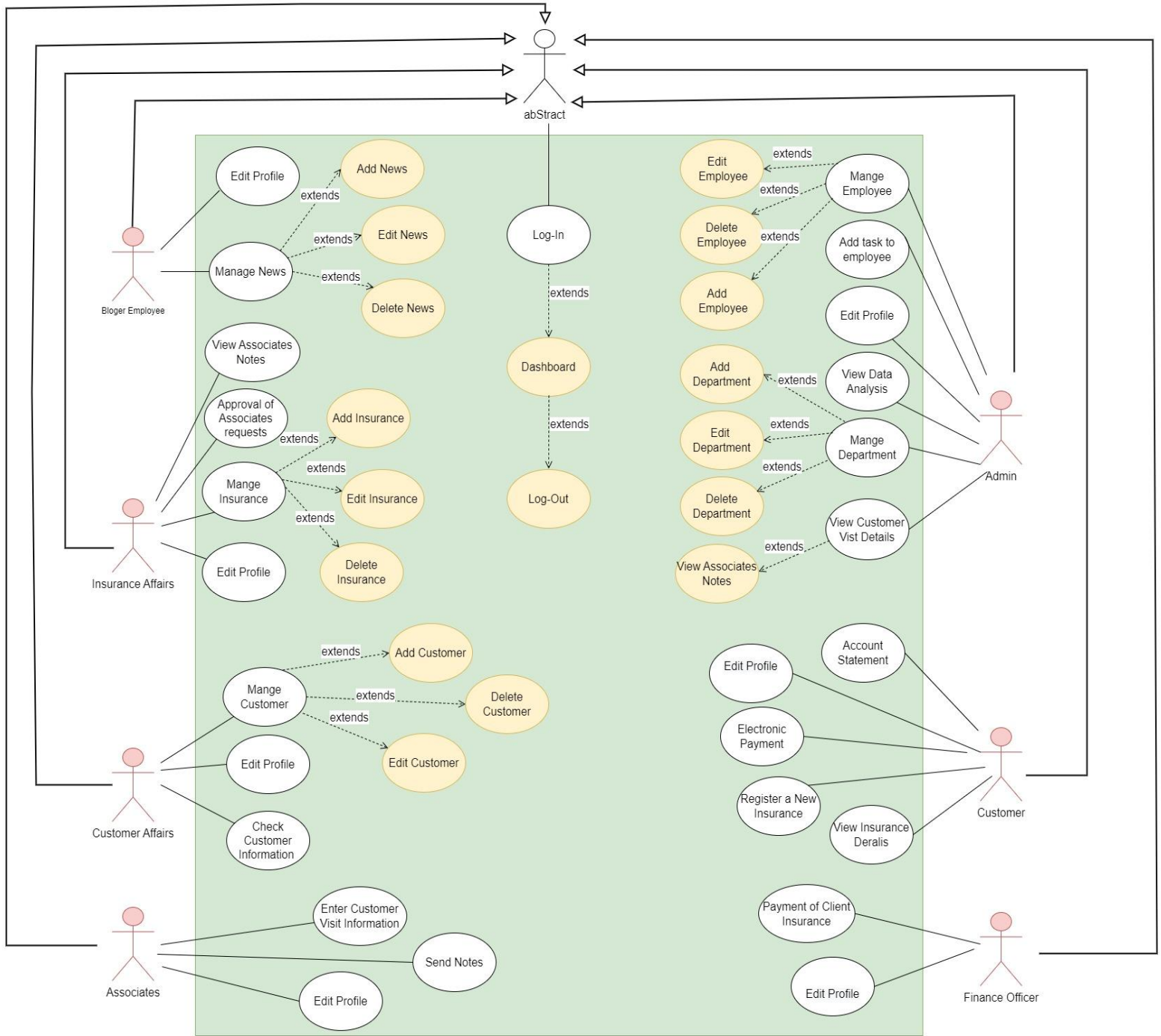


Figure 4 :Use-Case Diagram



3.5.2 Activity Diagram:

3.5.2.1 Activity Diagram for Check Customer Information :

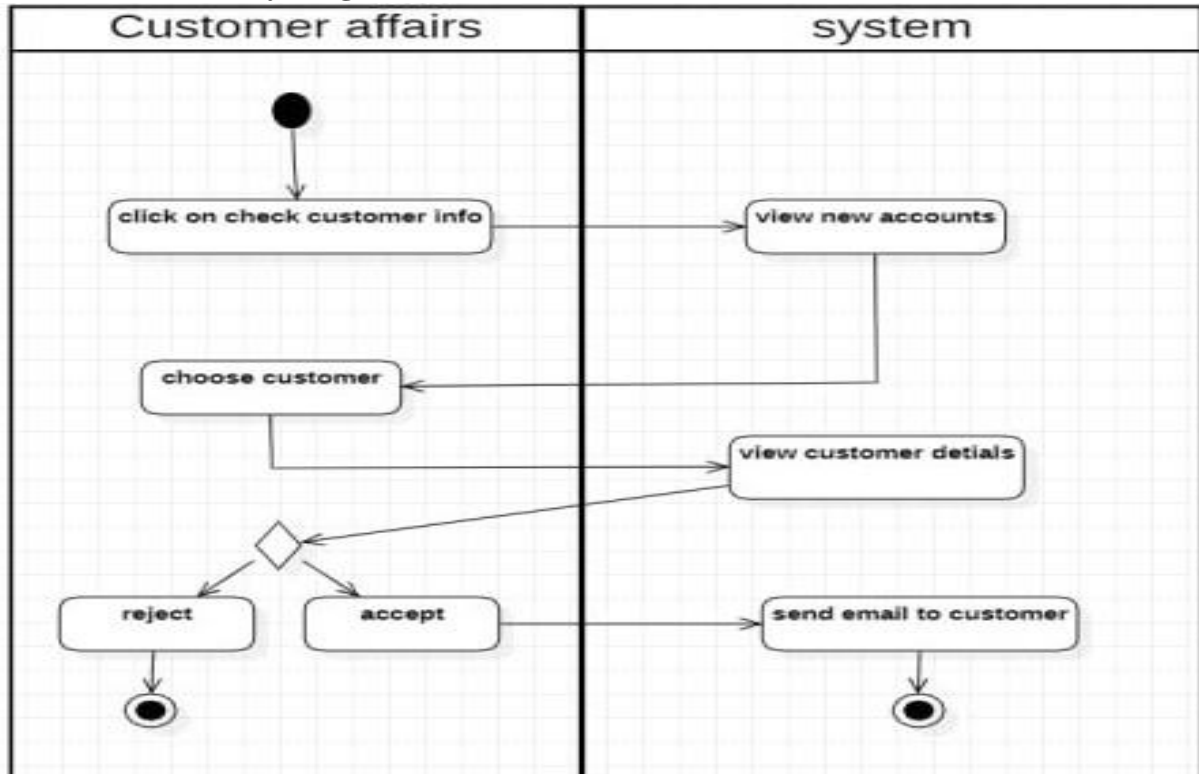


Figure (3.5): Activity-Diagram (Check Customer Information)

3.5.2.2 Activity Diagram for Delete News:

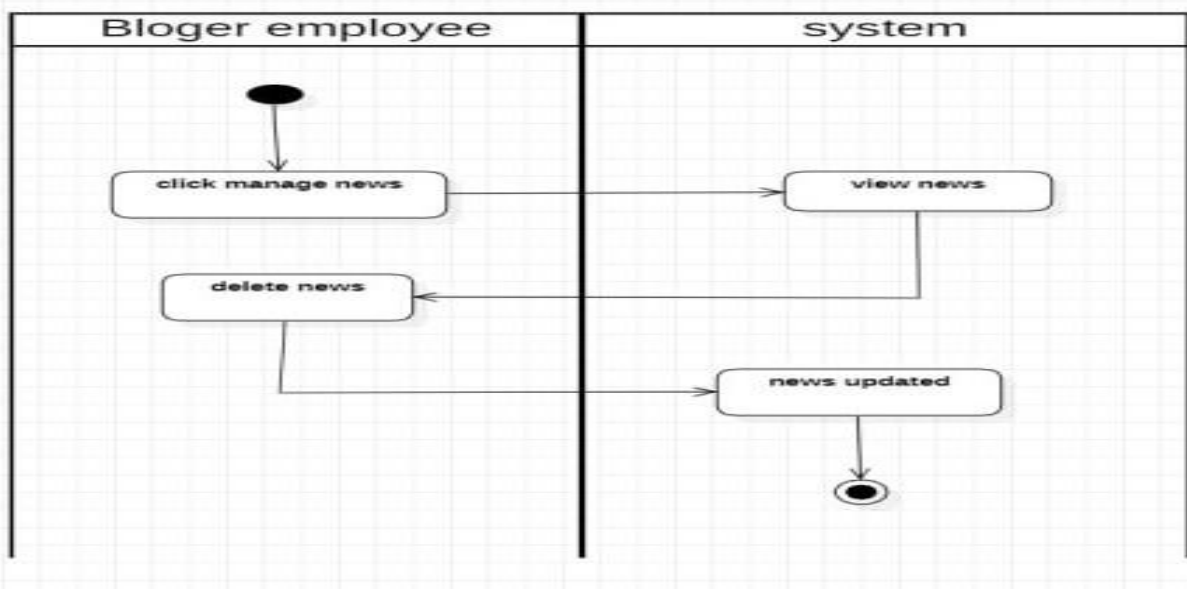


Figure (3.6): Activity-Diagram (Delete News)

3.5.2.3 Activity Diagram for Electronic Payment:

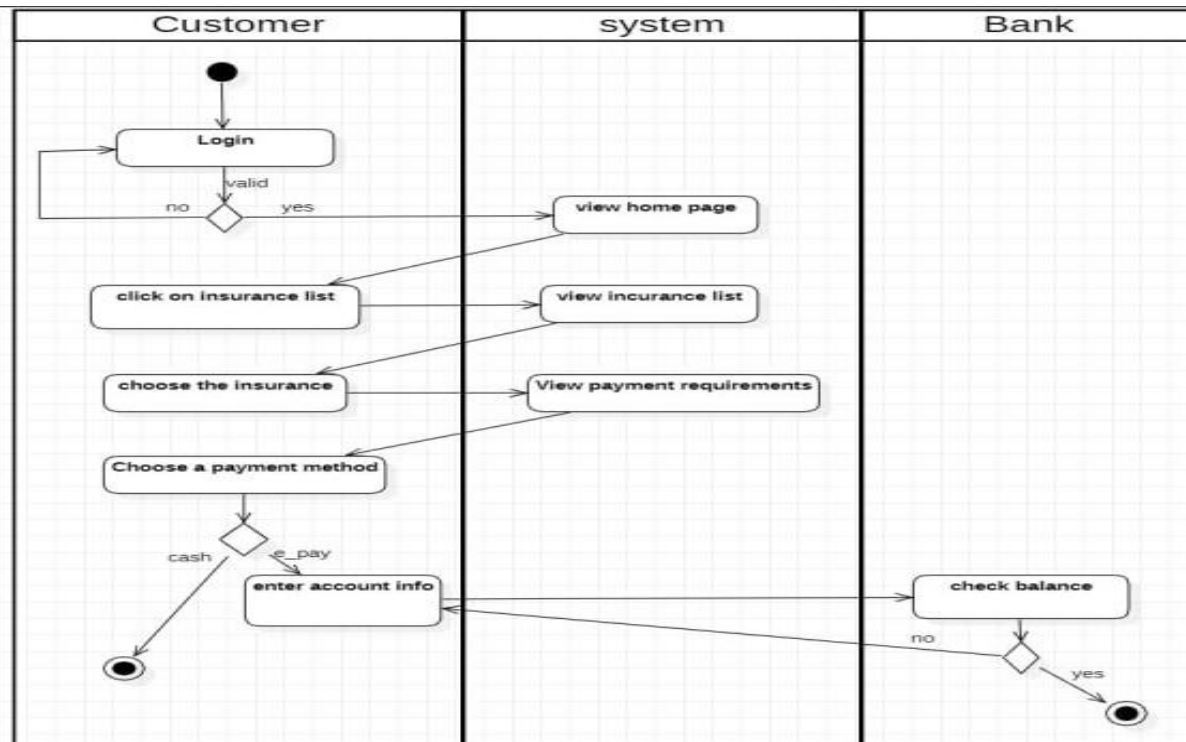


Figure (3.7): Activity-Diagram (Electronic Payment)

3.5.2.4 Activity Diagram for Log-In:

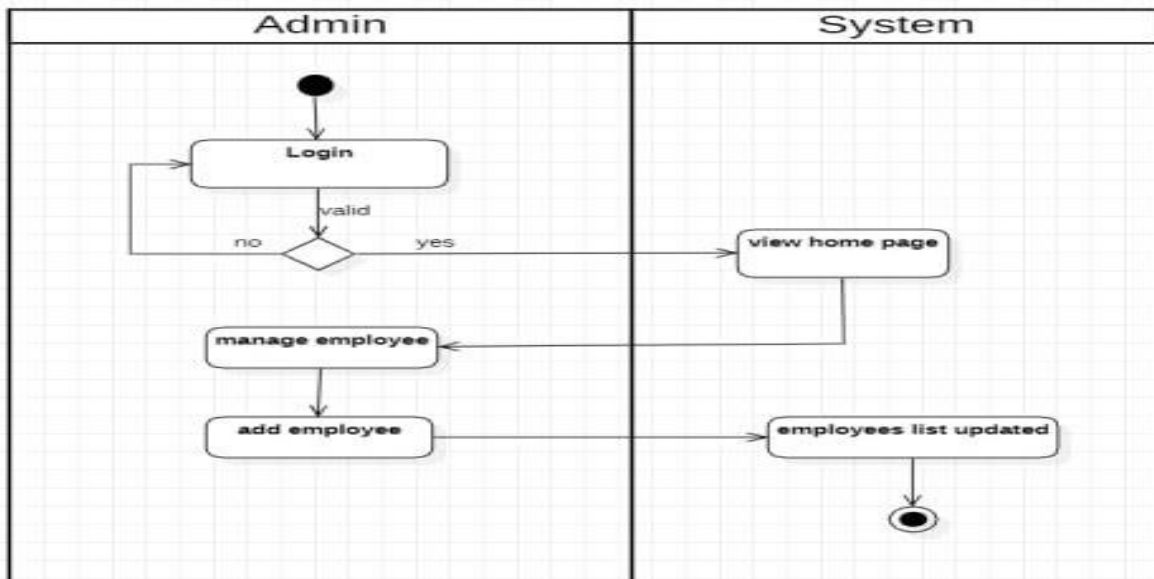
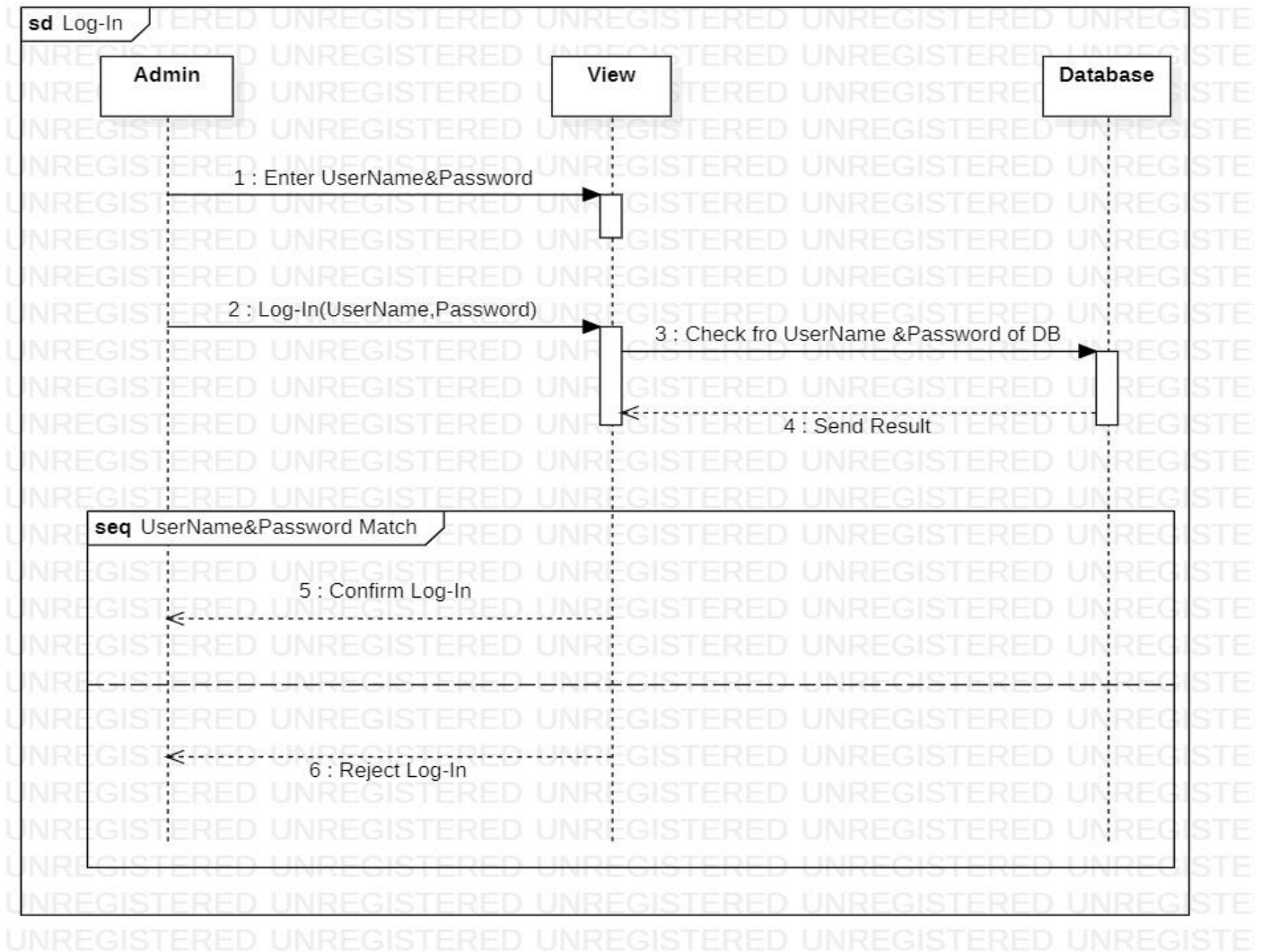


Figure (3.8): Activity-Diagram (Login Admin)

### 3.5.3 Sequence Diagram (Admin)(Log-In):



**Figure (3.9): Sequence-Diagram (Admin)(Login)**

3.5.4 Sequence Diagram (Customer)(Sign-Up):

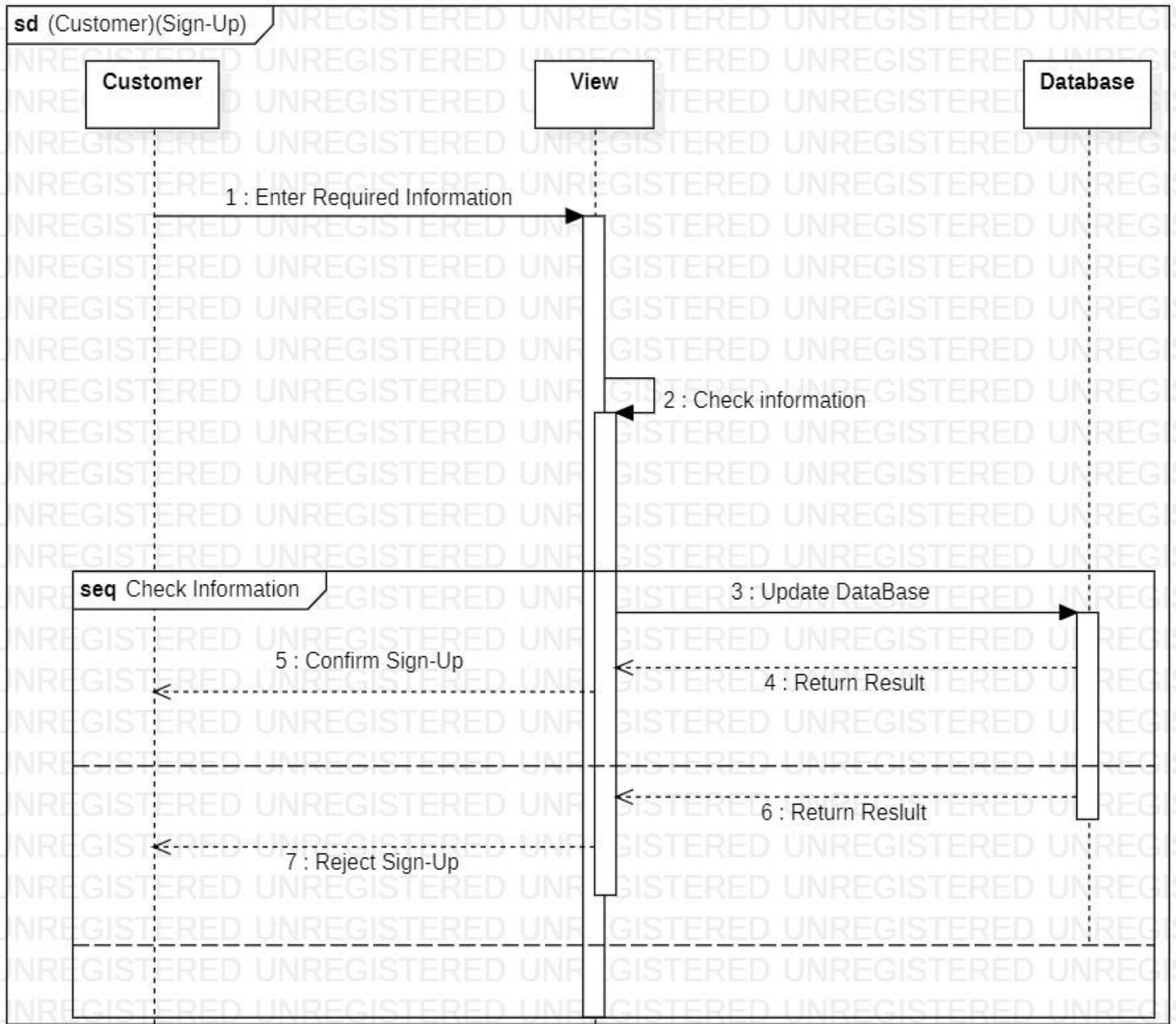
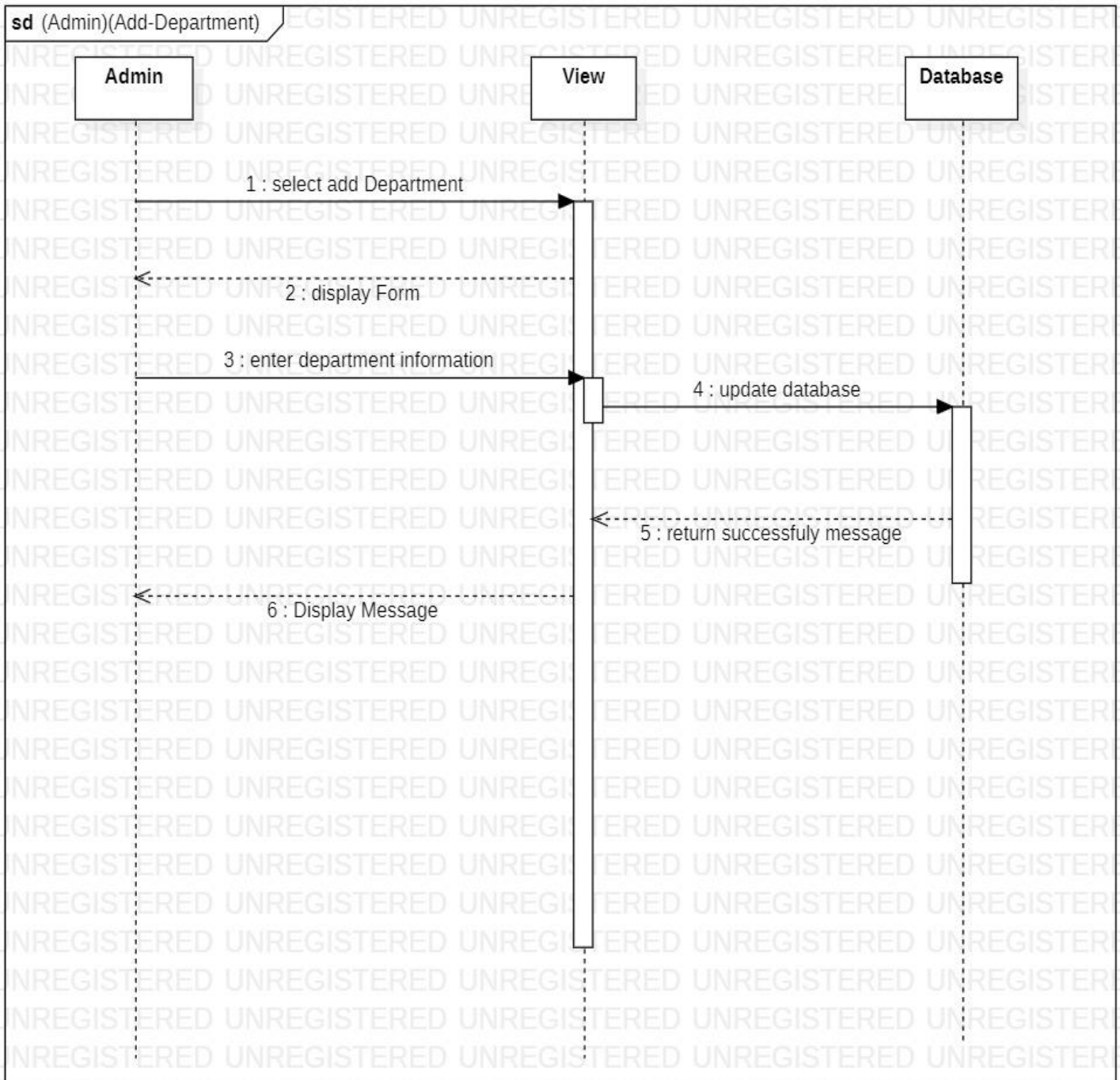


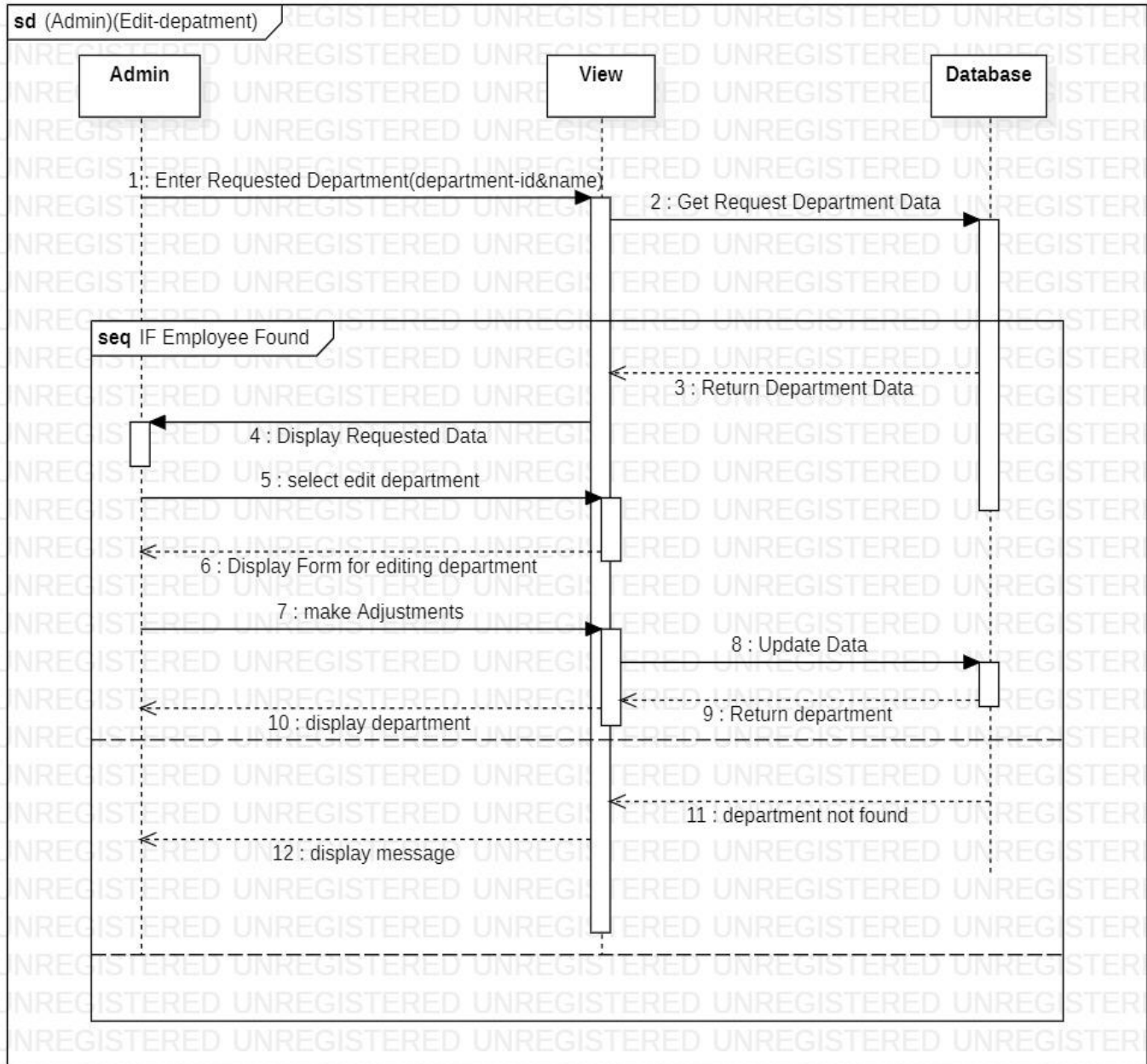
Figure (3.10): Sequence-Diagram (Customer)(Sign-Up)

### 3.5.5 Sequence Diagram (Admin)(Add-Department):



**Figure (3.11): Sequence-Diagram (Admin)(Add-Department)**

**3.5.6 Sequence Diagram (Admin)(Edit-Department):**



**Figure (3.12): Sequence-Diagram (Admin)(Edit-Department)**

### 3.5.7 Sequence Diagram (Admin)(Add-Employee):

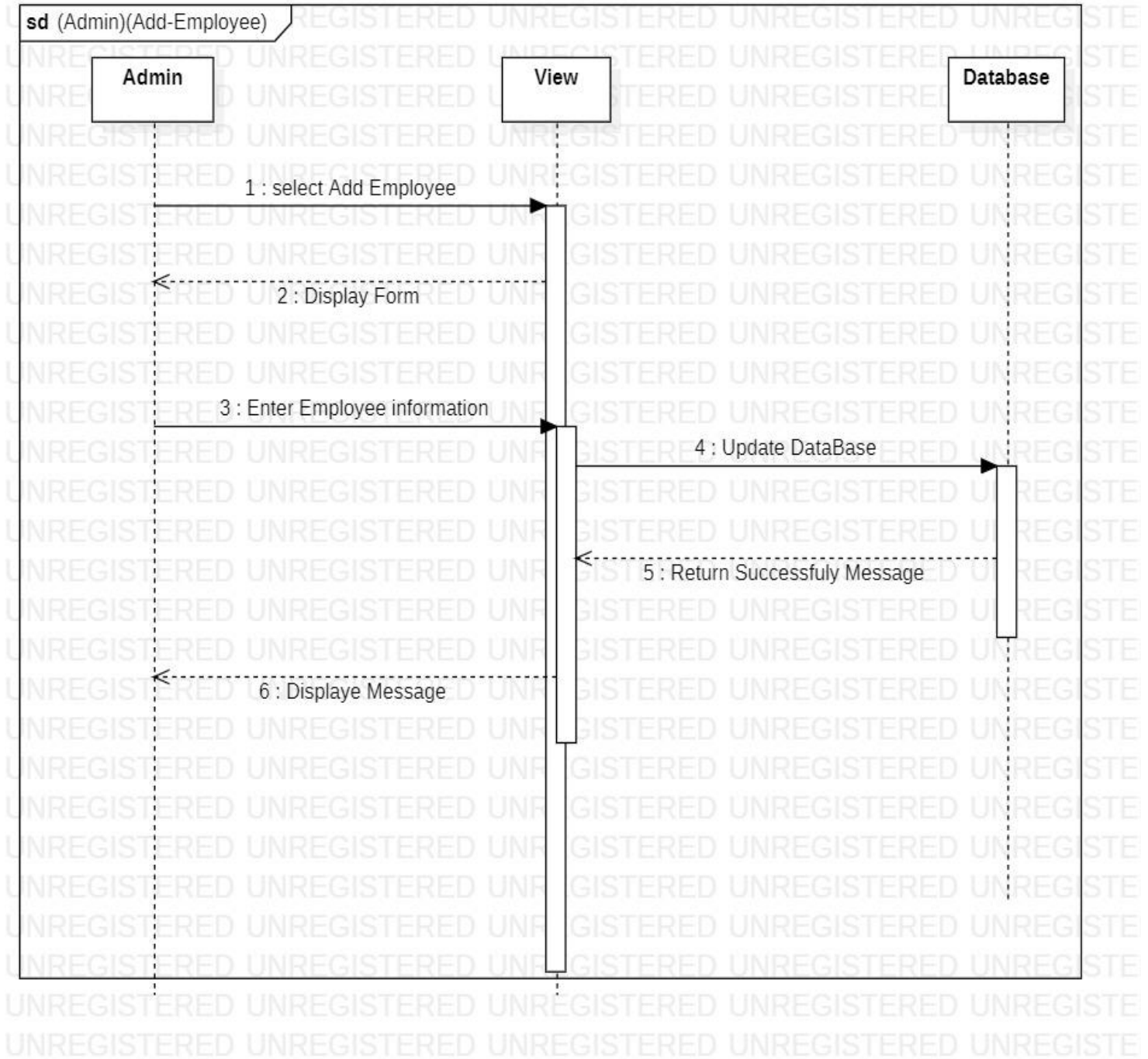
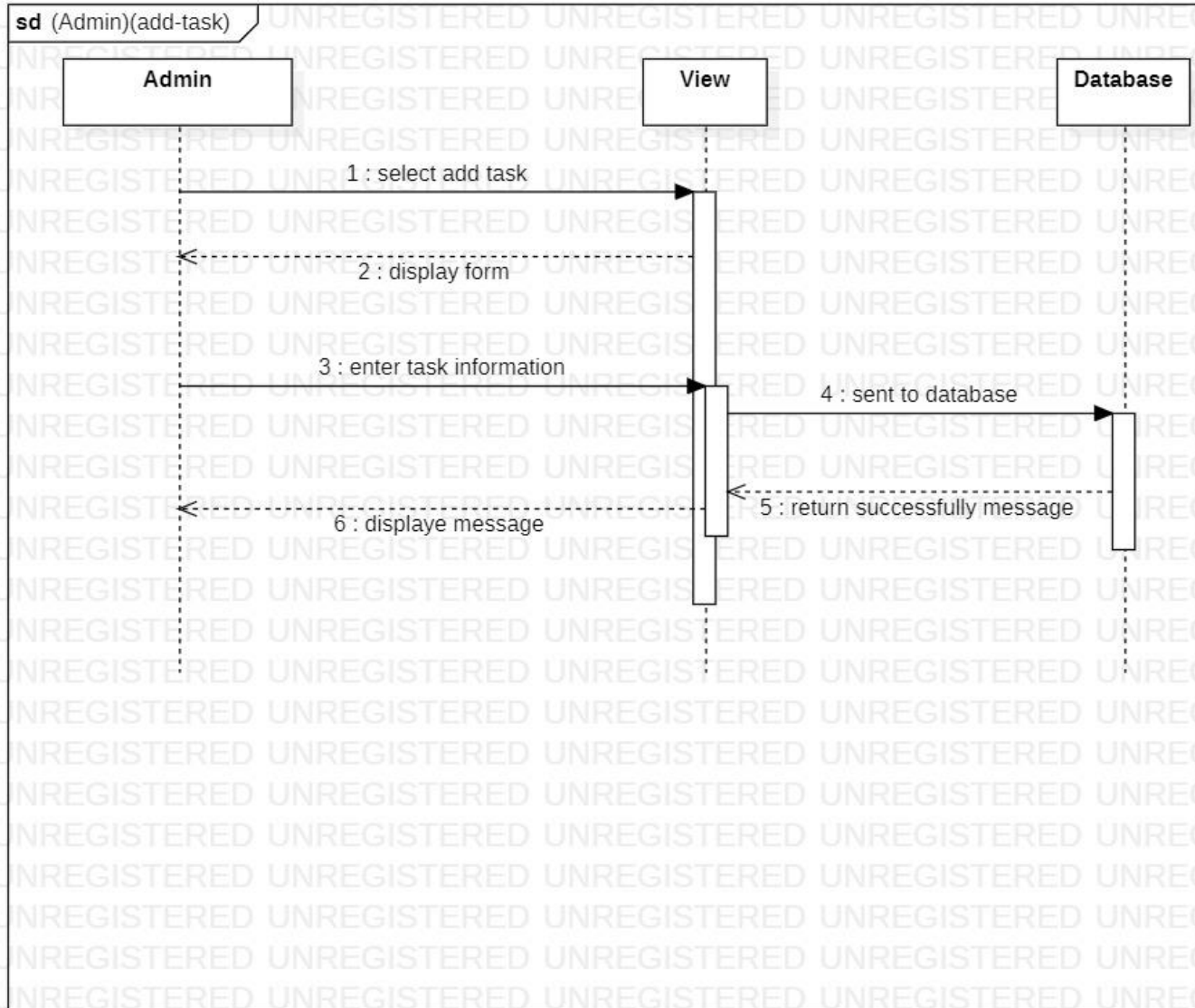


Figure (3.13): Sequence-Diagram (Admin)(Add-Employee)

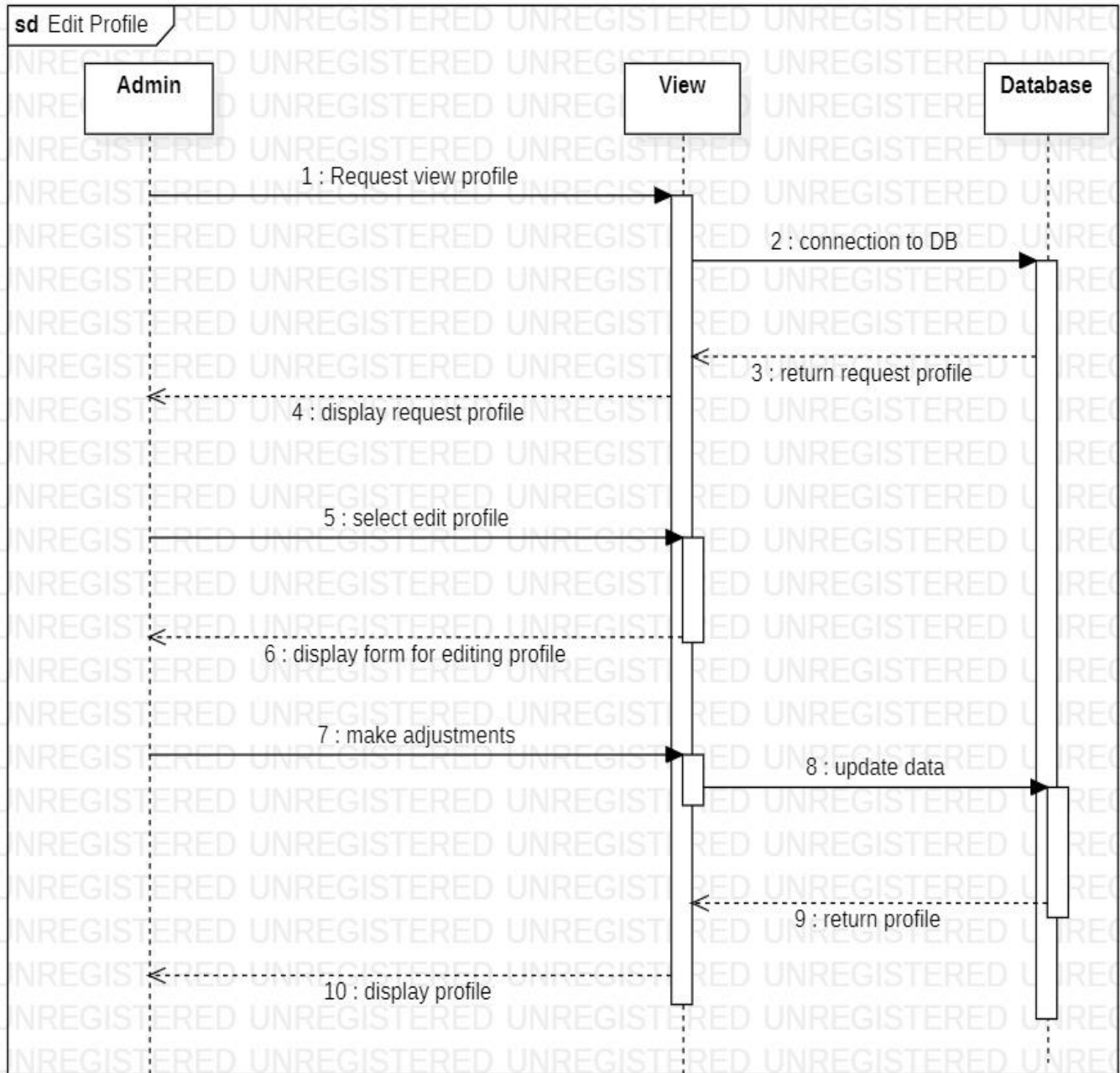
### 3.5.8 Sequence Diagram (Admin)(Add-Task):



**Figure (3.14): Sequence-Diagram (Admin)(Add-Task)**

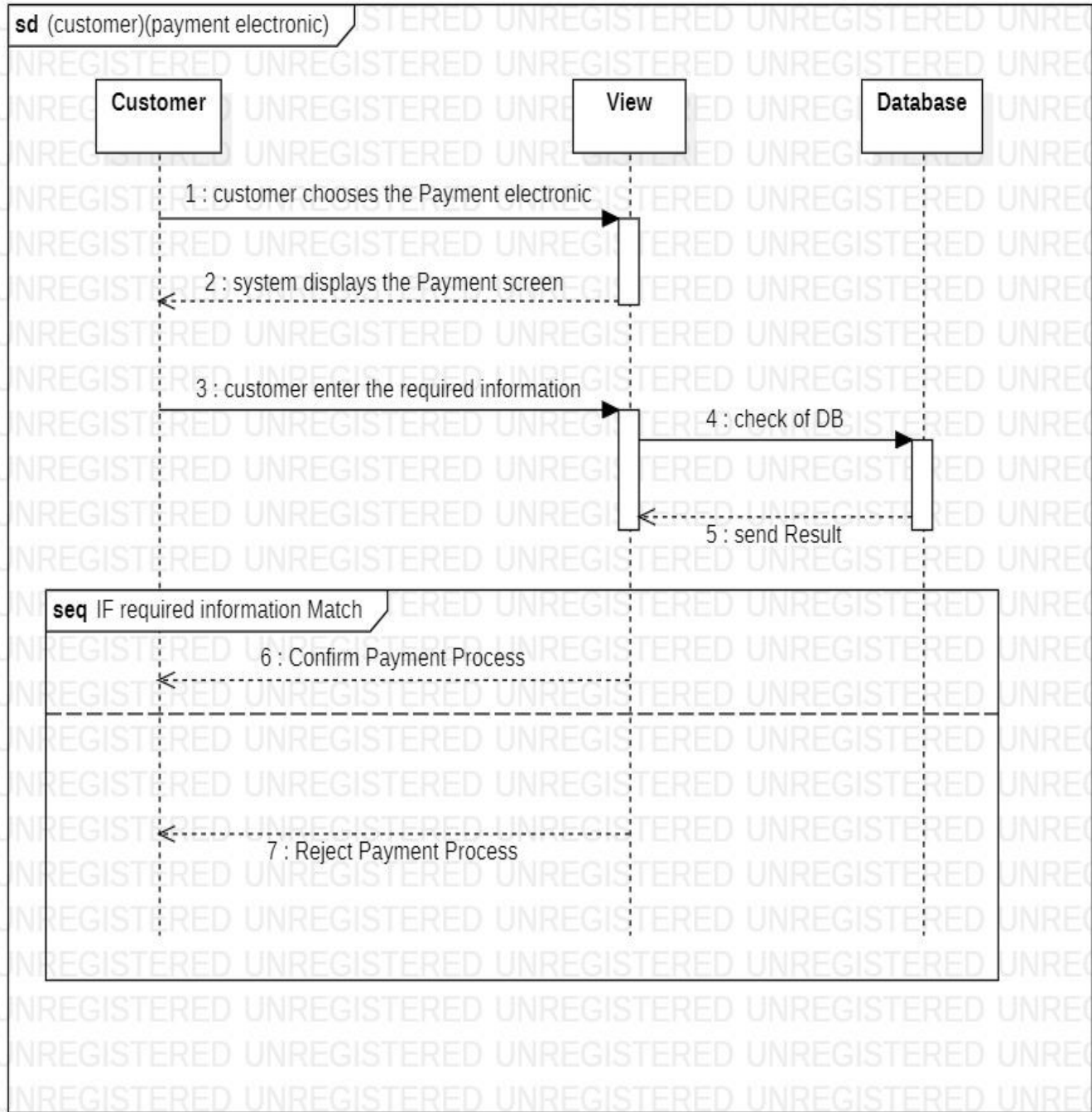


**3.5.9 Sequence Diagram (Admin)(Edit-Profile):**



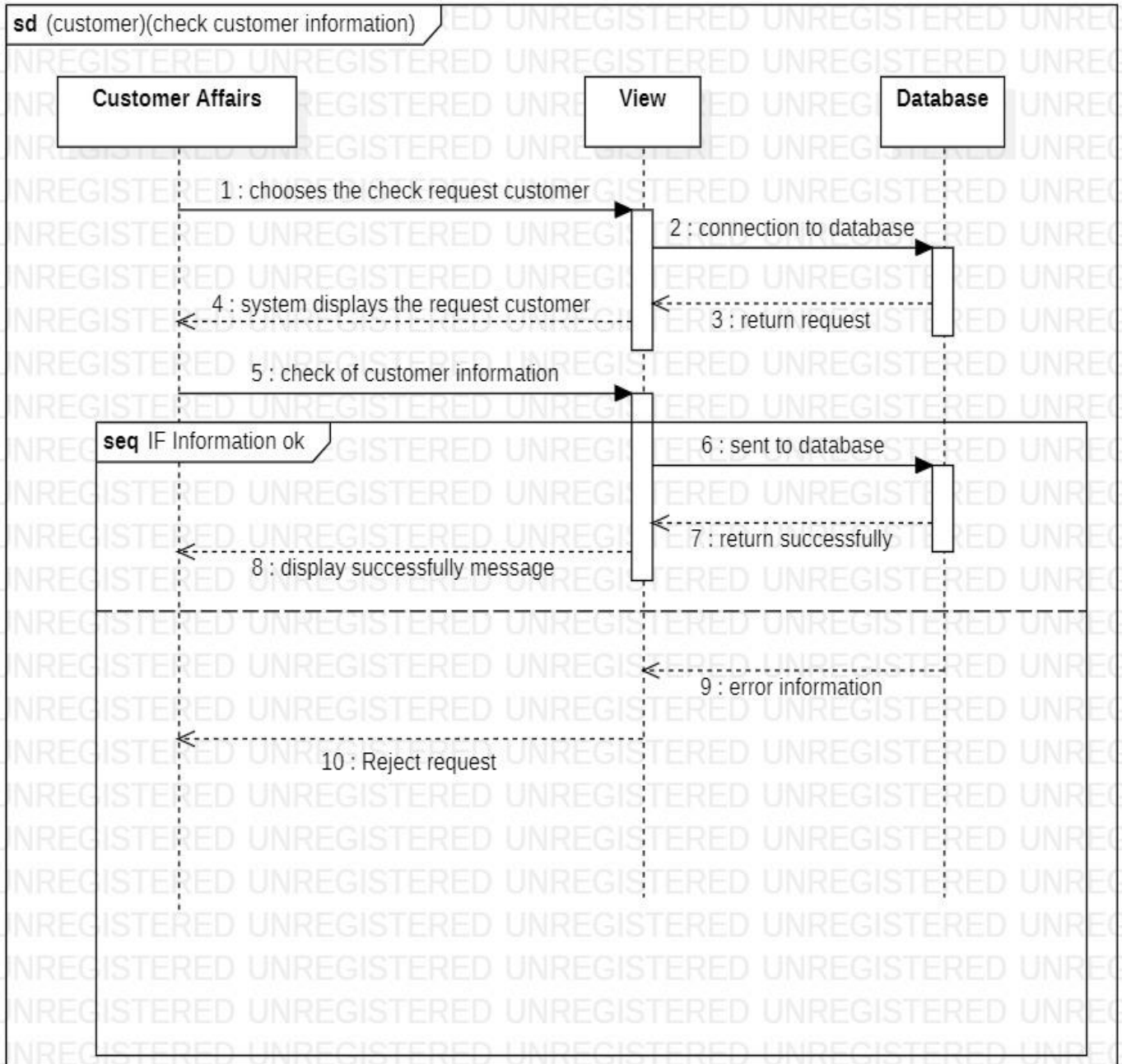
**Figure (3.15): Sequence-Diagram (Admin)(Edit-Profile)**

**3.5.10 Sequence Diagram (Customer)(Electronic-Payment):**



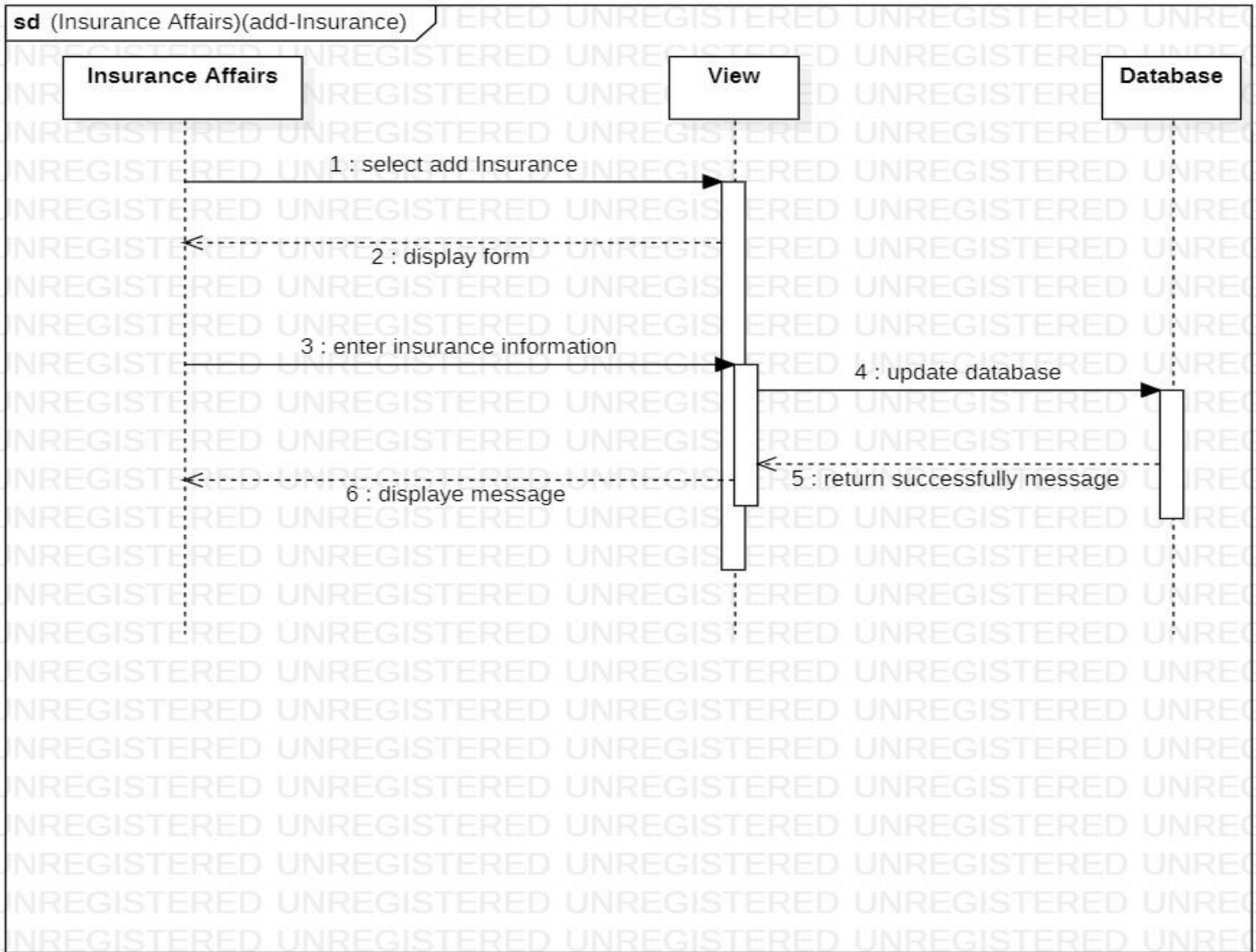
**Figure (3.16): Sequence-Diagram (Customer)(Payment-Electronic)**

**3.5.11 Sequence Diagram (Customer-Affairs)(Check-Information-Customer):**



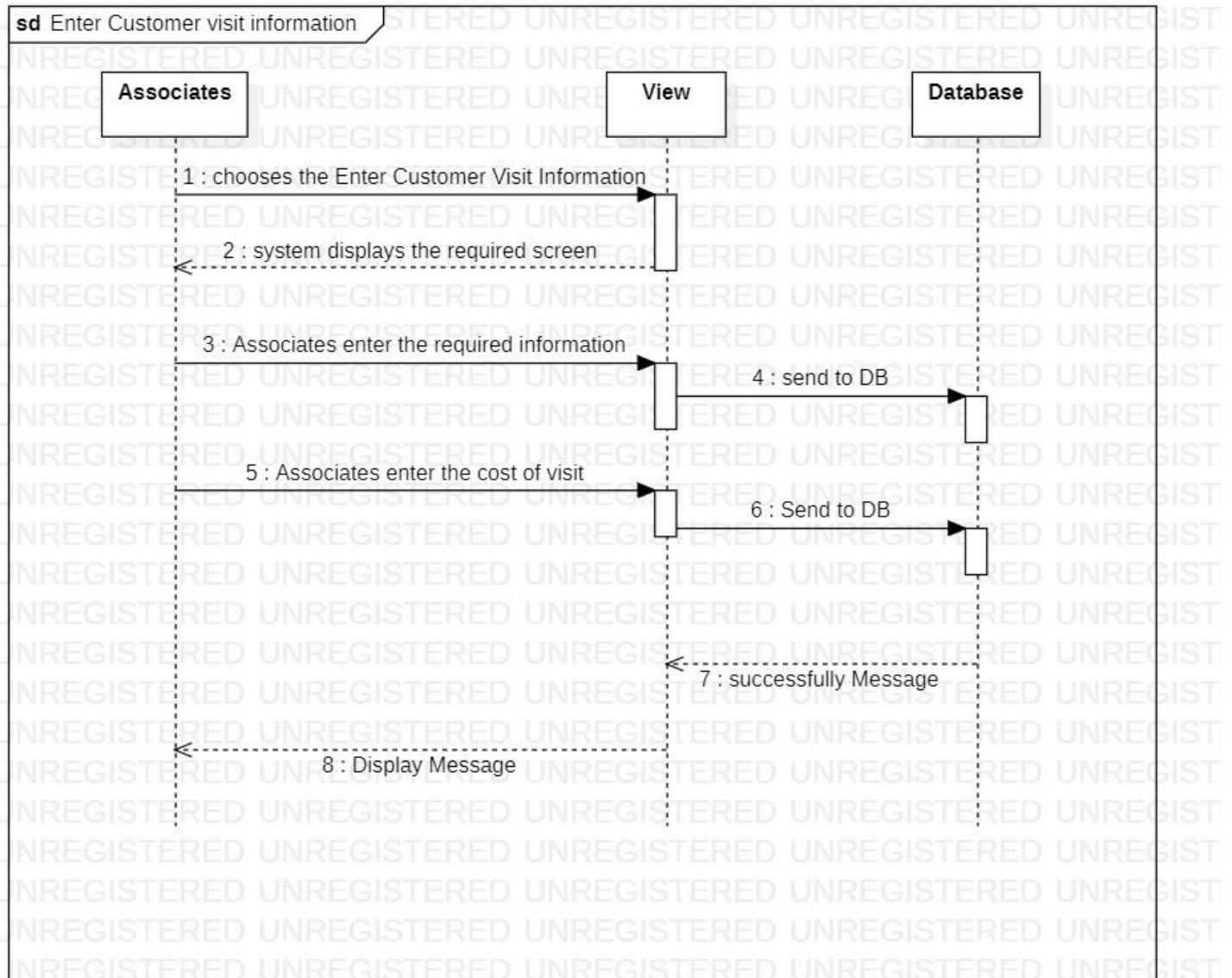
**Figure (3.17): Sequence-Diagram (Customer-Affairs)(Check Information Customer)**

**3.5.12 Sequence Diagram (Insurance-Affairs)(Add-Insurance):**



**Figure (3.18): Sequence Diagram (Insurance-Affairs)(Add-Insurance)**

### 3.5.13 Sequence Diagram (Associates)(Enter Customer Visit Information):



**Figure (3.19): Sequence Diagram (Associates)(Enter Customer Visit Information)**

**3.5.14 Use-Case Narrative :****3.5.14.1 Log-In****Table (3.2): Use-Case Narrative Log-In**

Use-Case Name:	Log-In abstract
Use-Case ID:	MSI-001
Short Description:	The Admin/Customer/Employee logs in with the username and password, and the system will display a message if he can log in or not.
Precondition:	The Guest must have an account.
Postcondition:	The username and password are correct.
Actors:	abstract.
Trigger:	The use case initiated when the customer/Admin/Employee selects this option from interface.
Standard Process:	1) The customer enters the data into the system. 2) The System will check the data. And the system displays a message to the customer that the login process is successful.
Alternative Processes:	Login Error: 2') The password is incorrect. 3') The customer is redirected to the step1.

**3.5.14.2 Payment of Client Insurance:****Table (3.3): Use-Case Narrative Payment of Client Insurance**

Use-Case Name:	Payment of client insurance .
Use-Case ID:	MSI-022
Short Description:	The customer will do the insurance and payment directly in the company by the finance officer employee.
Precondition:	Log in and choose customer insurance.
Postcondition:	The customer have a account in the system.
Actors:	Finance officer.

Trigger:	The use case initiated when the admin selects this option from interface
Standard Process:	<ol style="list-style-type: none"> <li>1) The finance office employee will look for the customer.</li> <li>2) Choose the insurance at the request of the client.</li> <li>3) The system displays the payment bill of this insurance.</li> <li>4) The finance office employee will confirm that the customer pay for him.</li> <li>5) The system added the pay to his account.</li> <li>6) The system displays the receipt of this pill.</li> <li>7) The finance office employee print the receipt and give it to the customer.</li> </ol>
Alternative Processes:	<p>The customer already have the insurance he is choose:</p> <ol style="list-style-type: none"> <li>3') The system shows a message that he had this insurance .</li> <li>4') The system returns the customer to step2.</li> </ol>

### 3.5.14.3 Add Employee :

**Table (3.4): Use-Case Narrative Add Employee**

Use-Case Name:	Mange Employee (Add Employee)
Use-Case ID:	MSI-002
Short Description:	Add new employee to the system this employee can be : Finance officer or blogger employee or insurance affairs or customer affairs.
Precondition:	Log in and choose Mange employee page.
Postcondition:	This information will be added in DB and display the employee table.
Actors:	Admin.
Trigger:	The use case initiated when the admin selects this option from interface
Standard Process:	<ol style="list-style-type: none"> <li>1) chooses add new employee.</li> <li>2) Enter the information of the new employee.</li> <li>3) Give the employee his username and password.</li> <li>4) Click Save.</li> <li>5) The system displays a message confirming the adding process.</li> </ol>
Alternative Processes:	<p>Wrong in input value :</p> <ol style="list-style-type: none"> <li>5') The system will ask the admin to re-enter the information correctly.</li> <li>6') The admin is redirected to the step1.</li> </ol>

**3.5.14.4 Delete Employee:****Table (3.5): Use-Case Narrative Delete Employee**

Use-Case Name:	Mange Employee (Delete Employee)
Use-Case ID:	MSI-003
Short Description:	Delete employee in the system.
Precondition:	Log in and choose Mange employee page.
Postcondition:	The employee will be deleted from DB .
Actors:	Admin.
Trigger:	The use case initiated when the admin selects this option from interface
Standard Process:	<ol style="list-style-type: none"> <li>1) Chooses delete employee.</li> <li>2) Confirm the delete operation.</li> <li>3) Click Yes.</li> <li>4) The system displays a message confirming the deleting process.</li> </ol>
Alternative Processes:	<p>Click No:</p> <ol style="list-style-type: none"> <li>4') The admin is redirected to the step1.</li> </ol>

**3.5.14.5 Edit Employee:****Table (3.6): Use-Case Narrative Edit Employee**

Use-Case Name:	Mange Employee (Edit Employee)
Use-Case ID:	MSI-004
Short Description:	Modify employee information in the system, such as place of residence or phone number or etc.
Precondition:	Log in and choose Mange employee page.
Postcondition:	This information will be modified in DB d display the employee table.
Actors:	Admin.
Trigger:	The use case initiated when the admin selects this option from interface



Standard Process:	<ol style="list-style-type: none"> <li>1) Chooses edit employee.</li> <li>2) Modify the information of the employee.</li> <li>3) Click Save.</li> <li>4) The system displays a message confirming the editing process.</li> </ol>
Alternative Processes:	<p>Wrong in input value:</p> <ol style="list-style-type: none"> <li>4') The system will ask the admin to re-enter the information correctly.</li> <li>5') The admin is redirected to the step2.</li> </ol>

### 3.5.14.6 Add Department:

**Table (3.7): Use-Case Narrative Add Department**

Use-Case Name:	Mange department (Add department )
Use-Case ID:	MSI-005
Short Description:	Add new department to the system.
Precondition:	Log in and choose Mange department page.
Postcondition:	This information will be added in DB and display the department table.
Actors:	Admin.
Trigger:	The use case initiated when the admin selects this option from interface
Standard Process:	<ol style="list-style-type: none"> <li>1) chooses add new department .</li> <li>2) Enter the information of the new department .</li> <li>3) Click Save.</li> <li>4) The system displays a message confirming the adding process.</li> </ol>
Alternative Processes:	<p>Wrong in input value :</p> <ol style="list-style-type: none"> <li>4') The system will ask the admin to re-enter the information correctly.</li> <li>5') The admin is redirected to the step2.</li> </ol>

**3.5.14.7 Edit Department:****Table (3.8): Use-Case Narrative Edit Department**

Use-Case Name:	Mange Department (Edit Department )
Use-Case ID:	MSI-006
Short Description:	Modify Department information in the system, such as Code of the department or etc.
Precondition:	Log in and choose Mange department page.
Postcondition:	This information will be modified in DB display the department table.
Actors:	Admin.
Trigger:	The use case initiated when the admin selects this option from interface
Standard Process:	<ol style="list-style-type: none"> <li>1) Chooses edit department .</li> <li>2) Modify the information of the department .</li> <li>3) Click Save.</li> <li>4) The system displays a message confirming the editing process.</li> </ol>
Alternative Processes:	<p>Wrong in input value:</p> <ol style="list-style-type: none"> <li>4') The system will ask the admin to re-enter the information correctly.</li> <li>5') The admin is redirected to the step2.</li> </ol>

**3.5.14.8 Delete Department:****Table (3.9): Use-Case Narrative Delete Department**

Use-Case Name:	Mange Department (Edit Department )
Use-Case ID:	MSI-006
Short Description:	Modify Department information in the system, such as Code of the department or etc.
Precondition:	Log in and choose Mange department page.
Postcondition:	This information will be modified in DB display the department table.
Actors:	Admin.

Trigger:	The use case initiated when the admin selects this option from interface
Standard Process:	5) Chooses edit department . 6) Modify the information of the department . 7) Click Save. 8) The system displays a message confirming the editing process.
Alternative Processes:	Wrong in input value: 4') The system will ask the admin to re-enter the information correctly. 5') The admin is redirected to the step2.

### 3.5.14.9 Admin-Profile & Edit Profile:

**Table (3.10): Use-Case Narrative Edit Profile**

Use-Case Name:	Admin-Profile & Edit profile
Use-Case ID:	MSI-008
Short Description:	Change the admin password
Precondition:	Log in and choose my profile page.
Postcondition:	The password will be updated in DB .
Actors:	Admin.
Trigger:	The use case initiated when the admin selects this option from interface
Standard Process:	1) Chooses change password. 2) Enter the old password. 3) Enter the new password. 4) Enter the confirm of the new password. 5) Click save. 6) The system displays a message confirming the changing process.
Alternative Processes:	The old password is wrong: 6') The system display message the old password is wrong. 7') The admin is redirected to the step2.  Wrong with the new password value: 6') The system display message the new password is too short or doesn't have a number with small and big letter. 7') The admin is redirected to the step3.  The new password doesn't match with the confirm : 6') The system display message the new password is new password doesn't match with the confirm of password. 7') The admin is redirected to the step4.

**3.5.14.10 Account Statement:****Table (3.11): Use-Case Narrative Account Statement**

Use-Case Name:	Account Statement
Use-Case ID:	MSI-009
Short Description:	The customer can check his financial account with the company, as it shows the payments he made with their details and the amount of his remaining financial account
Precondition:	Log in and choose Account Statement page.
Postcondition:	View all transactions applied to the account on the screen.
Actors:	Customer.
Trigger:	The use case initiated when the customer selects this option from interface
Standard Process:	1) Choosing an Account Statement service. 2) A summary of the client's latest activities appears on the screen.
Alternative Processes:	The customer's unwillingness to disclose his account: 2') Customer Choice Back to Home Page. 3') The system displays the home page of the client.

**3.5.14.11 Register a New Insurance:****Table (3.12): Use-Case Narrative Register a New Insurance**

Use-Case Name:	Register a new insurance
Use-Case ID:	MSI-010
Short Description:	The customer will add new insurance in his account to use it .
Precondition:	Log in and choose new insurance.
Postcondition:	You have to do the electric payment.
Actors:	Customer.
Trigger:	The use case initiated when the customer selects this option from interface
Standard Process:	1) The system displays all insurance in the company with all details . 2) The customer will choose one of them. 3) The system displays the electric payment of this insurance .
Alternative Processes:	The customer already has the insurance he is choose: 5') The system shows a message that he had this insurance . 6') The system returns the customer to step2.

**3.5.14.12 Electronic Payment:****Table (3.13): Use-Case Narrative Electronic Payment**

Use-Case Name:	Electronic Payment
Use-Case ID:	MSI-011
Short Description:	The customer pays the insurance bill through his bank account.
Precondition:	Choose an insurance that want to pay for it.
Postcondition:	The insurance will be added to the customer insurances.
Actors:	Customer.
Trigger:	The use case initiated when the customer selects this option from interface
Standard Process:	<ol style="list-style-type: none"> <li>1) The system displays the Pay screen.</li> <li>2) The customer chooses the bank he works with.</li> <li>3) The customer enters his card number.</li> <li>4) The customer confirms the payment.</li> <li>5) The system displays a message confirming the payment process.</li> </ol>
Alternative Processes:	<p>Wrong card number:</p> <ol style="list-style-type: none"> <li>5') The system shows a message that the card number is wrong.</li> <li>6') The system returns the customer to step3.</li> </ol> <p>There is not enough money to complete the payment:</p> <ol style="list-style-type: none"> <li>5') The system shows a message that the card can't afford this process.</li> <li>6') The system returns the customer to step2.</li> </ol>

**3.5.14.13 Customer-Profile & Edit Profile:****Table (3.14): Use-Case Narrative Edit Profile**

Use-Case Name:	Electronic Payment
Use-Case ID:	MSI-011
Short Description:	The customer pays the insurance bill through his bank account.
Precondition:	Choose an insurance that want to pay for it.
Postcondition:	The insurance will be added to the customer insurances.
Actors:	Customer.

Trigger:	The use case initiated when the customer selects this option from interface
Standard Process:	6) The system displays the Pay screen. 7) The customer chooses the bank he works with. 8) The customer enters his card number. 9) The customer confirms the payment. 10) The system displays a message confirming the payment process.
Alternative Processes:	Wrong card number: 5') The system shows a message that the card number is wrong. 6') The system returns the customer to step3.  There is not enough money to complete the payment: 5') The system shows a message that the card can't afford this process. 6') The system returns the customer to step2.

**3.5.14.14 View Insurance Details:****Table (3.15): Use-Case Narrative View Insurance Details**

Use-Case Name:	View Insurance Details
Use-Case ID:	MSI-013
Short Description:	View the insurance of data that the client has like covers insurance.
Precondition:	Log in and choose View Insurance Details page.
Postcondition:	View all insurance details applied on the screen.
Actors:	Customer.
Trigger:	The use case initiated when the customer selects this option from interface
Standard Process:	1) Choosing view Insurance details. 2) Choose one of the insurance . 3) Appears on the screen the insurance details.
Alternative Processes:	Customer don't have insurance : 2') Appears on the screen that customer don't have insurances

**3.5.14.15 Delete Blog:****Table (3.16): Use-Case Narrative Delete Blog**

Use-Case Name:	Mange News (Remove news)
Use-Case ID:	MSI-027
Short Description:	Remove news from blogs page.
Precondition:	Log in and choose Mange news page.
Postcondition:	The blog will be removed from DB .
Actors:	Blogger Employee.
Trigger:	The use case initiated when the admin selects this option from interface.
Standard Process:	<ol style="list-style-type: none"> <li>1) Chooses remove blog .</li> <li>2) Confirm the remove operation.</li> <li>3) Click Yes.</li> <li>4) The system displays a message confirming the removing process.</li> </ol>
Alternative Processes:	Click No: 4') The admin is redirected to the step1.

**3.5.14.16 Associates-Enter Customer Visit Information:****Table (3.17): Use-Case Narrative Associates-Enter Customer Visit Information**

Use-Case Name:	<b>Enter Customer Visit Information</b>
Use-Case ID:	MSI-012
Short Description:	The Associates Enter the Visit Information and Send to the Company.
Precondition:	Associates is login in the System .
Postcondition:	The Associates Enter the Visit Information and Send the Information to the Company.
Actors:	Associates.
Trigger:	The use case initiated when the Associates selects this option from interface

Standard Process:	11) The system displays the information entry screen. 12) The Associates enter the required information . 13) The Associates enter the cost of visit. 14) The Associates confirms the forms. 15) The system displays a message confirming the process.
-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 3.5.14.17 Check Customer Information:

**Table (3.18): Use-Case Narrative Check Customer Information**

Use-Case Name:	Check customer information
Use-Case ID:	MSI-016
Short Description:	The customer affairs employee will verify the information of the new customers who registered for the first time, he will verify the photo ID entered and other information, and on the basis of this, approval or rejection will be made.
Precondition:	Log in and choose check customer information.
Postcondition:	Added new customer to the system.
Actors:	Customer affairs.
Trigger:	The use case initiated when the admin selects this option from interface
Standard Process:	<ol style="list-style-type: none"> <li>1) The system displays all the new request .</li> <li>2) The customer affairs employee will choose one of them.</li> <li>3) The system displays the ID photo and information of this request.</li> <li>4) The customer affairs employee make sure that the information is correct and call the applicant of this request.</li> <li>5) The customer affairs employee approved the request.</li> <li>6) The system sent an email to applicant of the request , tell him that his request been approved and he can sign in.</li> <li>7) The system will added to the customer table.</li> <li>8) The system displays a message confirming the process.</li> <li>9) The customer affairs employee call the applicant that his been approved.</li> </ol>
Alternative Processes:	The customer affairs employee rejected the request : 5') The system will delete the request. 6') The system returns the customer affairs to step2.



**3.5.14.18 Delete Insurance:****Table (3.19): Use-Case Narrative Delete Insurance**

Use-Case Name:	Mange insurance (Remove insurance )
Use-Case ID:	MSI-020
Short Description:	Remove insurance in the system.
Precondition:	Log in and choose Mange insurance page.
Postcondition:	The insurance will be removed from DB .
Actors:	insurance affairs.
Trigger:	The use case initiated when the insurance affairs selects this option from interface
Standard Process:	<ol style="list-style-type: none"> <li>1) Chooses remove insurance .</li> <li>2) Confirm the remove operation.</li> <li>3) Click Yes.</li> <li>4) The system displays a message confirming the removing process.</li> </ol>
Alternative Processes:	<p>Click No:</p> <ol style="list-style-type: none"> <li>4') The insurance affairs is redirected to the step1.</li> </ol>

**3.5.14.19 Payment of Client Insurance:****Table (3.20): Use-Case Narrative Payment of Client Insurance**

Use-Case Name:	Payment of client insurance .
Use-Case ID:	MSI-022
Short Description:	The customer will do the insurance and payment directly in the company by the finance officer employee.
Precondition:	Log in and choose customer insurance.
Postcondition:	The customer have a account in the system.
Actors:	Finance officer.
Trigger:	The use case initiated when the admin selects this option from interface
Standard Process:	<ol style="list-style-type: none"> <li>1) The finance office employee will look for the customer.</li> <li>2) Choose the insurance at the request of the client.</li> <li>3) The system displays the payment bill of this insurance.</li> <li>4) The finance office employee will confirm that the customer pay for him.</li> </ol>

	<ul style="list-style-type: none"> <li>5) The system added the pay to his account.</li> <li>6) The system displays the receipt of this pill.</li> <li>7) The finance office employee print the receipt and give it to the customer.</li> </ul>
Alternative Processes:	<p>The customer already have the insurance he is choose:</p> <ul style="list-style-type: none"> <li>3') The system shows a message that he had this insurance .</li> <li>4') The system returns the customer to step2.</li> </ul>

### 3.5.14.20 Add Customer:

Table (3.21): Use-Case Narrative Add Customer

Use-Case Name:	Mange Customer (Add Customer )
Use-Case ID:	MSI-013
Short Description:	Add new customer to the system.
Precondition:	Log in and choose Mange customer page.
Postcondition:	This information will be added in DB and display the customer table.
Actors:	Customer affairs.
Trigger:	The use case initiated when the customer affairs selects this option from interface
Standard Process:	<ul style="list-style-type: none"> <li>1) chooses add new customer .</li> <li>2) Enter the information of the new customer .</li> <li>3) Give the customer his username and password.</li> <li>4) Click Save.</li> <li>5) The system displays a message confirming the adding process.</li> </ul>
Alternative Processes:	<p>Wrong in input value :</p> <ul style="list-style-type: none"> <li>5') The system will ask the customer affairs to re-enter the information correctly.</li> <li>6') The customer affairs is redirected to the step1.</li> </ul>

### 3.6 Entity Relationship Diagram (ERD) :

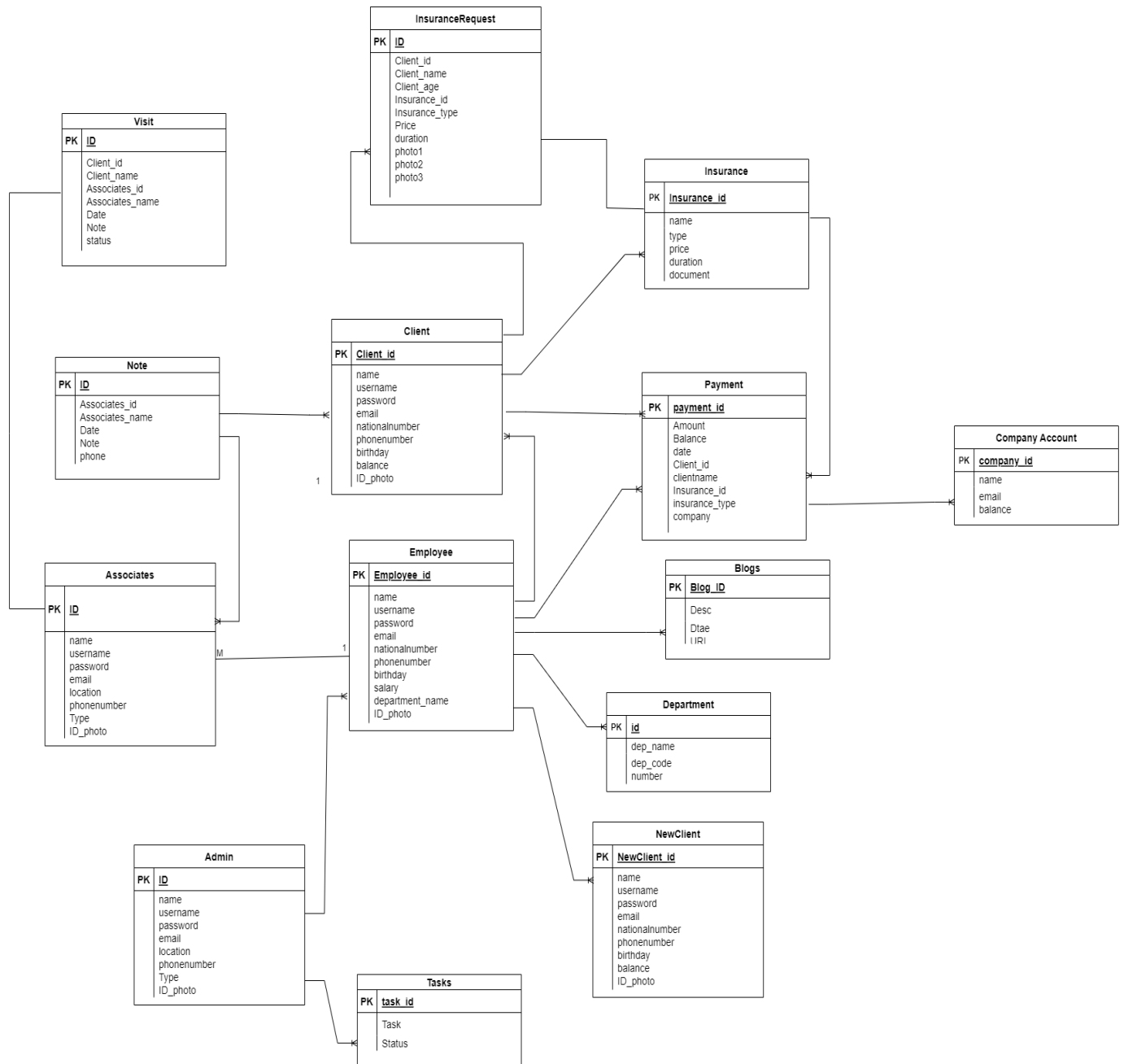


Figure (3.20): Entity Relationship Diagram

### 3.7 Data Base Table :

#### 3.7.1 Table Structure for admin :

**Table (3.22): Database Tables(Admin Table)**

Column	Data Type	Index	Description
<b>Id</b>	<b>Int</b>	<b>Primary key</b>	<b>Admin ID</b>
<b>Username</b>	<b>Text</b>	<b>Not null</b>	<b>Admin Username</b>
<b>Password</b>	<b>Text</b>	<b>Not null</b>	<b>Password for Admin</b>
<b>Name</b>	<b>Varchar</b>	<b>Not null</b>	<b>Admin Name</b>
<b>Phone Number</b>	<b>Varchar</b>	<b>Not null</b>	<b>Admin Phone</b>
<b>Address</b>	<b>Varchar</b>	<b>Not null</b>	<b>Admin Address</b>
<b>Email</b>	<b>Varchar</b>	<b>Not null</b>	<b>Admin Email</b>

#### 3.7.2 Table Structure for Employee :

**Table (3.23): DataBase Tables(Employee Table)**

Column	Data Type	Index	Description
<b>Id</b>	<b>Int</b>	<b>Primary key</b>	<b>Employee ID</b>
<b>Username</b>	<b>Text</b>	<b>Not null</b>	<b>Employee Username</b>
<b>Password</b>	<b>Text</b>	<b>Not null</b>	<b>Password of Employee</b>
<b>Name</b>	<b>Varchar</b>	<b>Not null</b>	<b>Employee Name</b>
<b>birthday</b>	<b>date</b>	<b>Not null</b>	<b>Employee Birthday</b>
<b>Email</b>	<b>Varchar</b>	<b>Not null</b>	<b>Employee Email</b>
<b>Department-Name</b>	<b>Varchar</b>	<b>Not null</b>	<b>Employee Department-Name</b>
<b>Salary</b>	<b>Int</b>	<b>Not null</b>	<b>Employee Salary</b>
<b>ID_Image</b>	<b>VARBINARY</b>	<b>Not null—</b>	<b>Employee ID-Image</b>

#### 3.7.3 Table Structure for Tasks:

**Table (3.24): DataBase Tables(Task Table)**

Column	Data Type	Index	Description
<b>Id</b>	<b>Int</b>	<b>Primary Key</b>	<b>Task ID</b>
<b>Description</b>	<b>Text</b>	<b>Not Null</b>	<b>Task Description</b>
<b>Status</b>	<b>Boolean</b>	<b>Not Null</b>	<b>Task Status</b>

## 3.7.4 Table Structure for Customer :

Table (3.25): DataBase Tables(Customer Table)

Column	Data Type	Index	Description
Id	Int	Primary key	Customer ID
Username	Text	Not null	Customer Username
Password	Text	Not null	Customer Password
Name	Varchar	Not null	Customer Name
birthday	date	Not null	Customer Birthday
Email	Varchar	Not null	Customer Email
Phone Number	Int	Not null	Customer Phone
National Number	Int	Not null	Customer National Number

## 3.7.5 Table Structure for Doctors:

Table (3.26): DataBase Tables(New Customer Check Table)

Column	Data Type	Index	Description
Id	Int	Primary key	New Customer ID
Username	Text	Not null	New Customer Username
Password	Text	Not null	New Customer Password
Name	Varchar	Not null	New Customer Name
birthday	date	Not null	New Customer Birthday
Email	Varchar	Not null	New Customer Email
Phone Number	Int	Not null	New Customer Phone
National Number	Int	Not null	New Customer National

## 3.7.6 Table Structure for New Customer Check:

Table (3.27): DataBase Tables(New Customer Check Table)

Column	Data Type	Index	Description
Id	Int	Primary key	New Customer ID
Username	Text	Not null	New Customer Username
Password	Text	Not null	New Customer Password
Name	Varchar	Not null	New Customer Name
birthday	date	Not null	New Customer Birthday
Email	Varchar	Not null	New Customer Email
Phone Number	Int	Not null	New Customer Phone
National Number	Int	Not null	New Customer National

## 3.7.7 Table Structure for Department:

Table (3.28): DataBase Tables(Department Table)

Column	Data Type	Index	Description
<b>Id</b>	Int	Not Null	Department Id
<b>Name</b>	Text	Not Null	Department Name
<b>code</b>	Float	Not Null	Department Code

## 3.7.8 Table Structure for Blog:

Table (3.29): DataBase Tables(Blog Table)

Column	Data Type	Index	Description
<b>Id</b>	Int	Primary Key	Blog Id
<b>Description</b>	Text	Not Null	Blog Description
<b>Date</b>	Date	Not Null	Blog Date
<b>Images</b>	VARBINARY	Not Null	Blog Images

## 3.7.9 Table Structure for Company Account:

Table (3.30): DataBase Tables(Company Account)

Column	Data Type	Index	Description
<b>Id</b>	Int	Primary Key	Company ID
<b>Name</b>	Varchar	Not Null	Company Name
<b>Email</b>	Varchar	Not Null	Company Email
<b>Balance</b>	Int	Not Null	Company Balance

## 3.7.10 Table Structure for Payment:

Table (3.31): DataBase Tables(Payment Table)

Column	Data Type	Index	Description
<b>Id</b>	Int	Primary Key	Payment ID
<b>Amount</b>	Varchar	Not Null	Payment Amount
<b>Insurance</b>	Varchar	Not Null	Payment Insurance
<b>Balance</b>	Int	Not Null	Payment Balance
<b>CustomerID</b>	Int	Not Null	Payment Customer ID
<b>CustomerName</b>	Varchar	Not Null	Payment Customer Name
<b>EmployeeID</b>	Int	Not Null	Payment Employee ID
<b>EmployeeName</b>	Varchar	Not Null	Payment Employee Name
<b>Company</b>	Varchar	Not Null	Payment Company

## 3.7.11 Table Structure for Associates:

Table (3.32): DataBase Tables (Associates)

Column	Data Type	Index	Description
<b>Id</b>	<b>Int</b>	<b>Primary key</b>	<b>Associates ID</b>
<b>Username</b>	<b>Text</b>	<b>Not null</b>	<b>Associates Username</b>
<b>Password</b>	<b>Text</b>	<b>Not null</b>	<b>Associates Passwrod</b>
<b>Name</b>	<b>Varchar</b>	<b>Not null</b>	<b>Associates Name</b>
<b>Location</b>	<b>date</b>	<b>Not null</b>	<b>Associates Location</b>
<b>Email</b>	<b>Varchar</b>	<b>Not null</b>	<b>Associates Email</b>
<b>PhoneNumber</b>	<b>Int</b>	<b>Not null</b>	<b>Associates Phone</b>
<b>Type</b>	<b>Int</b>	<b>Not null</b>	<b>Associates Type</b>

## 3.7.12 Table Structure for Visit:

Table (3.33): DataBase Tables (Visit)

Column	Data Type	Index	Description
<b>Id</b>	<b>Int</b>	<b>Primary key</b>	<b>V Aisit ID</b>
<b>Client_id</b>	<b>Int</b>	<b>Not null</b>	<b>Client_id Visit</b>
<b>Client_name</b>	<b>Varchar</b>	<b>Not null</b>	<b>Client_name Visit</b>
<b>Assocaites_id</b>	<b>Int</b>	<b>Not null</b>	<b>Associates_id Visit</b>
<b>Associates_name</b>	<b>Varchar</b>	<b>Not null</b>	<b>Associates_name Visit</b>
<b>Date</b>	<b>Datetime</b>	<b>Not null</b>	<b>Date Visit</b>
<b>Note</b>	<b>Varchar</b>	<b>Not null</b>	<b>Note Visit</b>
<b>Status</b>	<b>Int</b>	<b>Not null</b>	<b>Status Visit</b>

## 3.7.13 Table Structure for Note:

Table (3.34): DataBase Tables (Note)

Column	Data Type	Index	Description
<b>Id</b>	<b>Int</b>	<b>Primary Key</b>	<b>Note ID</b>
<b>Assocaites_id</b>	<b>Varchar</b>	<b>Not Null</b>	<b>Assocaites_id Note</b>
<b>Associates_name</b>	<b>Varchar</b>	<b>Not Null</b>	<b>Associates_name Note</b>
<b>Date</b>	<b>Datetime</b>	<b>Not Null</b>	<b>Date Note</b>
<b>Phone</b>	<b>Int</b>	<b>Not Null</b>	<b>Phone Note</b>
<b>Note</b>	<b>Varchar</b>	<b>Not Null</b>	<b>Note</b>

## 3.7.14 Table Structure for Insurance:

Table (3.35): DataBase Tables (Insurance)

Column	Data Type	Index	Description
<b>Id</b>	<b>Int</b>	<b>Primary Key</b>	<b>Insurance ID</b>
<b>Insurance_id</b>	<b>Int</b>	<b>Not Null</b>	<b>Insurance_id Insurance</b>
<b>Type</b>	<b>Varchar</b>	<b>Not Null</b>	<b>Type Insurance</b>
<b>Price</b>	<b>Int</b>	<b>Not Null</b>	<b>Price Insurance</b>
<b>Duration</b>	<b>Int</b>	<b>Not Null</b>	<b>Duration Insurance</b>
<b>Document</b>	<b>Varchar</b>	<b>Not Null</b>	<b>Document Insurance</b>

## 3.7.15 Table Structure for Insurance Request:

Table (3.36): DataBase Tables (Insurance Request)

Column	Data Type	Index	Description
<b>Id</b>	<b>Int</b>	<b>Primary key</b>	<b>Visit ID</b>
<b>Client_id</b>	<b>Int</b>	<b>Not null</b>	<b>Client_id Insurance Request</b>
<b>Client_name</b>	<b>Varchar</b>	<b>Not null</b>	<b>Client_name Insurance Request</b>
<b>Client_age</b>	<b>Int</b>	<b>Not null</b>	<b>Client_age Insurance Request</b>
<b>Insurance_id</b>	<b>Int</b>	<b>Not null</b>	<b>Insurance_id Insurance Request</b>
<b>Insurance_type</b>	<b>Varchar</b>	<b>Not null</b>	<b>Insurance_type Insurance Request</b>
<b>Price</b>	<b>Int</b>	<b>Not null</b>	<b>Price Insurance Request</b>
<b>Duration</b>	<b>Int</b>	<b>Not null</b>	<b>Insurance Request</b>
<b>Photo1</b>		<b>Not null</b>	<b>Insurance Request</b>
<b>Photo2</b>		<b>Not null</b>	<b>Insurance Request</b>
<b>Photo3</b>		<b>Not null</b>	<b>Insurance Request</b>

## 3.8 Unit Testing:

## 3.8.1 Unit Testing of the Add Employee:

Table (3.37): Unit Testing Add Employee

Add employee					
#	TS1				
Title	Verify "add employee" functionality				
Description	Test different scenarios that might happen if admin can add employee				
#	Summery	Pre-condition	Post-condition	Execution Steps	Expected Output
<b>TC 1</b>	Verify that admin successfully add employee	Open menu	employee is validated with database	Select add employee <ul style="list-style-type: none"> <li>- Add valid employee information</li> <li>- Click add</li> </ul>	Employee added
<b>TC 2</b>	Verify employee enable successfully name		Get error response	Select add employee <ul style="list-style-type: none"> <li>- Add invalid name employee(1111)</li> <li>- Click ok</li> </ul>	Get error that gradational invalid
<b>TC 3</b>	Verify employee enable successfully email	Show menu	Get error response	Select add employee <ul style="list-style-type: none"> <li>- Add valid name employee</li> <li>- Add invalid email</li> </ul>	Get error that gradational invalid



				employee(****) - Click ok	
<b>TC 4</b>	Verify employee enable has confirmed the password successfully	Show menu	Get error response	Select add employee - Add valid name employee - Add valid email employee - Add invalid password - Click ok	Get error that gradational invalid
<b>Tc4</b>	Verify employee enable that empty forms	Show menu	Get error response	- Empty form	Get error that gradational invalid

### 3.8.2 Unit Testing of the Login:

**Table (3.38) Unit Testing Login**

Log-in					
#	TS2				
Title	Verify "Log-in" functionality				
Description	Test different scenarios that might happen if user can Log-in				
#	Summery	Pre-condition	Post-condition	Execution Steps	Expected Output
<b>TC 1</b>	Verify that user Successfully log-in	Open Log in page	Successful Log-in	- Select log-in - Add valid login information - Click ok	Open dash bord
<b>TC 2</b>	Verify user enable successfully password		Get error response	- Select log-in - Add password invalid (**) - Click ok	Get error that gradational invalid
<b>TC 3</b>	Verify user enable Verify user enable successfully password email	Open Log in page	Get error response	- Select log-in - Add password valid - Add email invalid or already exists - Click ok	Get error that gradational invalid Or The email already exist

<b>TC 4</b>	Verify that user add empty form	Open Log in page	Get error response	<ul style="list-style-type: none"> <li>- Select log-in</li> <li>- Click ok</li> </ul>	Get error pleas enter Your information
-------------	---------------------------------	------------------	--------------------	---------------------------------------------------------------------------------------	----------------------------------------

### 3.8.3 Unit Testing of the Check Customer Information:

**Table (3.39): Unit Testing Check Customer Information**

Electronic payment					
#	TS3				
Title	Verify "customer Electronic payment " functionality				
Description	Test different scenarios that might happen if customer Electronic payment				
#	Summery	Pre-condition	Post-condition	Execution Steps	Expected Output
<b>TC 1</b>	Verify that customer successfully Electronic payment	Open menu	employee is validated with database	<ul style="list-style-type: none"> <li>- Select payment</li> <li>- Add valid information</li> <li>- Click add</li> </ul>	Electronic payment done
<b>TC 2</b>	Verify customer enable successfully name		Get error response	Select add employee <ul style="list-style-type: none"> <li>- Add invalid name employee(1111)</li> <li>- Click ok</li> </ul>	Get error that gradational invalid
<b>TC 3</b>	Verify customer enable successfully credit card	Show menu	Get error response	Select add employee <ul style="list-style-type: none"> <li>- Add valid name employee</li> <li>- Add invalid credit card (****)</li> <li>- Click ok</li> </ul>	Get error that gradational invalid
<b>TC 4</b>	Verify customar enable has confirmed the date	Show menu	Get error response	Select add employee <ul style="list-style-type: none"> <li>- Add valid name employee</li> <li>- Add valid email employee</li> <li>- Add invalid date</li> <li>- Click ok</li> </ul>	Get error that gradational invalid
<b>Tc4</b>	Verify customar enable that empty forms	Show menu	Get error response	<ul style="list-style-type: none"> <li>- Empty form</li> </ul>	Get error that gradational invalid

### 3.9 Requirements Tracking Matrix:

It is a tool used in detailed analysis for a project to track and relate requirements to other elements of the system such as components, design, tests, and deliverables. The matrix aims to ensure that each requirement is properly met and tracked over the various stages of development. When creating a requirement tracking matrix, a list of requirements is listed in the first two rows or columns of the table. Other columns are then used to track information associated with each requirement, such as components that implement the requirement, related design, associated tests, and implementation status of the requirement.

**Table (3.40): Requirements Tracking Matrix**

ID	Description of the Requirement	Associated Components	Associated Design	Design	Execution Status	Test
REQ001	Login Admin and Customer and Employee	Login Components	Login Page	Finished	Underway	TS2
REQ002	The admin sends note to the employees	Send Notes Components	Sends Notes Page	Finished	Underway	
REQ003	The admin Edit Employees	Edit Employees Components	Edit Employees Page	Finished	Underway	
REQ004	The admin Add Employees	Add Employees Components	Add Employees Page	Finished	Underway	TS1
REQ004	The admin Edit Department	Edit Department Components	Edit Department Page	Finished	Underway	
REQ005	The admin Add Department	Add Department Components	Add Department Page	Finished	Underway	
REQ006	The Admin View Customer Visit Details	View Customer Visit Details Components	View Customer Visit Details Page	Finished	Underway	
REQ007	Customer Affairs Edit Customer	Edit Customer Components	Edit Insurance Page	Finished	Underway	
REQ008	Customer Affairs Add Customer	Add Customer Components	A Add Customer Page	Finished	Underway	
REQ009	Customer Affairs Check Customer Information	Check Customer Information Components	Check Customer Information Pag	Finished	Underway	

REQ010	Insurance Affairs Edit Insurance	Edit Insurance Components	Edit Insurance Page	Finished	Underway	
REQ011	Insurance Affairs Add Insurance	Add Insurance Components	Add Insurance	Finished	Underway	
REQ012	Customer Account Statement	Account Statement Components	Account Statement Page	Finished	Underway	
REQ013	Customer Register a New Insurance	Register a New Insurance Components	Register a New Insurance Page	Finished	Underway	
REQ014	Customer View Insurance Details	View Insurance Details Components	View Insurance Details Page	Finished	Underway	
REQ015	Customer Electronic Payment	Electronic Payment Components	Electronic payment Page	Finished	Underway	TS3
REQ016	Finance Officer Payment of the Client Insurance	Payment of the Client Insurance Components	Payment of the Client Insurance Page.	Finished	Underway	
REQ017	Blogger Employee Add News	Add News Components	Add News	Finished	Underway	
	Blogger Employee Edit News	Employee Edit News Components	Employee Edit News Page.	Finished	Underway	
REQ018	Associates Send Notes	Send Notes Components	Send Notes Page	Finished	Underway	
REQ019	Associates Enter Customer Visit Information	Enter Customer Visit Information Components	Enter Customer Visit Information Page	Finished	Underway	
REQ020	Associates Edit Profile	Edit Profile Components	Edit Profile Page	Finished	Underway	

### 3.10 System Architecture:

System architecture refers to the overall structure and organization of components in a system. Aim to achieve objectives effectively and systematically, taking into account compatibility, scalability, security, maintainability, and performance. In an e-learning system, it can include components such as a database, user interface, content management system, and assessment tools.

### **3.10.1 Client-Server Architecture:**

The client-server pattern is an architectural pattern for software design, in which an application is divided into two main parts: the client and the server. The client interacts with the server by repeatedly sending requests and responses.

### **3.10.2 Features of this Architecture:**

1. **Differentiation between functions:** This pattern helps to differentiate the functions of the program into different components, as the program is divided into two main parts: the server and the client.
2. **Ease of development:** This style makes it easier for developers to develop applications, as it becomes clear to them the roles of each of the components.
3. **Ease of maintenance:** The roles of each of the components are clear to the manufacturer, which makes it easy to maintain the application in the event of any problems.
4. **Changeability:** This mode makes it easy for the user to make any changes in a part of the components without affecting the rest of the components. Increased use efficiency
5. **Resources:** This style helps increase resource efficiency, as work is divided between the server and the client, making it easy to determine the resources each component needs.

### **3.10.3 Why to Use?**

1. **Application performance optimization:** Work is distributed between the client and the server, which helps improve application performance.
2. **Ease of maintenance:** Client and server functions are separated, making it easy to perform updates and repairs on each aspect separately.
3. **Increased security:** User data and systems used in the server are separated, making it difficult for an unauthorized person to access this information.
4. **Scalability:** You can add new servers to your system without affecting the rest of the application.

3.10.4 Client-Server Diagram:

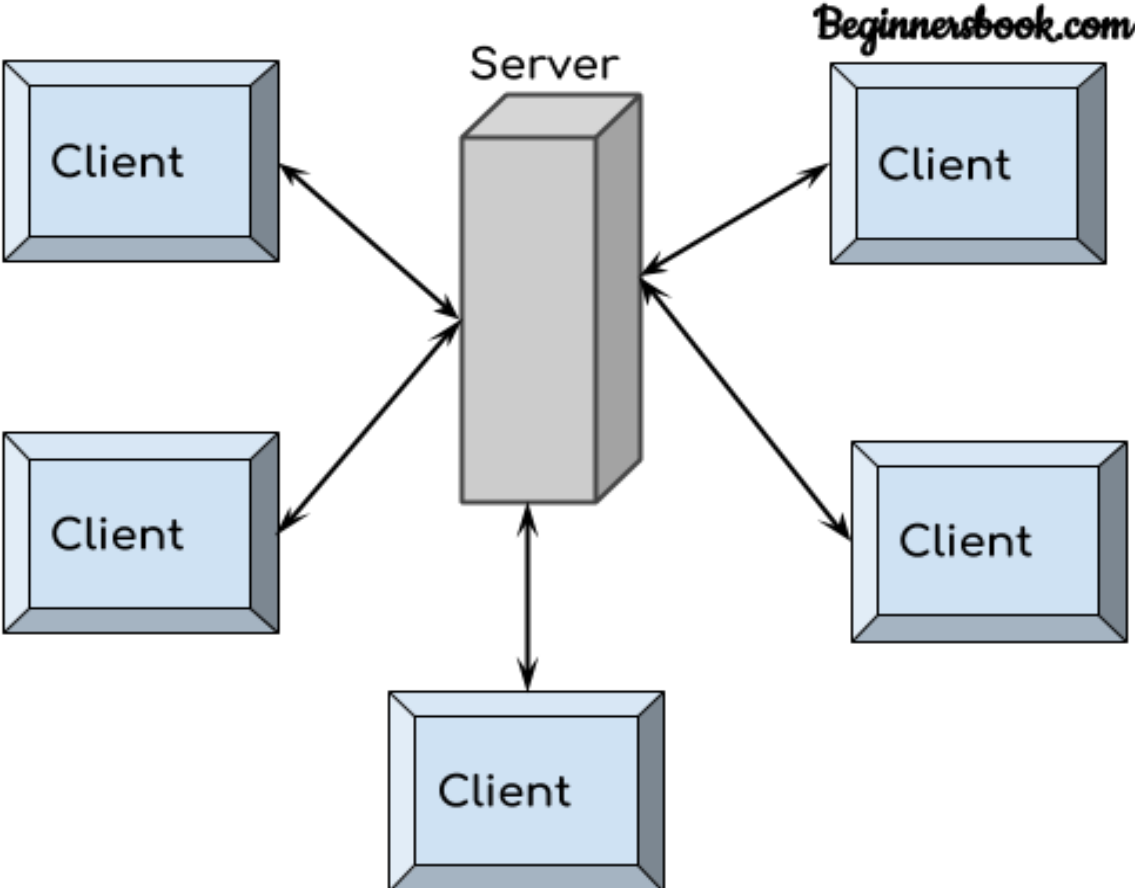


Figure (3.21): Client-Server Diagram

**Chapter Four**  
**The Design Phase of the Proposed**  
**System**

## 4 Chapter Four-Design of the Proposed Project

### 4.1 Class Diagram:

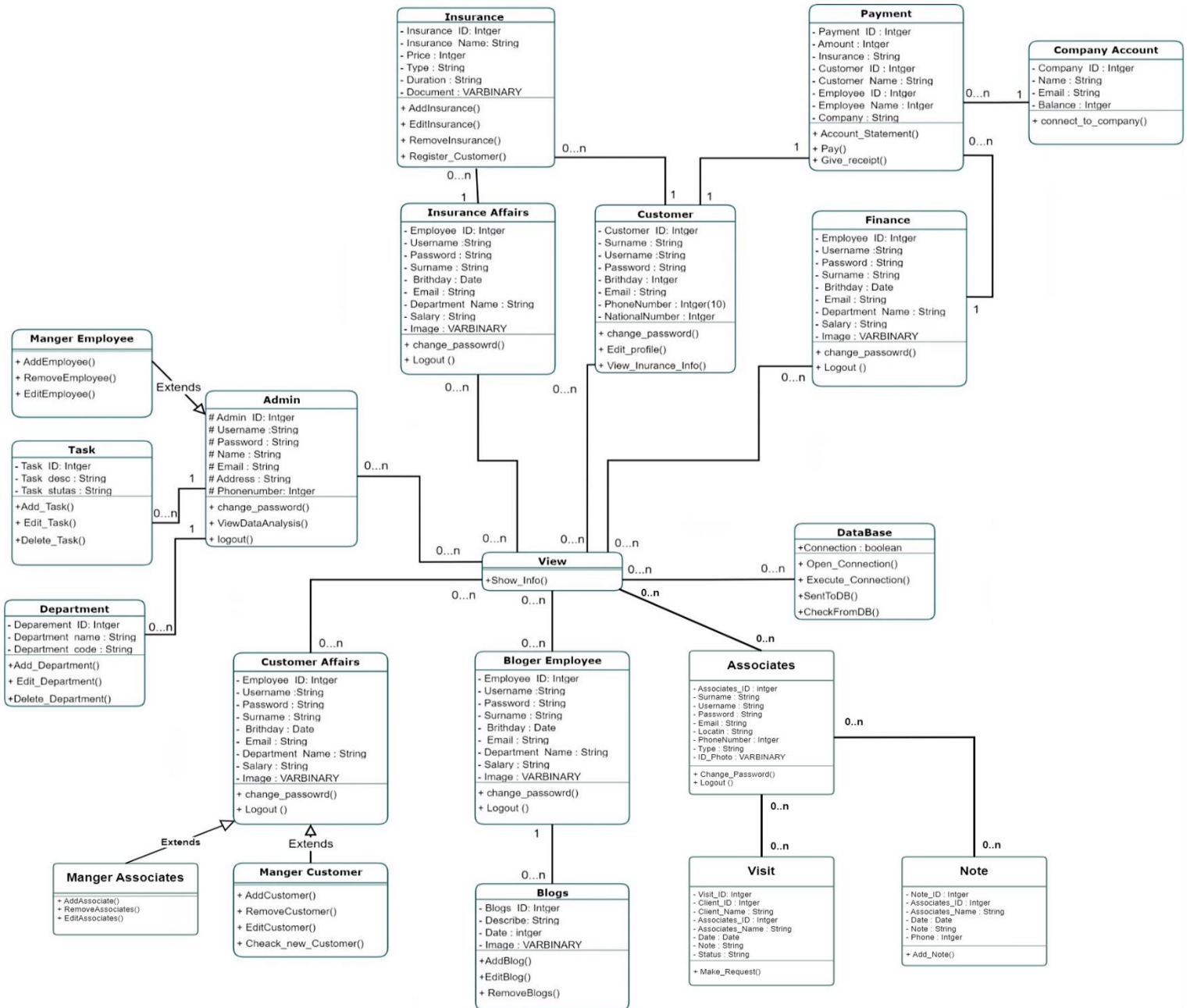


Figure (4.22): Class Diagram



## 4.2 Design Models at the Physical Level:

An example of a physical-level design template for an insurance company management project:

### 1. The Devices Used:

1. Server: Used to host the application and databases.
2. Laptop computers: used by the admin, employee, client and associated to access the system.

### 2. Software and Technologies Used:

1. Programming languages: using programming languages such as PHP, HTML and CSS to develop the user interface and interaction processes on the platform.
2. Databases: Using a database management system such as MySQL to store user data.
3. Web servers: Using a web server such as Apache to run the application and connect it to browsers.
4. Framework: Using the CodeIgniter framework to facilitate application development and management of system-related activities.

### 3. Technical infrastructure:

1. Internet: The project relies on an internet connection to allow users to access the platform.
2. Security and Protection: Security technologies such as SHA1 and MD5 encryption are used in the login.

The above form contains technical details about the hardware, software and infrastructure that will be used to implement this system at the physical level. You can elaborate on more details and specific technologies depending on the project's needs and technical requirements.

4.3 Web Page Site:

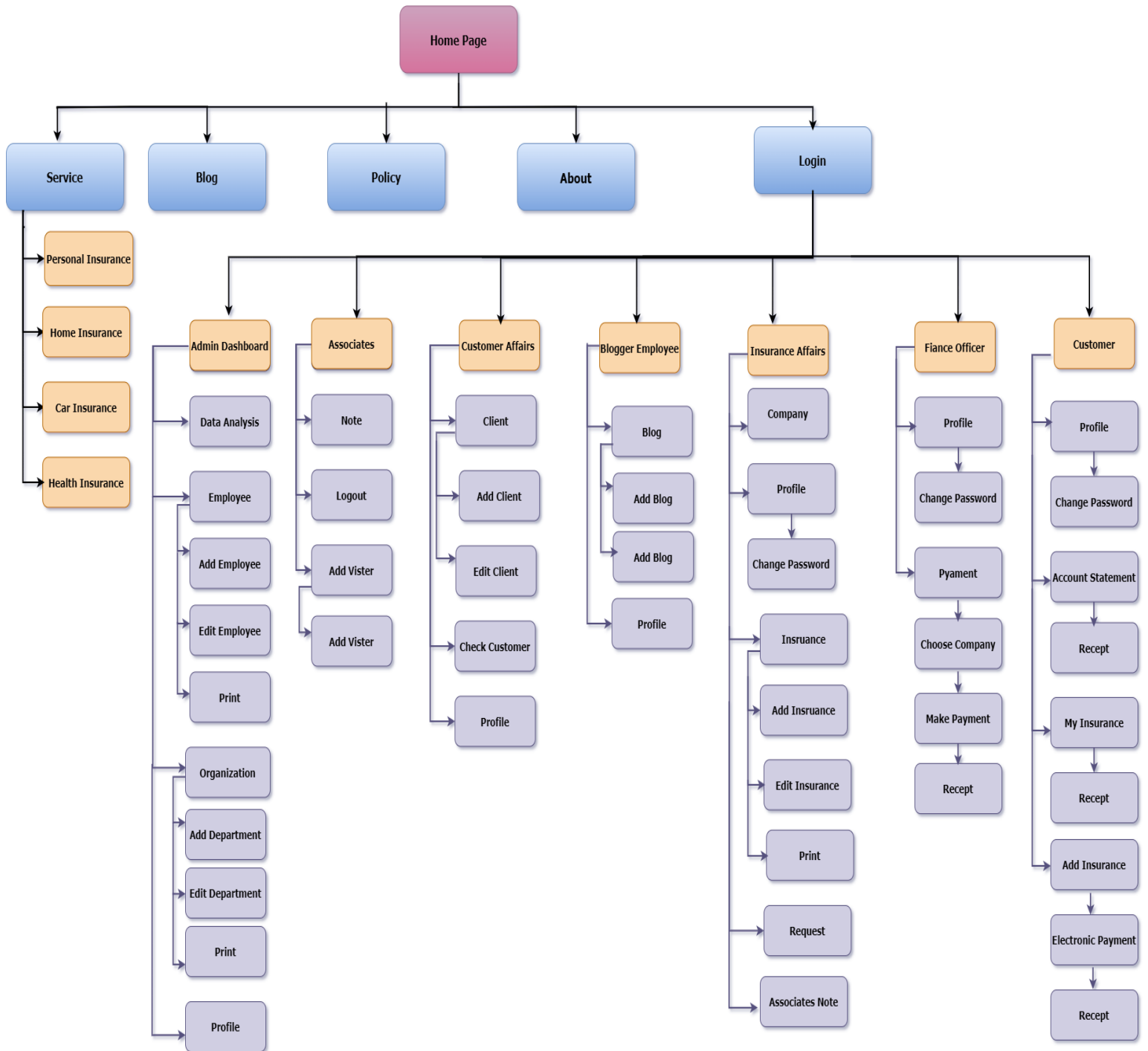


Figure (4.23): Web Page Site

## 4.4 Update Requirements Tracking Matrix:

Table (4.41): Update Requirements Tracking Matrix

ID	Description of the Requirement	Associated Components	Associated Design	Design	Execution Status	Test
REQ001	Login Admin and Customer and Employee	Login Components	Login Page	Finished	Done	Done
REQ002	The admin sends note to the employees	Send Notes Components	Sends Notes Page	Finished	Done	
REQ003	The admin Edit Employees	Edit Employees Components	Edit Employees Page	Finished	Done	
REQ004	The admin Add Employees	Add Employees Components	Add Employees Page	Finished	Done	Done
REQ004	The admin Edit Department	Edit Department Components	Edit Department Page	Finished	Done	
REQ005	The admin Add Department	Add Department Components	Add Department Page	Finished	Done	
REQ006	The Admin View Customer Visit Details	View Customer Visit Details Components	View Customer Visit Details Page	Finished	Done	
REQ007	Customer Affairs Edit Customer	Edit Customer Components	Edit Insurance Page	Finished	Done	
REQ008	Customer Affairs Add Customer	Add Customer Components	A Add Customer Page	Finished	Done	
REQ009	Customer Affairs Check Customer Information	Check Customer Information Components	Check Customer Information Pag	Finished	Done	
REQ010	Insurance Affairs Edit Insurance	Edit Insurance Components	Edit Insurance Page	Finished	Done	
REQ011	Insurance Affairs Add Insurance	Add Insurance Components	Add Insurance	Finished	Done	
REQ012	Customer Account Statement	Account Statement Components	Account Statement Page	Finished	Done	
REQ013	Customer Register a New Insurance	Register a New Insurance Components	Register a New	Finished	Done	

			Insurance Page			
REQ014	Customer View Insurance Details	View Insurance Details Components	View Insurance Details Page	Finished	Done	
REQ015	Customer Electronic Payment	Electronic Payment Components	Electronic payment Page	Finished	Done	Done
REQ016	Finance Officer Payment of the Client Insurance	Payment of the Client Insurance Components	Payment of the Client Insurance Page.	Finished	Done	
REQ017	Blogger Employee Add News	Add News Components	Add News	Finished	Done	
	Blogger Employee Edit News	Employee Edit News Components	Employee Edit News Page.	Finished	Done	
REQ018	Associates Send Notes	Send Notes Components	Send Notes Page	Finished	Done	
REQ019	Associates Enter Customer Visit Information	Enter Customer Visit Information Components	Enter Customer Visit Information Page	Finished	Done	
REQ020	Associates Edit Profile	Edit Profile Components	Edit Profile Page	Finished	Done	

# **5 Chapter Five**

## **Implementation of Project**

## 5.1 Programming Resource:

### 5.1.1 Bootstrap5 (Front-End-Framework):

It is the most famous and powerful libraries that combine HTML, CSS and JS to build pages and applications, and it is mainly useful for building compatible sites with different screen sizes and sizes, including, cell phones, and it is a public and open-source library for everyone on the site.

We have relied on this platform in our project because:

1. Great flexibility and ease of use.
2. Designed to be compatible with all devices.
3. Compatible with all browsers.
4. Offers a stable and reusable design.
5. It can be learned easily and quickly.
6. Great scalability using JavaScript.
7. It has support for jQuery.
8. Support for the JavaScript API.

Can be used in any editor or programming environment.

## 5.2 Programming Tools:

### 5.2.1 XAMPP Local Host:

The Project was implemented using the local server (Apache Web Server) and has the following features:

1. Free and open source: the program has been well received and has built its own large community of users and developers, and received the necessary updates as quickly as possible, making it an ideal choice for many.
2. XAMPP supports many programs and services in embedded form, including the following:
  1. X: Supports Multiple Platforms.
  2. A: Supports Apache HTTP Server.
  3. M: Supports MariaDB (formerly MySQL rules).
  4. P: Supports the Programming Language PHP.
  5. P: Supports the Programming Language Perl.
3. Multi-Platform: Supports (Windows, Linux, MacOS).

Easy to Install and Used: The program comes with all its contents within a single installation package, which makes it easy for the user to install and set it up automatically without the need to do any manual setup process.

### **5.3 Technologies Used:**

#### **5.3.1 Html5:**

HTML, or Hypertext Markup Language, allows web users to create and structure sections, paragraphs, and links using elements, tags, and attributes. However, it's worth noting that HTML is not considered a programming language as it can't create dynamic functionality.

#### **5.3.2 CSS3:**

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colours are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

#### **5.3.3 JAVASCRIPT:**

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive.

Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Common examples of JavaScript that you might use every day include the search box on Amazon, a news recap video embedded on the New York Times, or refreshing your Twitter feed.

Incorporating JavaScript improves the user experience of the web page by converting it from a static page into an interactive one. To recap, JavaScript adds behaviour to web pages.

#### **5.3.4 PHP:**

PHP(short for Hypertext Pre-processor) is the most widely used open source and general-purpose server-side scripting language used mainly in web development to create dynamic websites and applications. It was developed in 1994 by Rasmus Leadoff. A survey by W3Tech shows that almost 79% of the websites in their data are developed using PHP. It is not only used to build the web apps of many tech giants like Facebook but is also used to build many CMS (Content Management System) like WordPress, Drupal, Shopify, WooCommerce etc. PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website. For

example, PHP can receive data from forms, generate dynamic page content, can work with databases, create sessions, send and receive cookies, send emails etc. There are also many hash functions available in PHP to encrypt user's data that makes PHP secure and reliable to be used as a server-side scripting language. So, these are some of the abilities of PHP that makes it suitable to be used as server-side scripting language. You will get to know more of these abilities in further tutorials.

### **5.3.5 SMTP:**

The SMTP for Gmail is a feature you can use to send emails through different email clients using Gmail address or domain.

It was used when the employee accepts the sign-up request, after which an email is sent to the new client that his request has been accepted.

## **5.4 Interfaces Pages:**

### **5.4.1 (Admin) Log-In Page:**

On this page, the system manager enters his user's name and password, and then the site verifies the entered user name and password, and if they are correct, then it will move it to the main interface, but if the user's name or password is wrong, then it will show a message that there is no one with this information entered.

### **5.4.2 (Admin) Dashboard:**

On this page, the site displays the system control panel, through which all information that includes a customer, employee, department, notes, insurances, data analysis, and personal information can be viewed.

### **5.4.3 (Customer Affairs) View Customer Details:**

On this page, the site displays the interface of the details of the customer's visit to the subscribers, whether car companies, hospitals, or doctors, and views whether there are any comments by the subscribers.

### **5.4.4 (Admin) Add Task to Employee:**

On this page, the site displays an interface through which the admin sends messages to all employees.



#### **5.4.5 (Admin) Profile:**

On this page, the site displays all the information about the Admin, which include ID, User-Name, Email , Password, Privilege and he can modify his login information.

#### **5.4.6 (Admin) Mange Employee:**

On this page, the system displays the employee management interface, through which the admin can add, modify, and delete the employee and view the employee's information.

##### **5.4.6.1 (Admin) Add Employee:**

On this page, the site displays the interface for adding a new employee. The admin enters the required information for the employee within the existing fields, and then creates a new employee.

##### **5.4.6.2 (Admin) Edit Employee:**

On this page, the site displays an employee modification interface. The admin modifies the information to be modified for the employee within the existing fields, and then amends the information for the employee.

##### **5.4.6.3 (Admin) Delete Employee:**

On this page, the site displays an interface for deleting an employee. The admin searches for the employee to be deleted, and the admin deletes him if he exists.

#### **5.4.7 (Customer) Electronic Payment:**

On this page, the site displays the electronic payment interface for the customer, through which he can pay the dues to the company that he subscribes to.

#### **5.4.8 (Customer) Register a New Insurance:**

On this page, the site displays the registration interface for a new insurance. The customer selects the insurance to be registered with, enters the required information, and then registers for it.

#### **5.4.9 (Finance Officer) Payment of Customer Insurance:**

On this page, the site displays an interface through which the company employee pays the customer's dues when the customer is unable to pay electronically.

**5.4.10 (Customer Affairs) Check Customer Information:**

On this page, the website will display the interface for verifying the information that the customer entered for the first time in registering with the system and verifying whether the information entered is correct or not.

**5.4.11 (Associates) Enter Customer Visit Information:**

On this page, the site presents an interface for the person connected to the system, an interface through which he can enter notes or problems and send them to the admin to view them.

**5.4.12 (Insurance Affairs) Mange Insurance:**

On this page, the site displays an interface for the employee to manage insurance operations, including adding, deleting, and modifying all insurances.

**5.4.12.1 (Insurance Affairs) Add Insurance:**

On this page, the site displays the interface for adding a new insurance. The insurance affairs enters the required information for the insurance within the existing fields, and then creates a new insurance.

**5.4.12.2 (Insurance Affairs) Edit Insurance:**

On this page, the site displays an insurance modification interface. The insurance affairs modifies the information to be modified for the insurance within the existing fields, and then amends the information for the insurance.

**5.4.12.3 (Insurance Affairs) Delete Insurance:**

On this page, the site displays an interface for deleting an insurance. The insurance affairs searches for the insurance to be deleted, and the insurance affairs deletes him if he exists

5.5 Interfaces Screen:

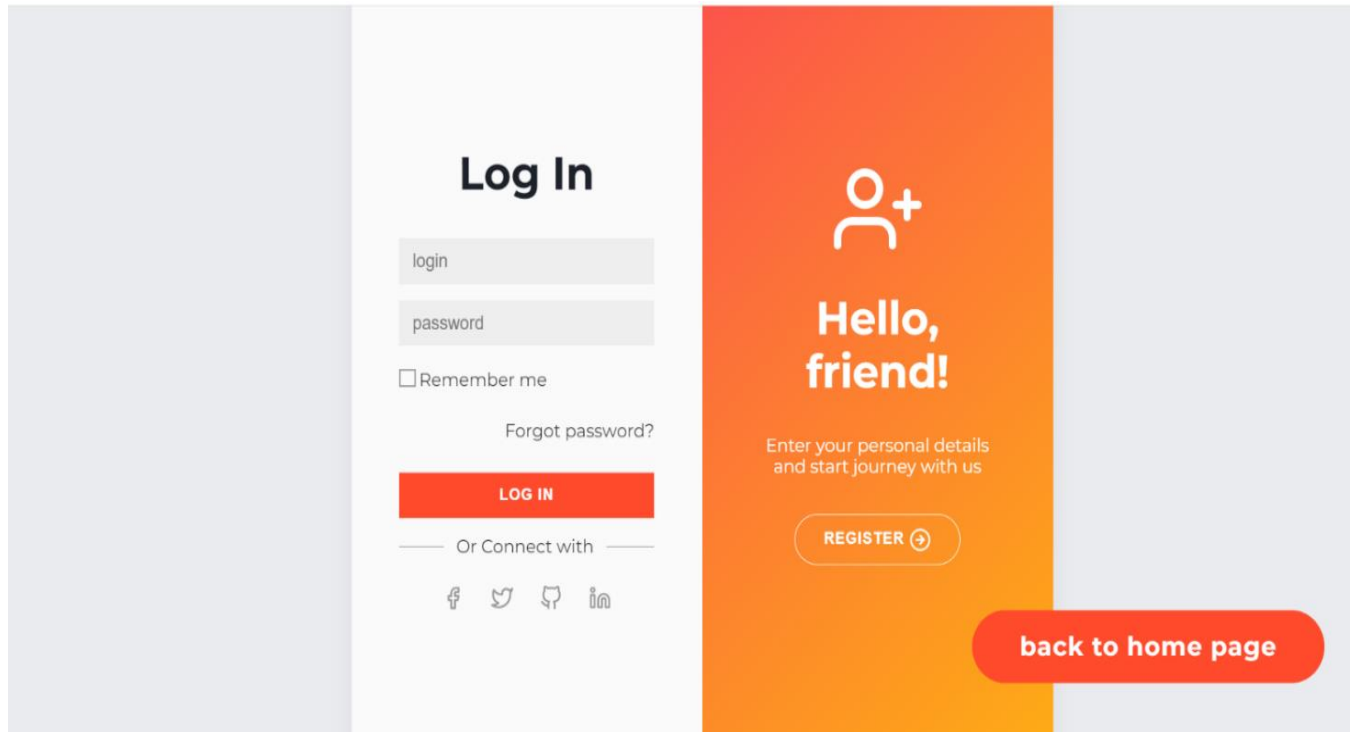


Figure (5.24): Admin – Login Page

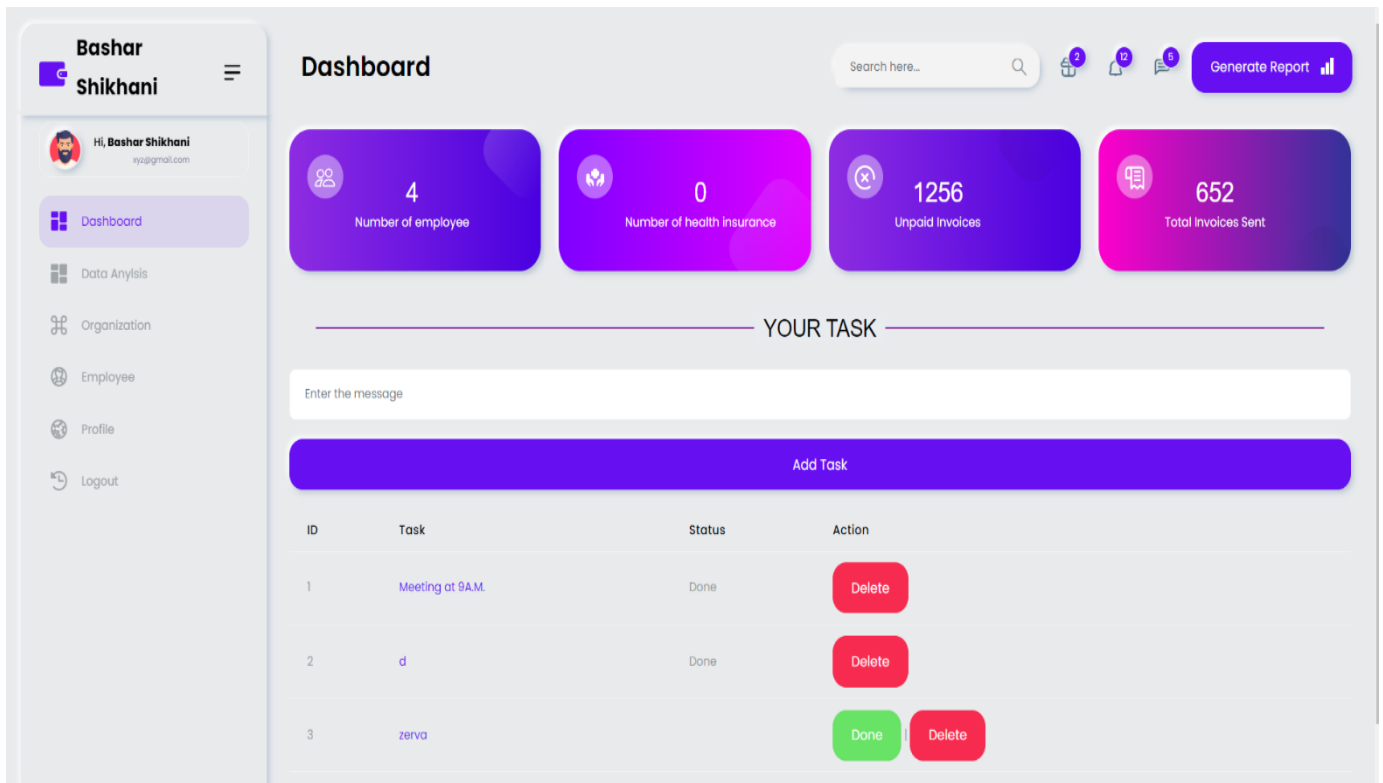


Figure (5.25): Admin – Dashboard Page

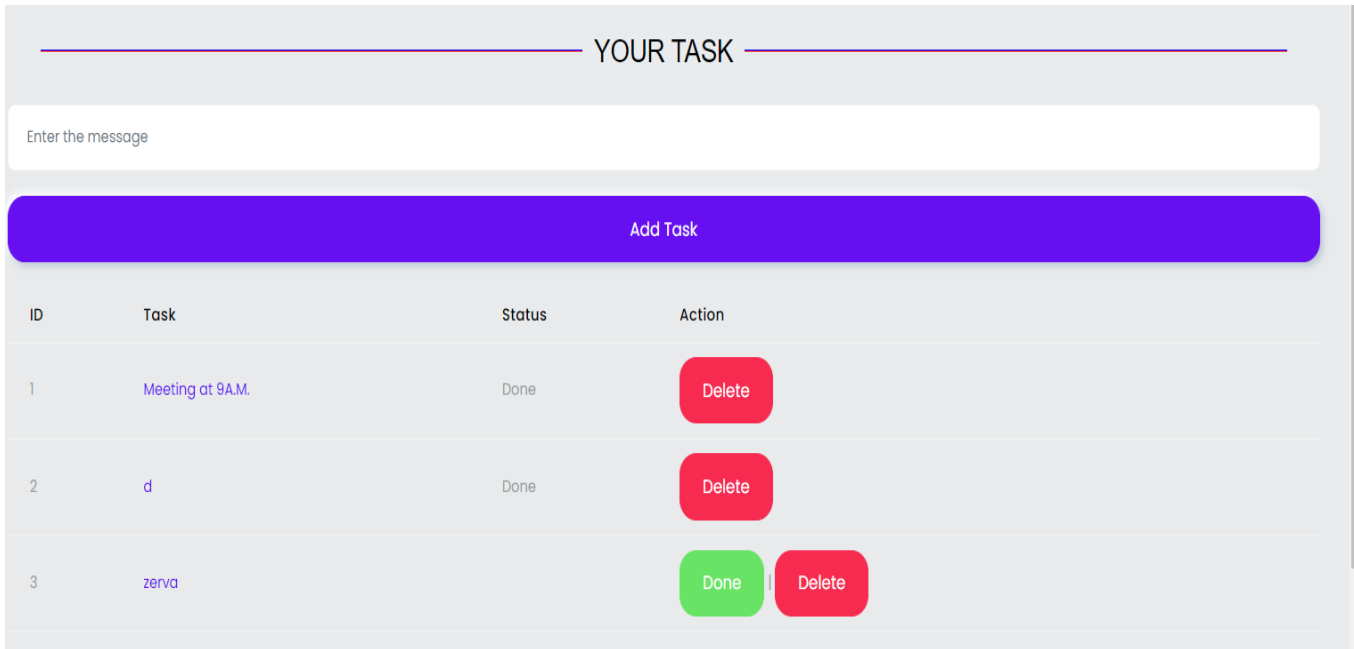


Figure (5.26): Admin – Add Task to Employee Page

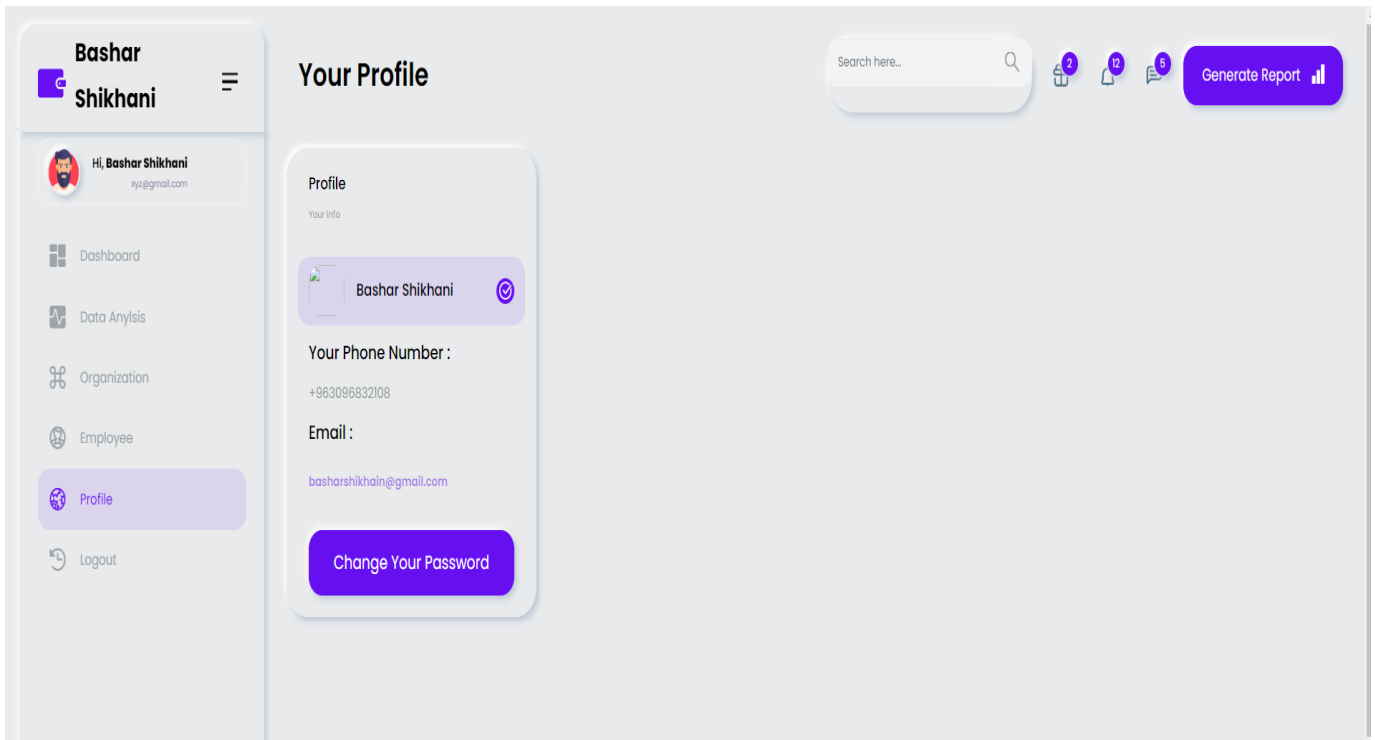


Figure (5.27): Admin – Profile Page

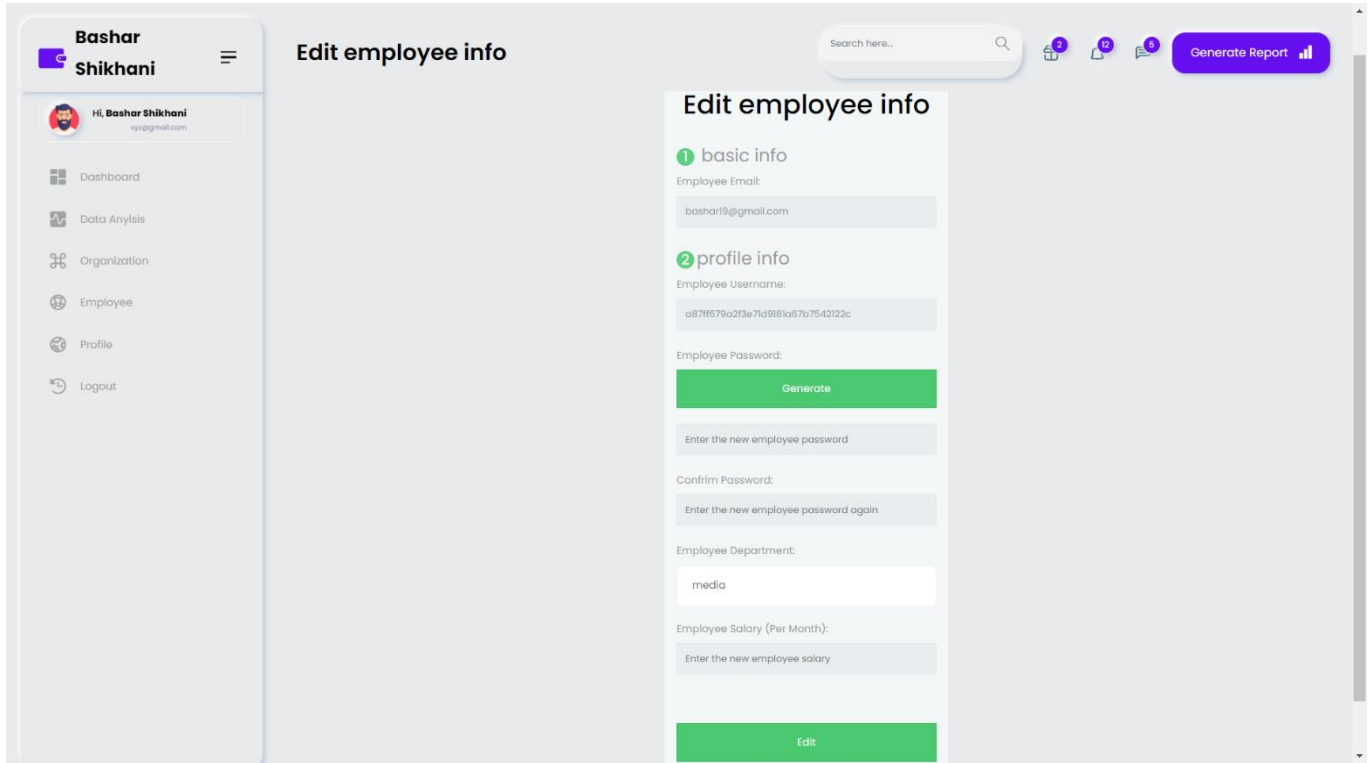


Figure (5.28): Admin – Add New Employee Page

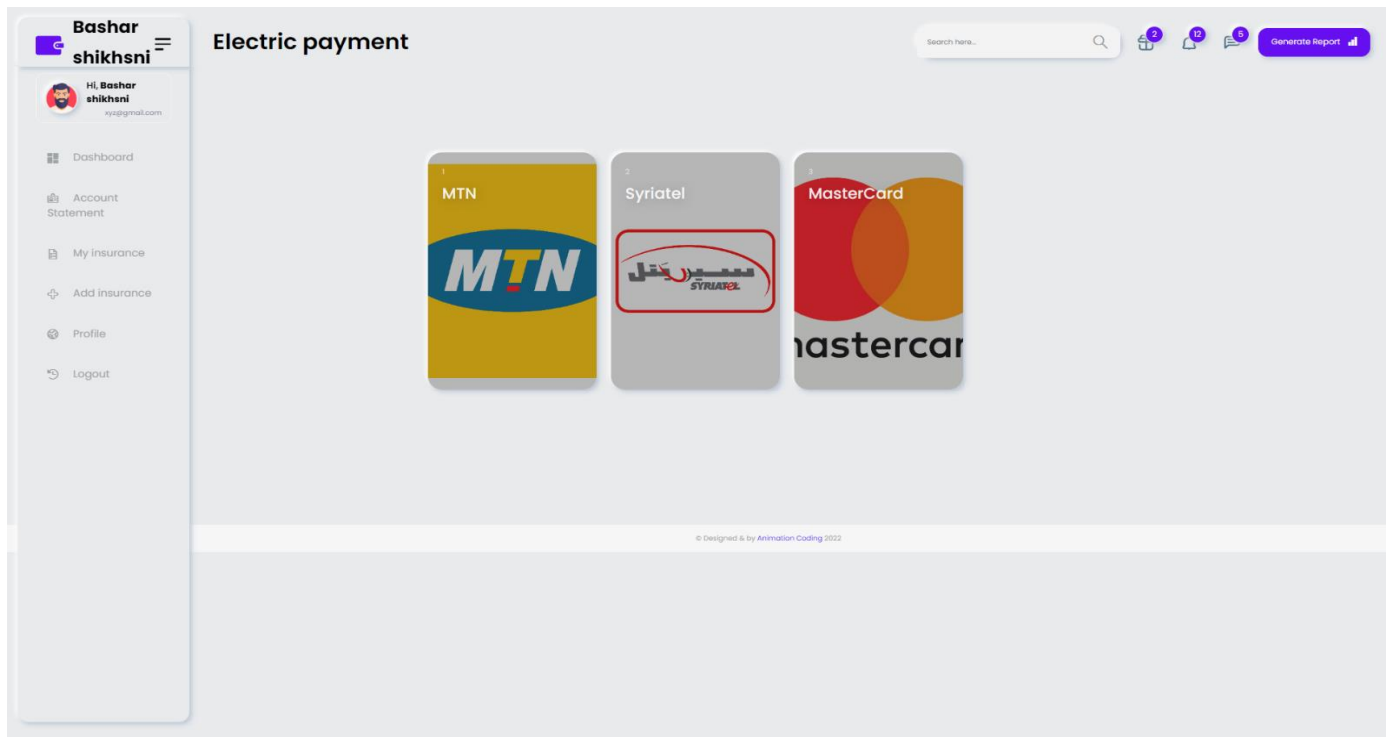


Figure (5.29): Customer – Electronic Payment Page

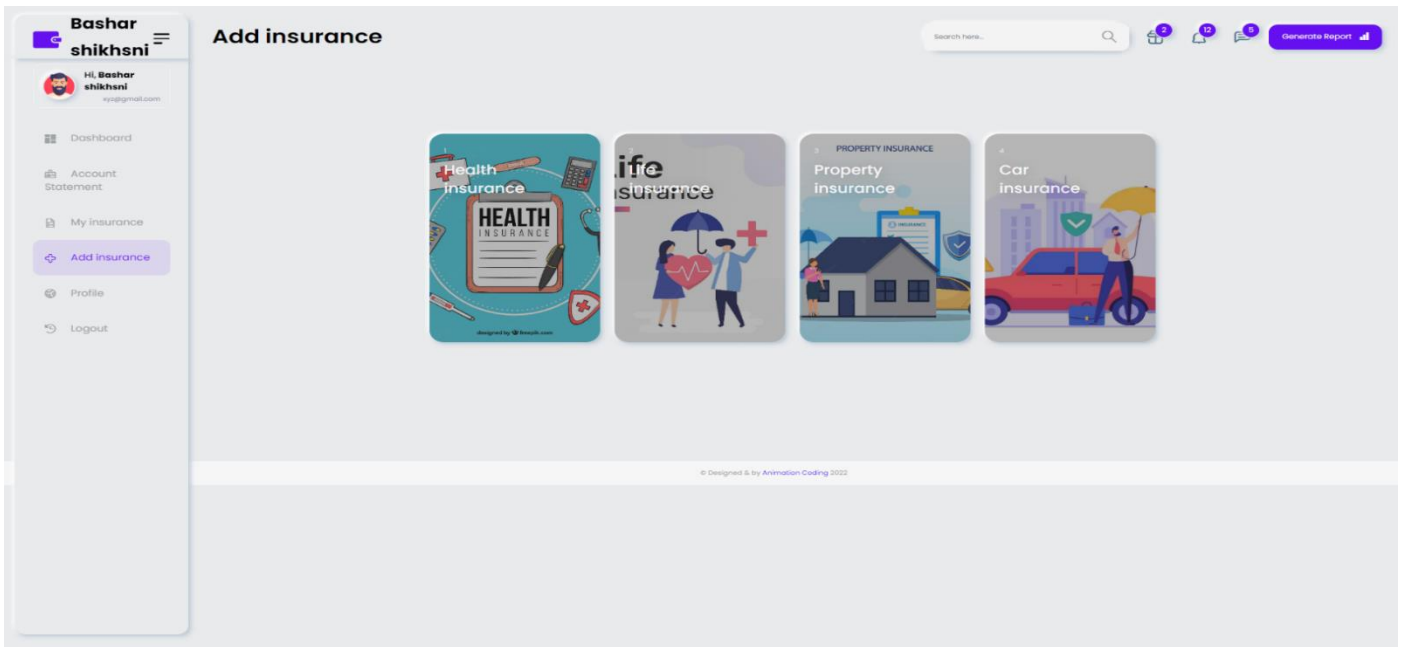


Figure (5.30): Customer – Register A New Insurance Page

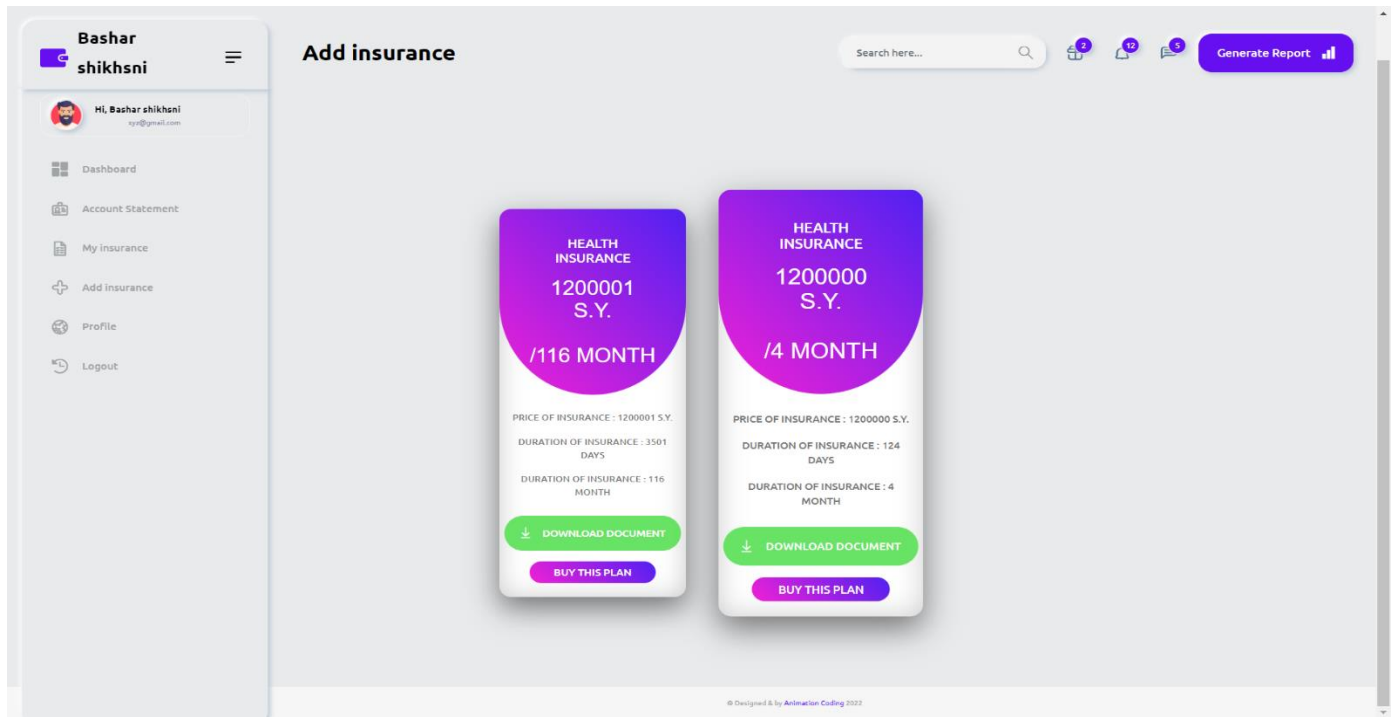


Figure (5.31): Customer – Register A New Insurance Page

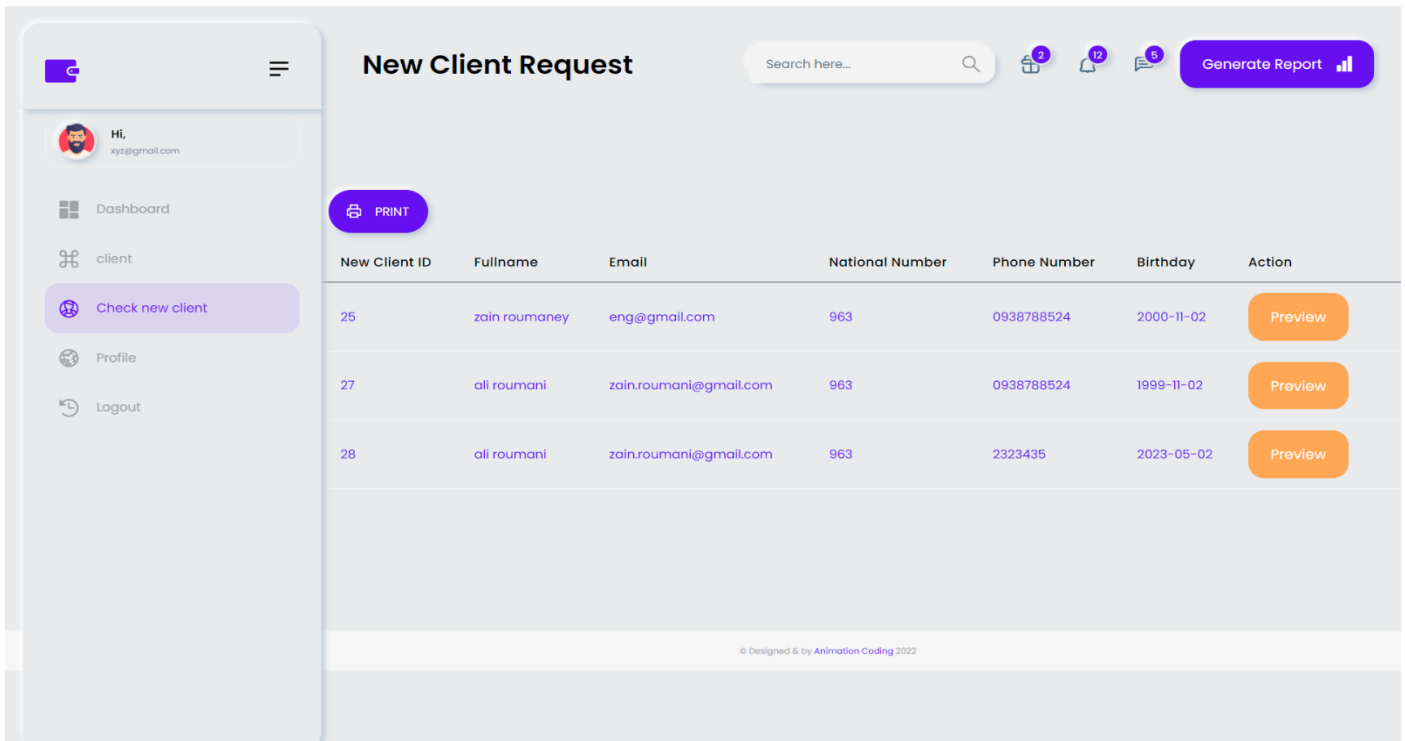


Figure (5.32): Customer Affairs – Check Customer Information Page

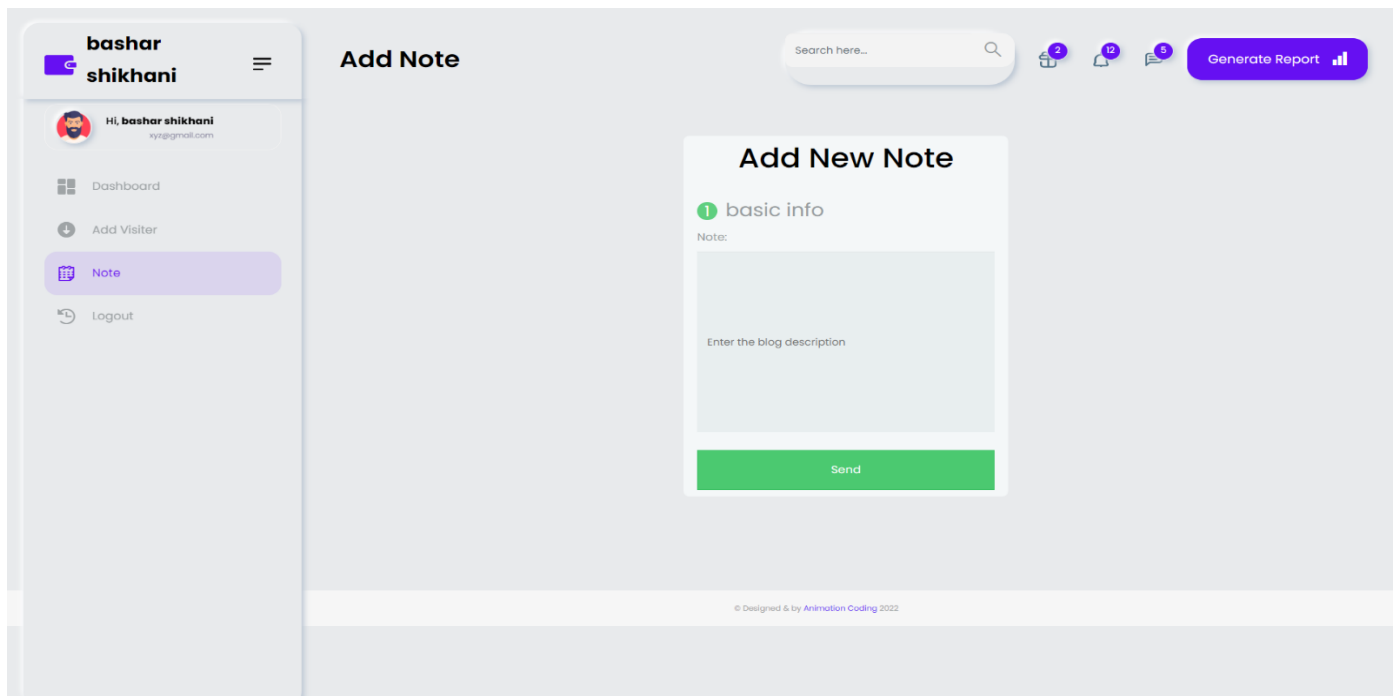


Figure (5.33): Associates - Enter Customer Visit Information Page

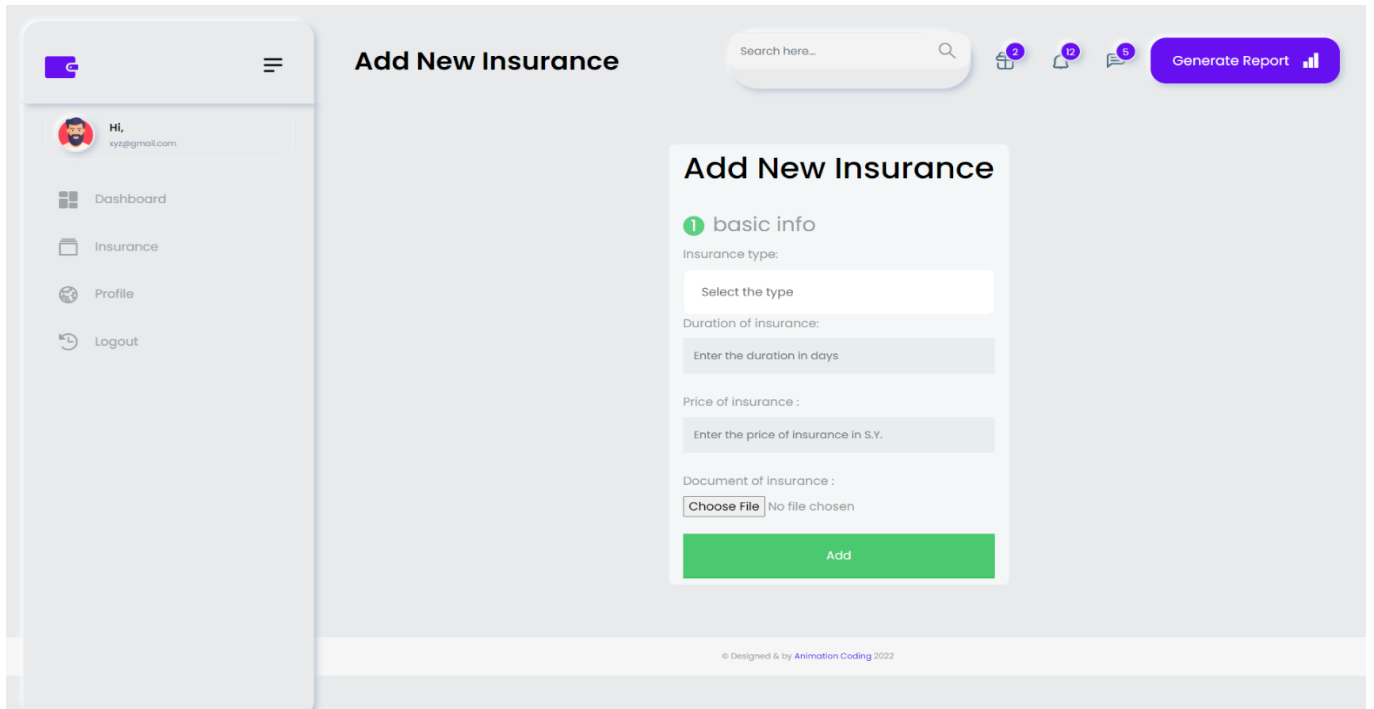


Figure (5.34): Insurance Affairs – Add Insurance Page

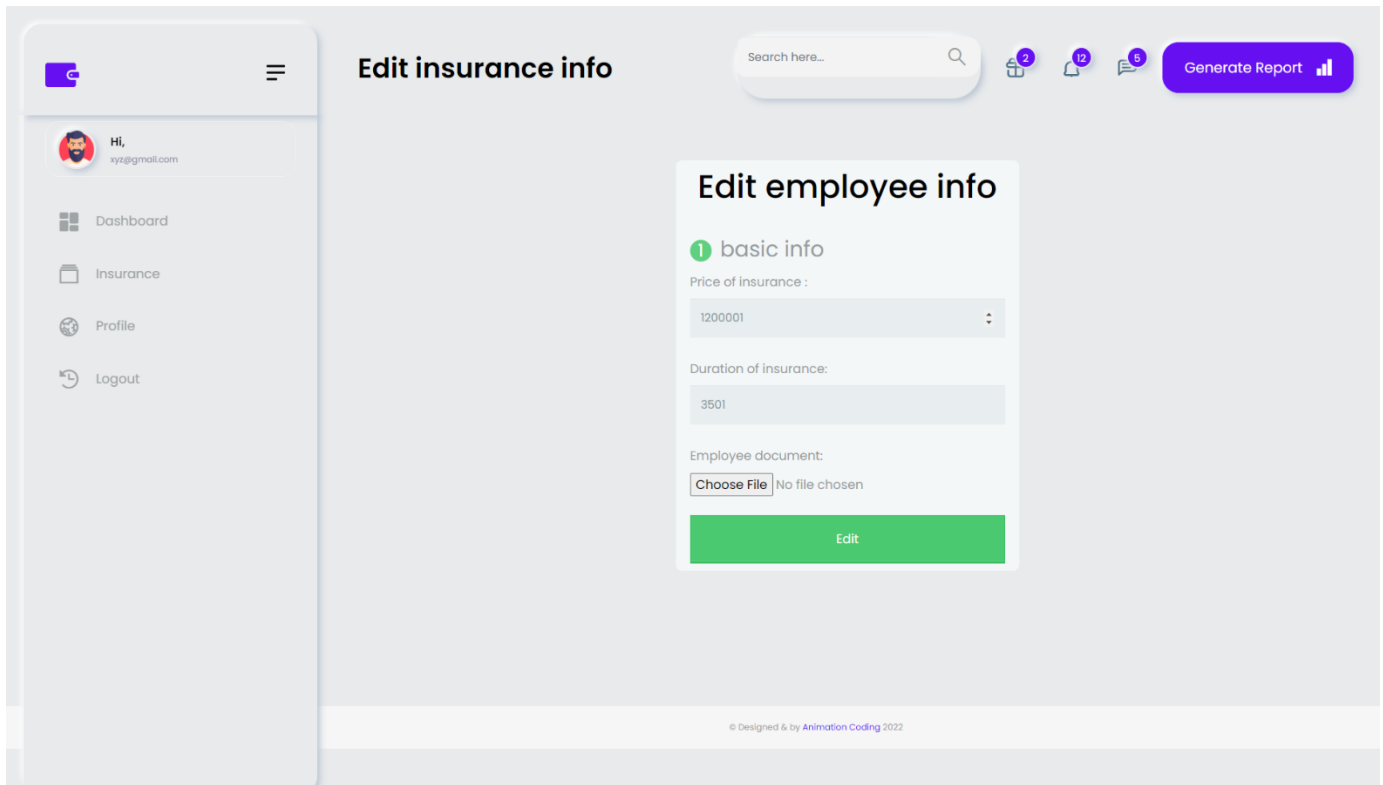


Figure (5.35): Insurance Affairs – Edit Insurance



## **6 Chapter Six Test and Result**

## **6.1 Introduction:**

The biggest challenge lies in verifying the correct functioning of the system and performing the required functions correctly, and this requires a sufficient amount of tests, and of course, the correct way to test any system must be carried out according to a correct methodology and successive stages to ensure the correctness of the workflow, and therefore a set of tests has been implemented During the development stages of the system, and when performing the test, it is ensured that there are no errors within the software because some of these errors remain latent until the system is running, which leads to the failure of the system to perform its functions. The errors detected vary in their degree of seriousness, because the error resulting from the software investigation stage is considered one of the easy mistakes, while the error resulting from the lack of a correct understanding of the problem, which appears clearly in the analysis stage, which leads to errors in all subsequent stages, and the same is true for the design, the error in this stage It may be due to the designer's inexperience or not understanding the analysis properly.

### **6.1.1 Tests During Development:**

#### **6.1.1.1 *Software Unit Testing:***

It is the test that each member of the project has done on the software unit that he has developed, and thus each individual is responsible for doing this type of test.

#### **6.1.1.2 *Integration Test :***

It is the test through which the compatibility of the software interfaces with each other and their ability to work with each other is verified. This test is done when the software modules are grouped together to perform a function of the system.

#### **6.1.1.3 *Quality Tests:***

They are the tests for the domain in which the system was developed.

### **6.1.2 Tests After Development:**

#### **6.1.2.1 *Google Lighthouse:***

A new tool by Google is to test and give you the recommendation to improve performance, SEO, security, best practices, and accessibility. There are more than 75 metrics it tests and gives you overall scoring. You might be interested in some of the following popular ones as a site owner, SEO analyst, and webmaster.

1. Performance – time to interactive, latency, speed index, resources optimization, TTFB, asset delivery, scripts execution time, DOM size, etc.
2. SEO – Mobile friendly, meta, crawling, canonical, structure, etc.
3. Best Practices – Image optimization, JS libraries, browser error logging, accessible over HTTPS, known JS vulnerabilities, etc.
4. Accessibility – Page elements, language, ARIA attributes, etc.
5. PWA (Progressive Web Application) – redirect HTTP to HTTPS, response code ok, fast loading on 3G, splash screen, viewport, etc.

### 6.1.3 Testing Result (In Performance):

#### 6.1.3.1 Arope App:

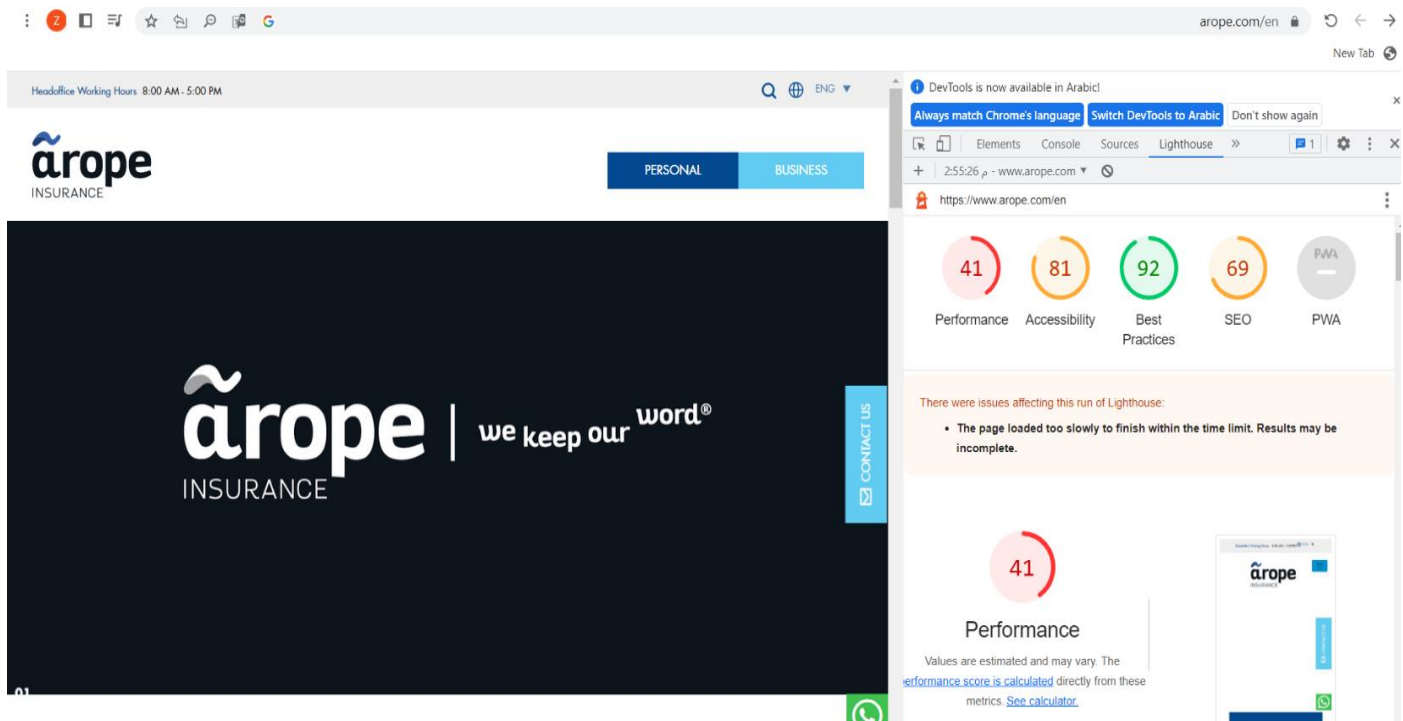


Figure (6.36): Testing Result-Arope App

### 6.1.3.2 BINDQH App:

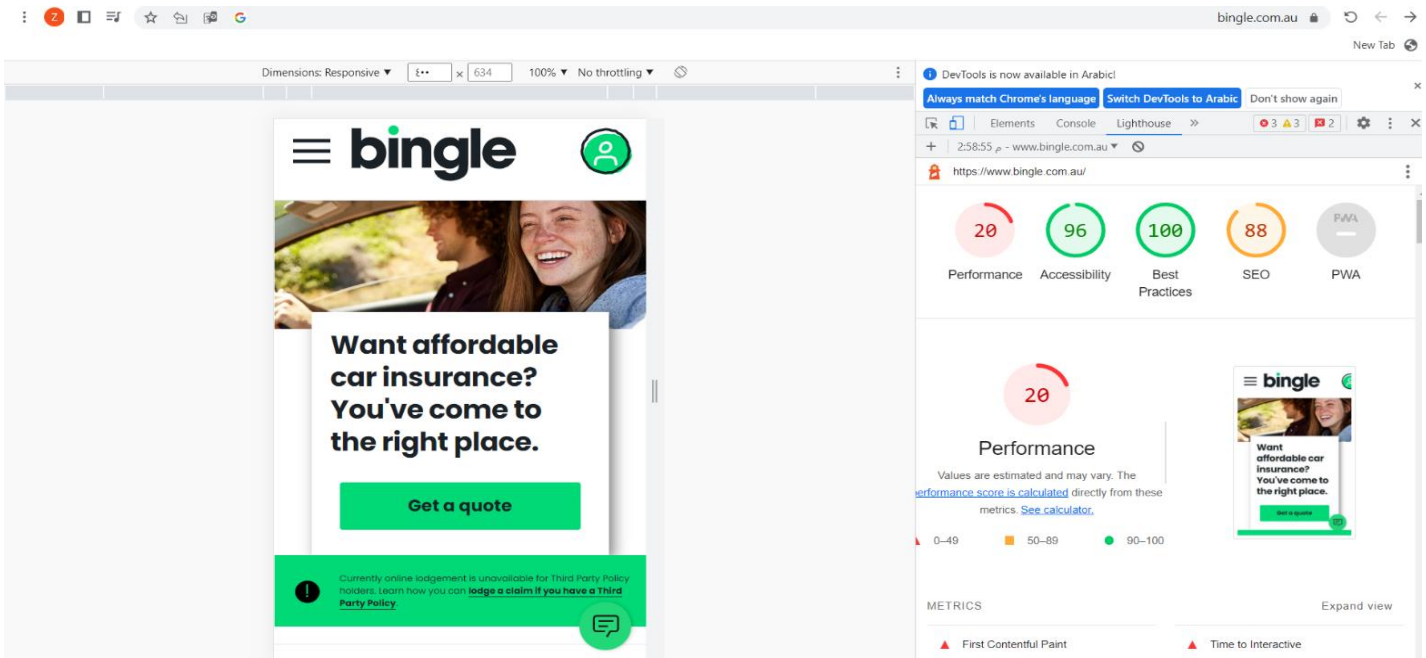


Figure (6.37): Testing Result - BINDQH App

### 6.1.3.3 TLDCRM App:

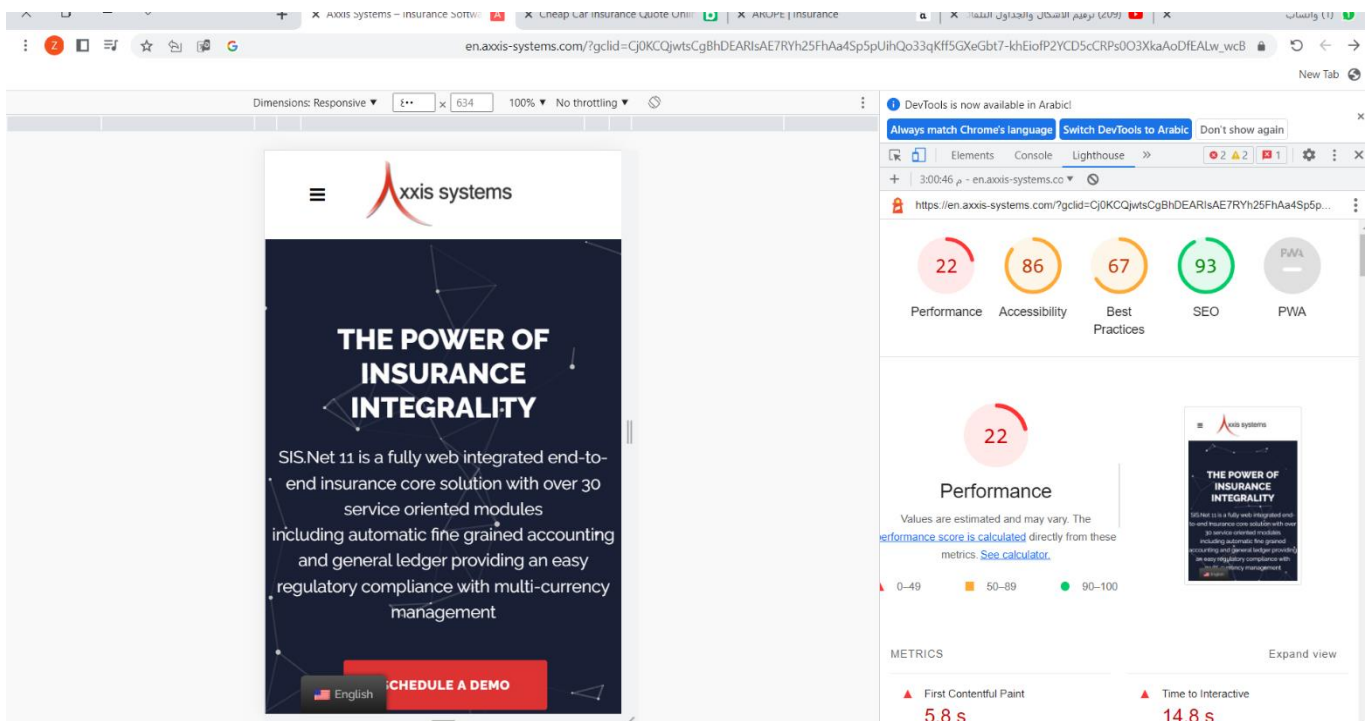


Figure (6.38): Testing Result – TLDCRM

### 6.1.3.4 Management Insurance System:

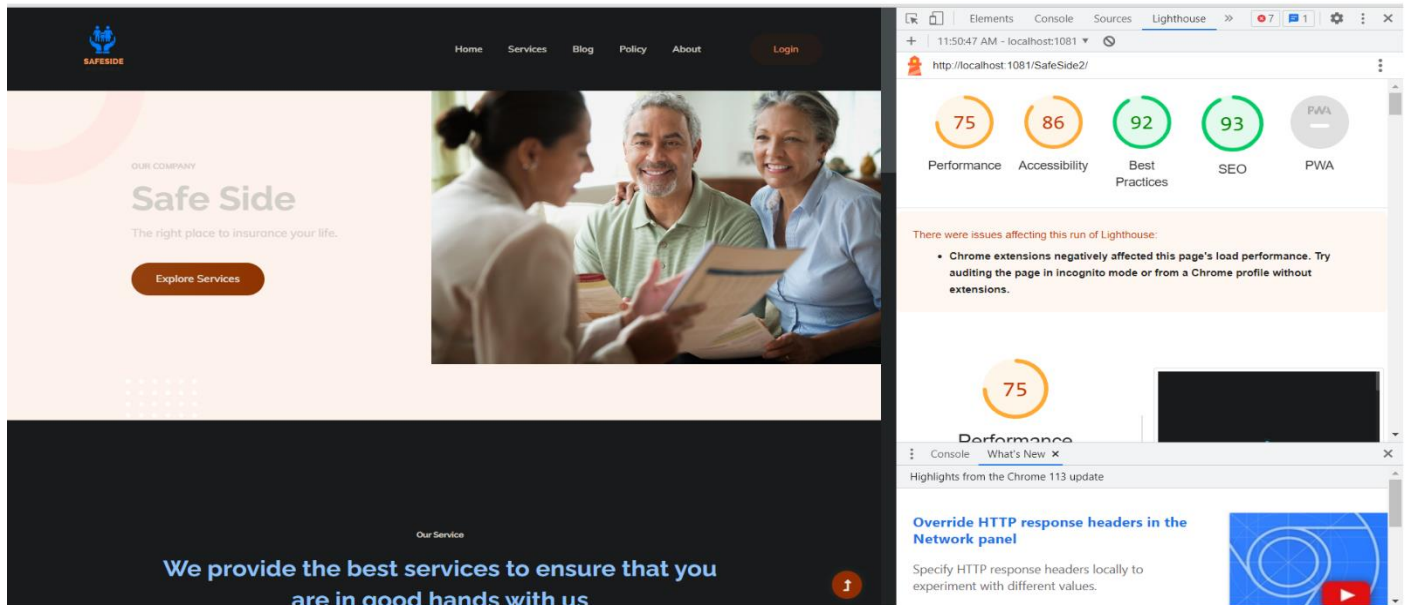


Figure (6.39): Testing Result – Management Insurance System

# **Chapter Seven**

## **Consultation and Future Prospects**

## **7 Conclusion and Future Prospects**

### **7.1 Conclusion :**

Nowadays, the traditional ordering methods are dying out. Its new era where technology dominates human life. With technological software and hardware, exceptions are reduced and even terminated. People also prefer an easy, fast and safe way for every part of their life. This project is designed to meet the requirements of the insurance subscription system. In our project: With this insurance company management system; Insurance companies can be satisfied convenient facilities for its customers. Through this platform we developed, we hope to reduce waste of time, avoid misunderstanding, provide easy data flow, satisfy customers, and reduce hard work. We believe we have achieved our goals and are satisfied with the code we have developed.

### **7.2 Future prospects**

1. Develop the Shape of the System Interfaces, Which Makes the System More Efficient and Effective.
2. Application Development on Android Devices.
3. Application Development on IOS Devices.
4. Develop Message System Between Customer & Admin.
5. Add more types of insurance to the project and expansion.

## **8 Chapter Eight References**



1. <https://www.nasasoft.com/insurancemanagement-systems-ultimate-guide>
2. <https://www.bindhq.com/>
3. [https://islandpress.org/books/understanding-disaster-insurance?gclid=Cj0KCQjwk7ugBhDIARIsAGuvGPaM5zcDZsT2uAEed6DEzbL8Lml4vGfXq44nLbZ1A3b2E92VuUdyCPJ0aAscxEALw\\_wcB](https://islandpress.org/books/understanding-disaster-insurance?gclid=Cj0KCQjwk7ugBhDIARIsAGuvGPaM5zcDZsT2uAEed6DEzbL8Lml4vGfXq44nLbZ1A3b2E92VuUdyCPJ0aAscxEALw_wcB)
4. <https://www.samsolutions.com/blog/insurance-agency-management-software>
5. <https://www.ventivtech.com/products/insurance-policy-management-system>
6. <https://www.insurancethoughtleadership.com/agent-broker/5-must-haves-agency-management-systems>
7. <https://www.tokiomarine.com.eg/ar/insurance-benefits/>
8. <https://www.faydety.com/learn/>
9. <http://www.aropesyria.com/index-ar.html>
10. <https://app.diagrams.net/>
11. <https://www.theoperationsmanagement.com/customer-support-3611>

<https://www.theoperationsmanagement.com/types-of-e-services-3599>