A

Project Report

On

Online Insurance Portal

Submitted by

Omkar Jeevan More

Roll No.: 22137

MCA–I

SEM–II

Under the guidance of

Dr.Ramesh Jadhav

For the Academic Year 2022-23

Sinhgad Technical Education Society's

Sinhgad Institute of Management

Vadgaon Bk Pune 411041

(Affiliated to SPPU Pune & Approved by AICTE New Delhi)

Sinhgad Institutes

SINHGAD TECHNICAL EDUCATION SOCIETY'S

SINHGAD INSTITUTE OF MANAGEMENT®

Affiliated to Savitribai Phule Pune University, Approved by AICTE, New Delhi

Prof. M. N. Navale	Dr. (Mrs.) Sunanda M. Navale	Dr. Chandrani Singh
M.E. (ELECT.), MIE, MBA	B.A., MPM, Ph.D	MCA, ME, (Com. Sci.), Ph.D
FOUNDER PRESIDENT	FOUNDER SECRETARY	DIRECTOR - MCA

Date:

CERTIFICATE

This is to certify that Mr. Omkar Jeevan More has successfully completed his/ her project work entitled **"Online Insurance Portal"** in partial fulfillment of MCA – I SEM –II Mini Project for the year 2022-2023. He/ She has worked under our guidance and direction.

Dr.Ramesh Jadhav	Dr. Chandrani Singh
Project Guide	**Director, SIOM-MCA**

Examiner 1	Examiner 2

Date:

Place: Pune

Celebrating 25 Years OF ACADEMIC EXCELLENCE

DECLARATION

I certify that the work contained in this report is original and has been done by meunder the guidance of my supervisor(s).

- The work has not been submitted to any other Institute for any degree or diploma.
- I have followed the guidelines provided by the Institute in preparing the report.
- I have conformed to the norms and guidelines given in the Ethical Code ofConduct of the Institute.
- Whenever I have used materials (data, theoretical analysis, figures, and text) fromother sources, I have given due credit to them by citing them in the text of the report and giving their details in the references.

Name and Signature of Project Team Members*:*

Sr. No.	Seat No.	Name of students	Signature of students
1	**188**	**Omkar Jeevan More**	

ACKNOWLEDGEMENT

It is very difficult task to acknowledge all those who have been of tremendous help in this project. I would like to thank my respected guide **Dr.Ramesh Jadhav** for providing me necessary facilities to complete my project and also for their guidance and encouragement in completing my project successfully without which it wouldn’t be possible. I wish to convey my special thanks and immeasurable feelings of gratitude towards **Dr. Chandrani Singh, Director SIOM-MCA.** I wish to convey my special thanks to all teaching and non-teaching staff members of **Sinhgad Institute of Management, Pune** for their support.

Thank You

Yours Sincerely,

Omkar Jeevan More

INDEX

CHAPTER 1: INTRODUCTION

This is an exciting and challenging time for insurance , Customer behavior is changing rapidly. Technology and in particular the growth of online and social media, is driving a fundamental shift in customer expectations in terms of how products are marketed, priced, sold and serviced, and how companies are perceived. Pure internet businesses have set new standards for customer-centricity and engagement that raise the performance bar for players in every retail business sector So to fulfil the requirements and need of the insurer we have developed one methodology which will attract the people. And the system will be a user friendly so all peoples who have knowledge about insurance or not they all will accept the insurance.

1.1.Abstract

The main purpose of this project is to estimate some important statistics of Micro insurance in the current market of India by gathering and analyzing data which is available. The statistics will be useful to prepare better insurance plans to boost up the Life Insurance industry in India.The Online Insurance System is a web-based application developed to streamline the insurance purchasing and claim processing procedures. It provides users with a convenient platform to browse and purchase insurance policies online, eliminating the need for physical paperwork and reducing the overall processing time. This project aims to leverage the power of Python programming language to create an efficient and user-friendly system.

1.2. Existing System and Need for System :

Existing System: In the traditional insurance industry, purchasing insurance policies and processing claims typically involve cumbersome paperwork, manual processes, and long turnaround times. Customers often have to visit insurance offices in person, fill out numerous forms, submit physical documents, and wait for manual verification and processing. This approach is time-consuming, inefficient, and prone to errors. It creates inconvenience for customers and leads to delays in policy issuance and claim settlements.

Need for an Online Insurance System:

1.Convenience: The modern era demands convenience and efficiency in all aspects of life, including insurance. An online insurance system provides users with the convenience of browsing and purchasing policies from anywhere, at any time. It eliminates the need for physical paperwork, visits to insurance offices, and reduces the overall processing time.

2.Time Efficiency: Online insurance systems significantly reduce the time required for policy issuance and claim processing. Automation of various processes, such as policy generation, premium calculations, and claim verification, accelerates the entire workflow. Users can quickly access policy information, make payments online, and receive instant policy documents.

1.3.Scope of System :

following are the key components and functionalities within the scope of the project:

1.User Registration and Authentication:

- Allow users to create accounts and provide necessary personal information.

- Implement a secure authentication mechanism to ensure authorized access to the system.

2.Insurance Policy Management:

- Display a catalog of available insurance policies, including health, automobile, property, and life insurance.

- Enable users to browse and compare different policy options based on their requirements.

- Provide detailed policy information such as coverage, premium amounts, and terms and conditions.

3.Online Policy Purchase:

- Allow users to select desired insurance policies and customize coverage options if available.

- Provide a secure online payment gateway for users to make premium payments.

1.4. Operating Environment Hardware and Software

HARDWARE SPECIFICATION –

Processor: Any Processor above 1 GH

RAM: Minimum 2GB.

Hard Disk: Minimum 500 GB.

Front End: HTML,CSS

Middle Layer: Python

Back-End:SQL-Lite

Software: Vs Code

Operating System: Windows Family

1.5.Brief Description of Technology used :

Python's extensive ecosystem and the availability of numerous libraries and frameworks make it a popular choice for web application development. The combination of Python, Django/Flask frameworks, and additional libraries provides a solid foundation for building an efficient and scalable online insurance system.

Python Programming Language:

Python is a versatile and widely-used programming language known for its simplicity and readability.It provides a large number of libraries and frameworks that facilitate rapid development and ease of maintenance.Python's syntax and structure make it suitable for both small and large-scale projects

Django Framework:

Django is a high-level web framework written in Python, designed for efficient web development.It follows the Model-View-Controller (MVC) architectural pattern, providing a structured approach to building web applications.Django offers built-in features such as URL routing, form handling, database abstraction, and user authentication.It promotes code reusability, scalability, and security.

CHAPTER 2: PROPOSED SYSTEM

2.1. Feasibility Study :

1. Technical Feasibility:

   - Availability of Required Technology: The project relies on established technologies such as Python, web frameworks (Django/Flask), and database systems (e.g., MySQL, PostgreSQL). These technologies have extensive documentation, active communities, and readily available resources, making them technically feasible for the project.

2. Economic Feasibility:

   - Cost-Benefit Analysis: Evaluate the potential costs associated with the project, including development resources, infrastructure, licensing fees, and ongoing maintenance. Compare these costs against the anticipated benefits, such as increased operational efficiency, reduced administrative costs, and improved customer satisfaction.

3. Operational Feasibility:

   - User Acceptance: Conduct surveys or interviews with potential users (policyholders, insurance agents, administrators) to gather feedback on the proposed online insurance system. Assess their willingness to adopt and use the system and identify any specific operational requirements or challenges.

4. Legal and Regulatory Feasibility:

   - Compliance: Evaluate the system's compliance with relevant legal and regulatory requirements, such as data protection laws (e.g., GDPR), insurance industry regulations, and security standards.
   - Privacy and Security: Ensure that the system adheres to data privacy regulations and implements robust security measures to protect sensitive user information.

5. Schedule Feasibility:

   - Project Timeline: Assess the proposed project timeline and evaluate whether it is realistic and achievable. Consider the availability of resources, potential risks, and dependencies on external factors.
   - Project Management: Evaluate the project management approach, including the allocation of tasks, monitoring progress, and mitigating risks to ensure timely project completion.

2.2. Objectives of the proposed system :

1. To Automate Insurance Processes: The primary objective of the online insurance system is to automate various insurance processes, reducing manual effort, paperwork, and processing time. The system aims to streamline policy issuance, premium calculations, claim submissions, and processing, resulting in increased efficiency and accuracy.
2. To Enhance User Experience: The proposed system aims to provide a user-friendly interface that enhances the overall user experience. It should be intuitive, easy to navigate, and provide clear and concise information about insurance policies, coverage options, and claim procedures. The system should empower users to make informed decisions and conveniently manage their insurance policies.
3. Enable Online Policy Purchases: The system aims to provide users with the ability to browse, select, and purchase insurance policies online. It should facilitate secure online payments, generate policy documents electronically, and provide immediate policy coverage. This objective aims to provide users with convenience, flexibility, and a seamless policy purchasing experience.
4. Streamline Claim Processing: The proposed system intends to simplify the claim submission and processing procedures. Users should be able to submit claims online, attach relevant supporting documents electronically, and track the progress of their claims. The system should automate claim verification, streamline communication between policyholders and insurance agents, and expedite the claim settlement process.

2.3. Users of the system

Admin:

1.Dashboard: In this section admin can see all detail in brief.

2.Insurance Category: In this section admin can manage categories (add and update).

3.Insurance Subcategory: In this section admin can manage sub category (add and update).

4.Insurance Policy: In this section admin can manage insurance policy (add and update policy).

5.User Detail: In this section admin can manage all user detail.

6. Policy Holders: In this section admin can manage all insured policy. Admin can view policy on the basis of status (pending policy, approved policy and disapproved policy). Admin also can approve pending policy.

7. Tickets: In this section admin can view detail of issues raised by user and can also update remark on particular tickets.

Admin can also update his profile and change his password.

User/Customer :

In this module there is Two Section

1.Insurance : In this section user can apply for policy and check his policy is approve or reject(which is done by admin).

2.Policy: In this section user can raise ticket against any complain and see status of his/her policy , User can also update his/ her profile, change password and recover password.

3. Individuals or organizations who purchase insurance policies through the online system.

4.They can browse and select insurance policies, make premium payments, manage their policies, and submit claims.

CHAPTER 3: ANALYSIS AND DESIGN

3.1.Title of Project

- Online Insurance Portal

3.2. System Requirements (Functional & Non Functional)

1.Functional requirements :

Functional requirements drive the application architecture of a system. Functional requirementsare the intended behaviors of the system. This behavior may be expressed as services, tasks orfunctions that the system is required to perform. The functional requirements that the proposedsystem Perform.

-Registration of the new insurance

-Payment management

-Generating report

-Delete report

-View report

2.Non-functional requirements :

Non-functional requirement is that describes about how the system will do the functional requirements. It describes performance, maintainability, security, usability, availability, accuracy, and reliability of proposed.

system.Performance: The proposed system perform its operations within a minimum amount of time and the user gets the expected result within a few seconds and the system is effective.

Accessibility: The system can be accessible based on the accessible privilege or based on autontication.

Accuracy: proposed system will reduce error because all operation can be check correctly and validate that whatever information is coming from the data base and input to the database.

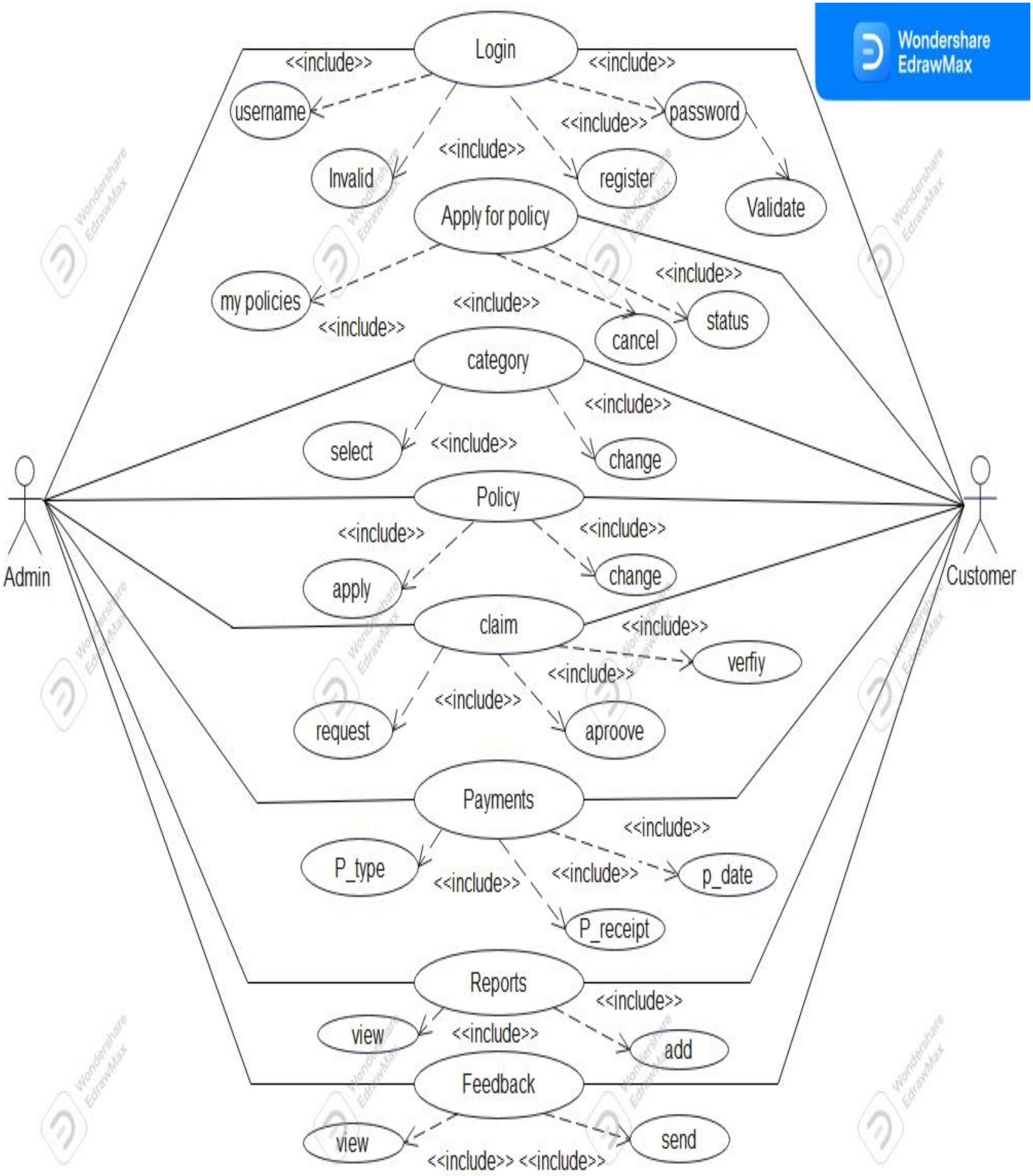
Maintenance: our system can be easy maintainable and updateable, if the system get any failure.

Extensibility: - Adding features and carry-forward of customizations at next major version upgrade

along with the business re-engineering is possible.

3.3.Use/System Case Diagrams

3.3.1.Use Case Diagram of User



Use Case Name:	System
Description:	User and Admin do the activities e.g. Login Management, User profile management, Profile Management,Category selection,etc.
Actor :	User and Admin
Pre Condition:	The user must be logged-In.
Main Flow:	1: Login Management. 2: profile management. 3: Category. 4: Claim management. 5.payment management.
Relation Ship:	Include: Update Include: delete Include: Payments Details. Include: Report Details.
Post Condition:	A particular user will be logged-In. after do the different activities

3.3.1: Login Management.

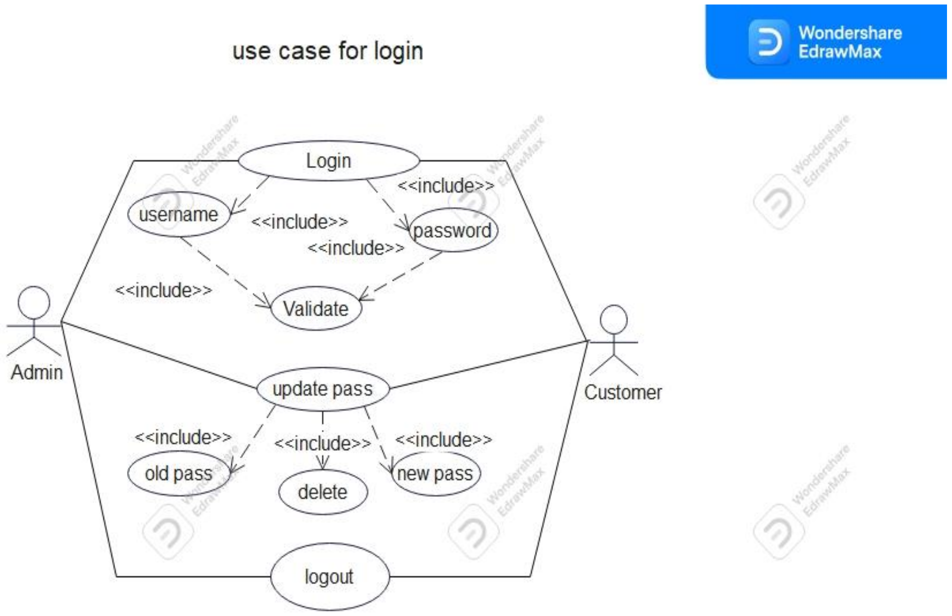
3.3.2: Registration profile management.

3.3.3: Dashboard management.

3.3.6: Report management.

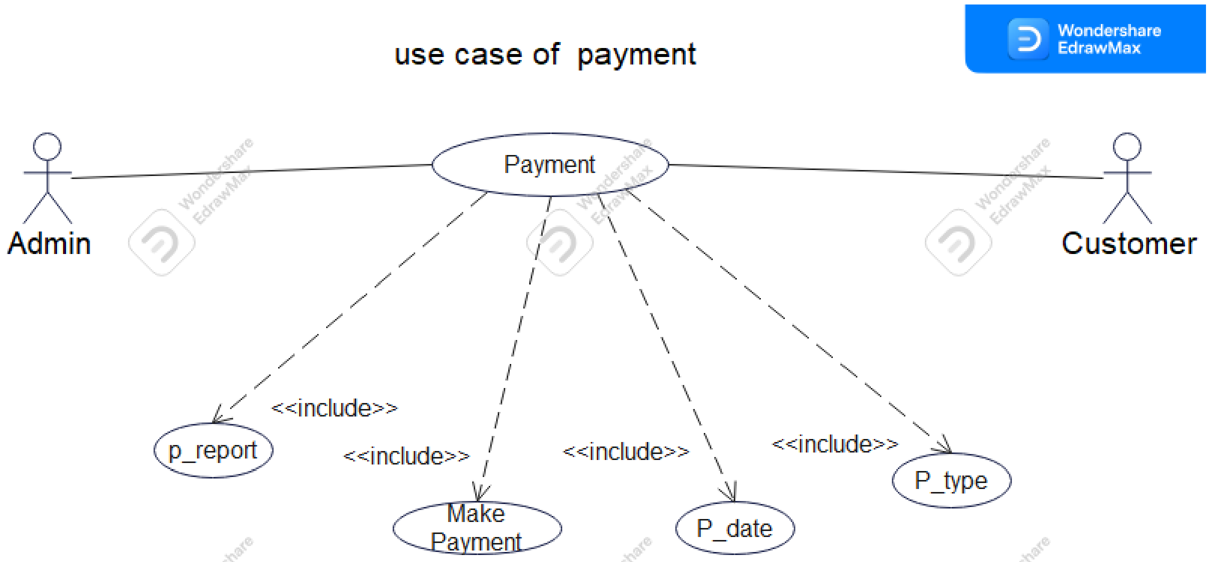
3.3.7: Feedback management.

3.3.2.Use case for login



Use Case Name:	Login
Description:	Admin or User View the login details.
Actor :	Admin , User
Pre Condition:	The user must be logged-In.
Main Flow:	1: User Register the new login Details. 2: Admin allocates policies in system/website. 3: User View the policy details. 4: Admin update the policy Details.
Relation Ship:	Include: User Details. Include: Policy Details Include : update Include : delete
Post Condition:	A particular User will be checked login details.

3.3.3.use case for payment



Use Case Name:	Payment
Description:	Admin or User View the payment details.
Actor :	Admin , User
Pre Condition:	The user must be logged-In.
Main Flow:	1: User Register the new login Details. 2: Admin allocates policy wise payment in system/website. 3: User View the payment details. 4: payment management
Relation Ship:	Include: User Details. Include: Payment Details Include: update Include: delete
Post Condition:	A particular User will be checked payment details.

3.4.Activity Diagram

3.4.1 Activity Diagram for login page

3.4.1 Login Management.

3.4.1.1.Check user Login.

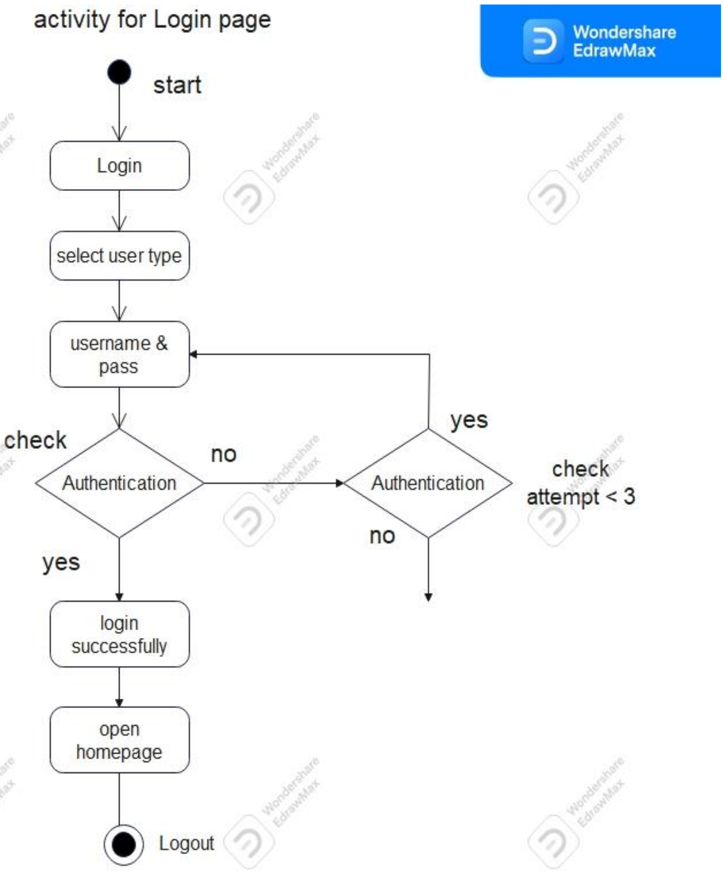
3.4.2Profile Management

3.4.2.1.Admin Registers the User Details.

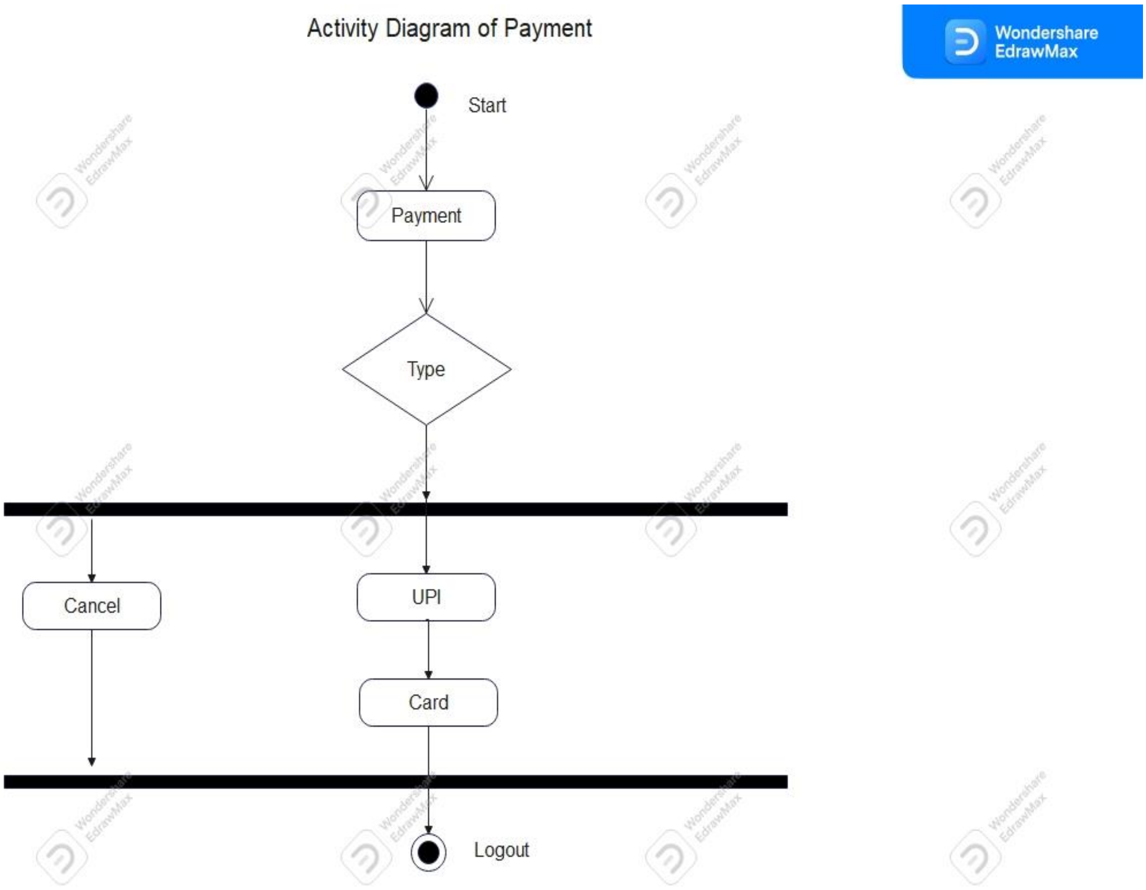
3.4.2.2.User View Self profile.

3.4.4 User can update details.

3.4.4.1.user edit details



3.4.2.Activity diagram of payment



3.4.3.Activity diagram of user

3.4.3 Login Management.

3.4.3.1 User can Login**.**

3.4.4 Registration Management

3.4.4.1 User can register details**.**

3.4.4.2 User view self profile.

3.4.5 Dashboard Management

3.4.5.1 User can view dashboard.

3.4.6 Claim Management.

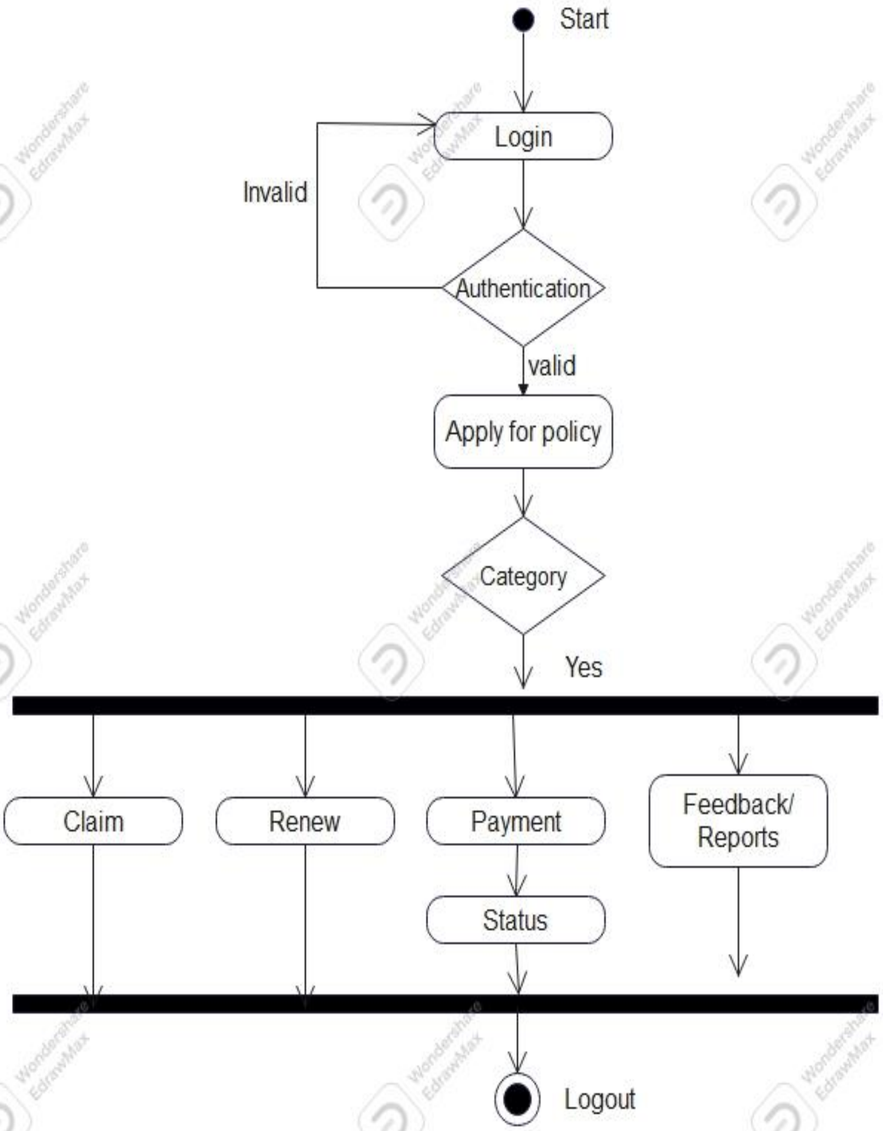
3.4.6.1 Apply and view

3.4.7 Feedback Management

3.4.7.1 Give Feedback.

3.4.7.2 View Feedback**.**

Activity Diagram for User/Customer



3.5 .Sequence Diagrams

3.5.1 Login Management.

3.5.1 User can Login**.**

3.5.2 Registration Management

3.5.2.1 User can register details**.**

3.5.2.2 User view self profile.

3.5.3 Dashboard Management

3.5.3.1 User can view dashboard.

3.5.4 Report Management.

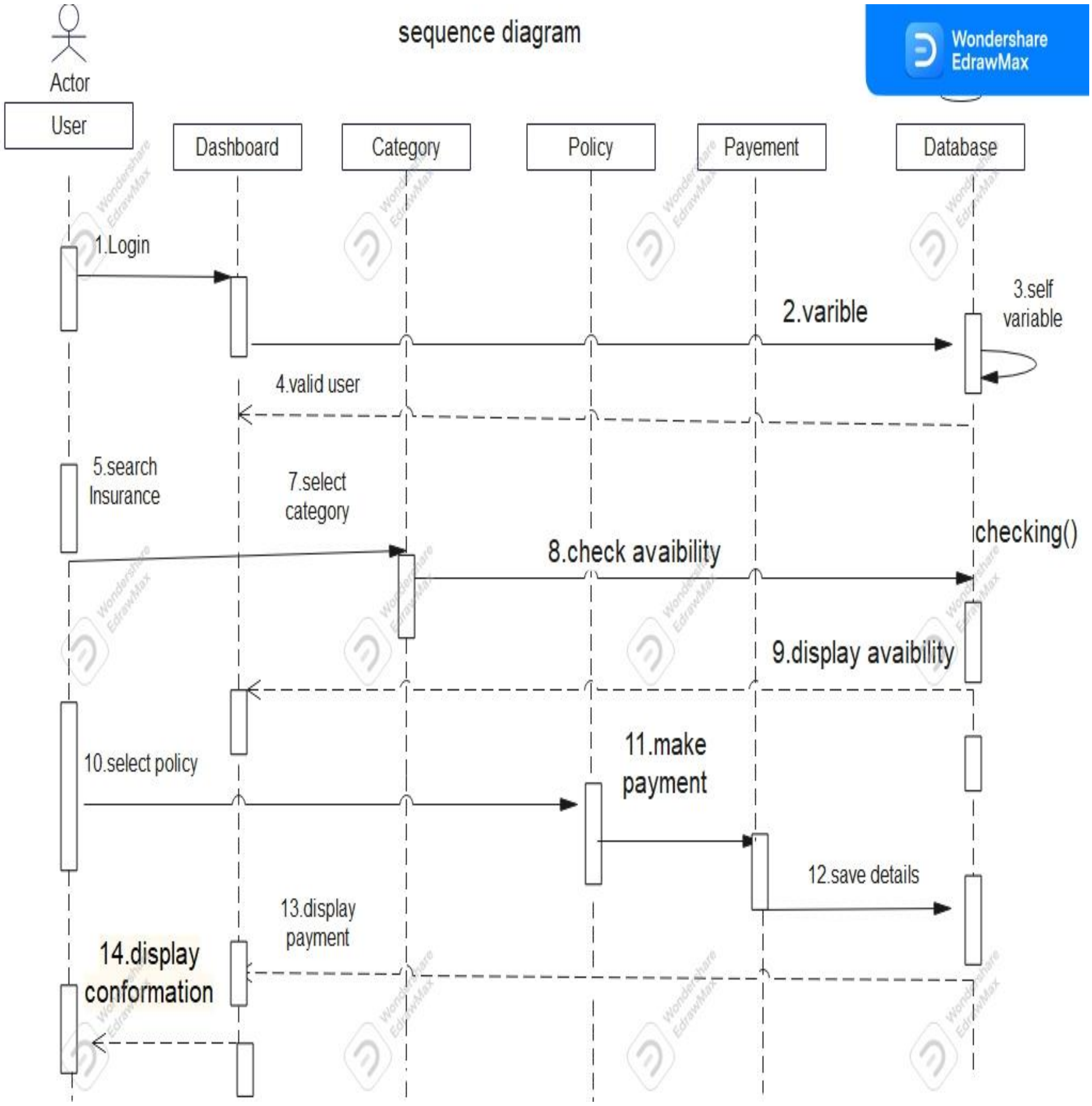
3.5.6.1 Admin view.

3.5.5 Feedback Management

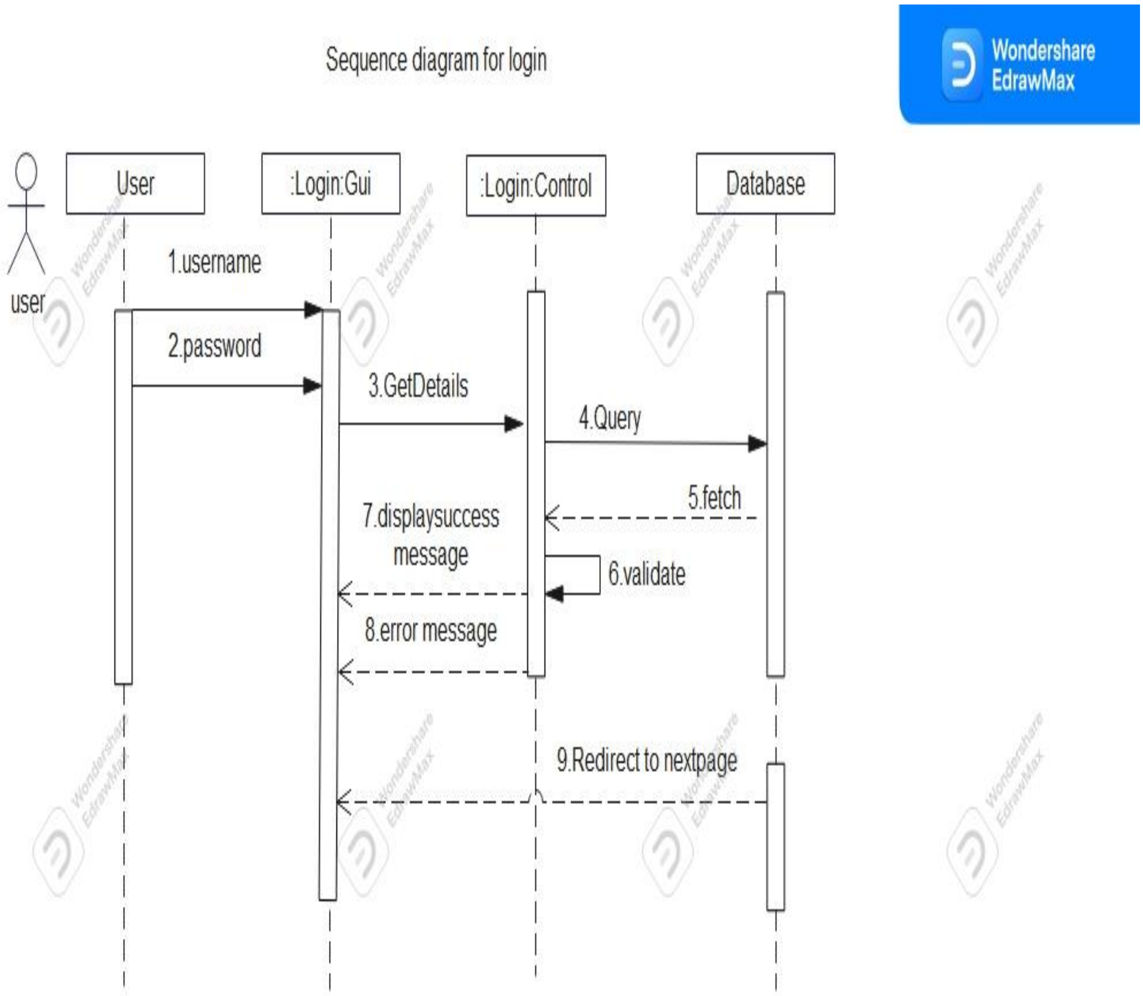
3.5.5.1 Give Feedback.

3.5.5.2 View Feedback**.**

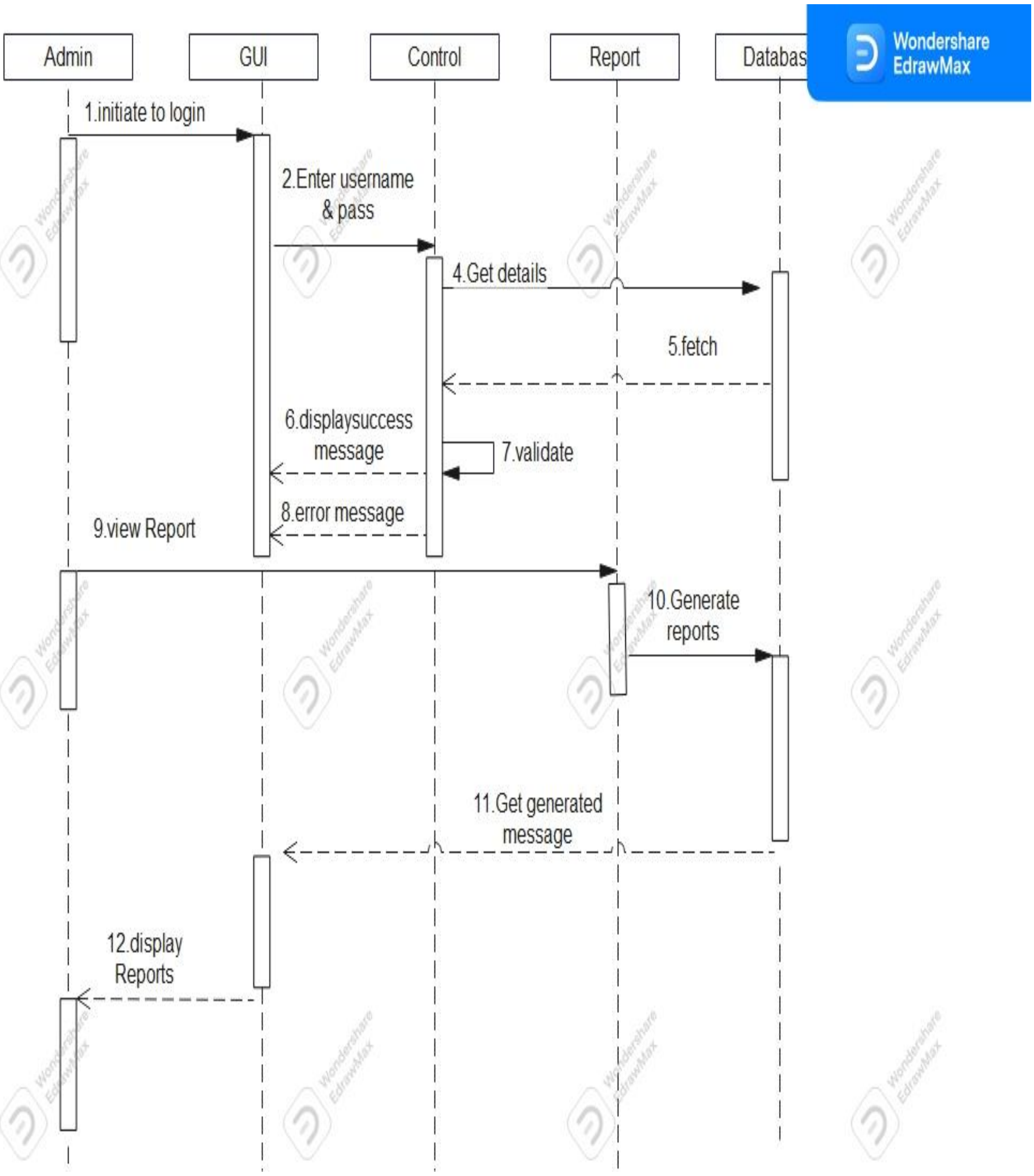
3.5.1.Sequence Diagram for User



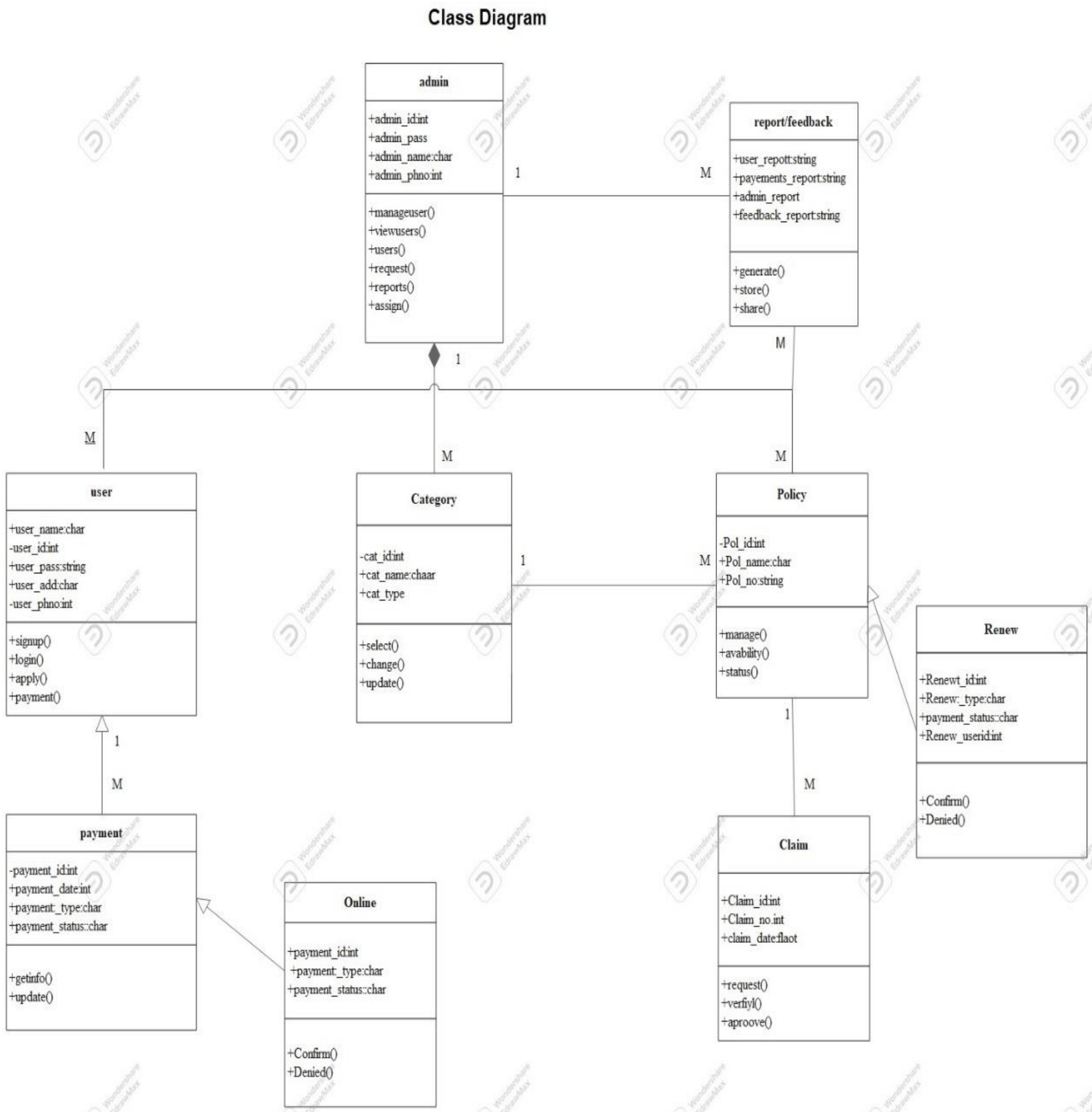
3.5.2.Sequence diagram for login page



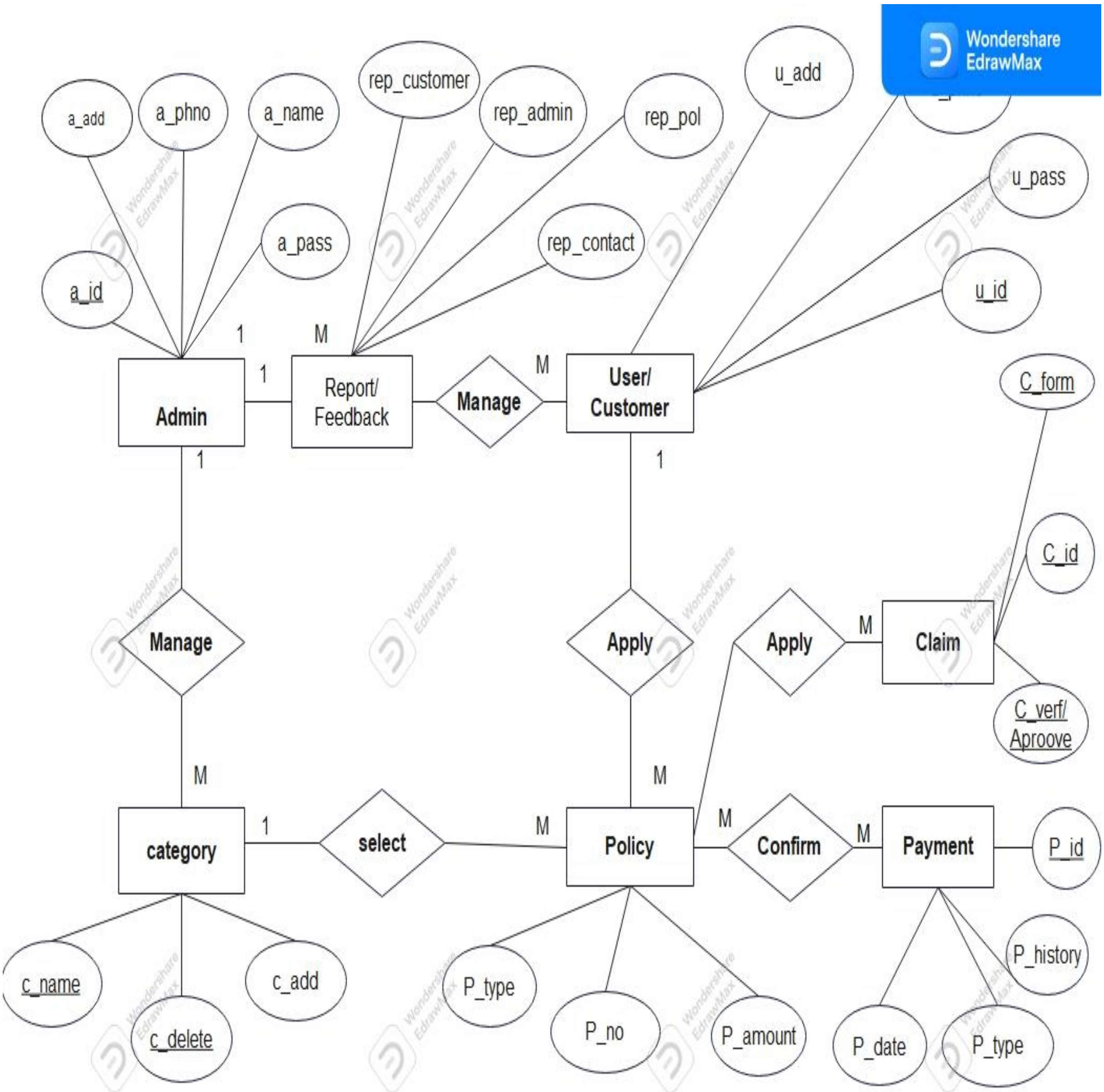
3.5.3.Sequence diagram for Admin



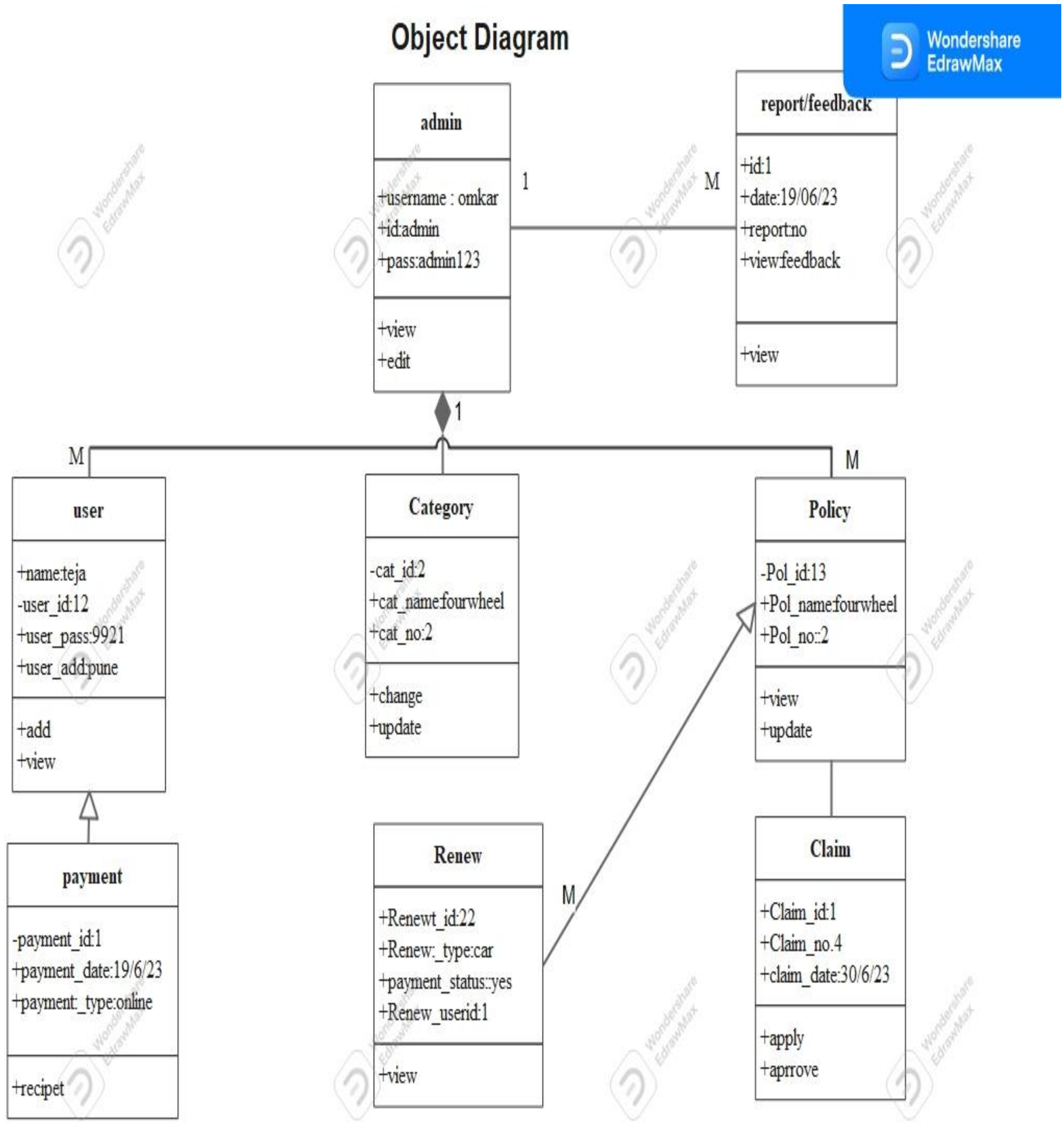
3.6.Class Diagram



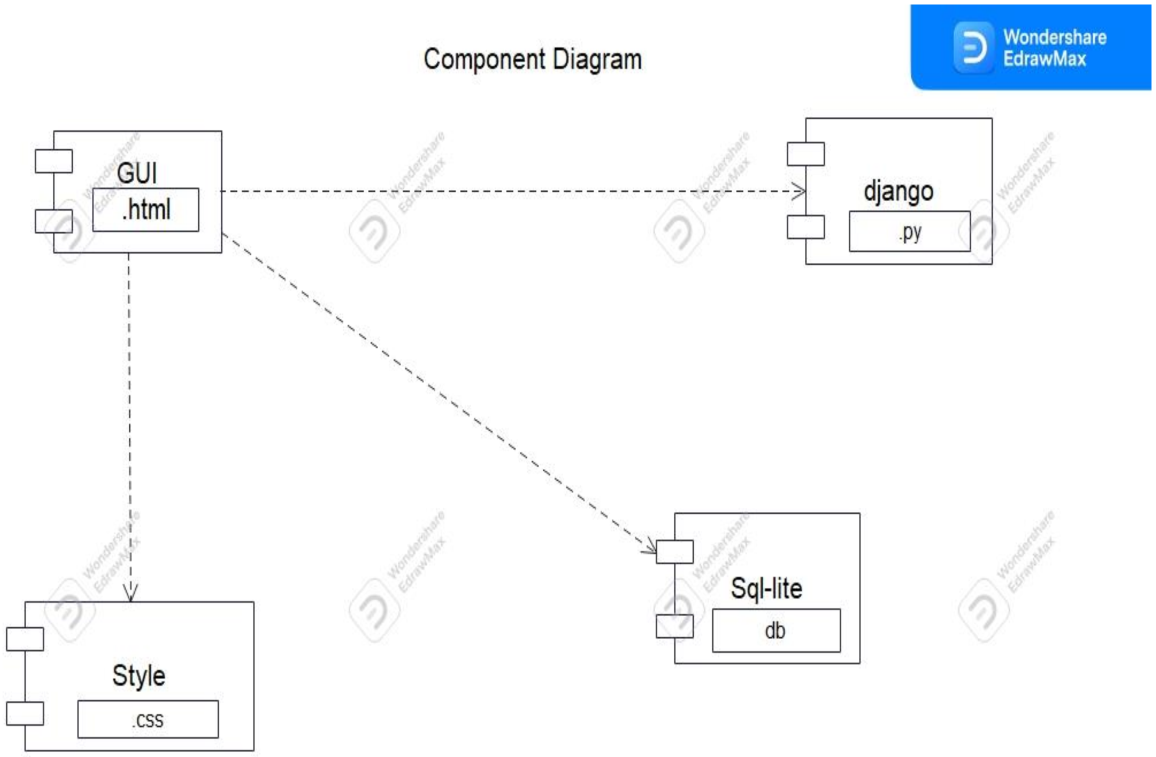
3.7.ERD Diagram



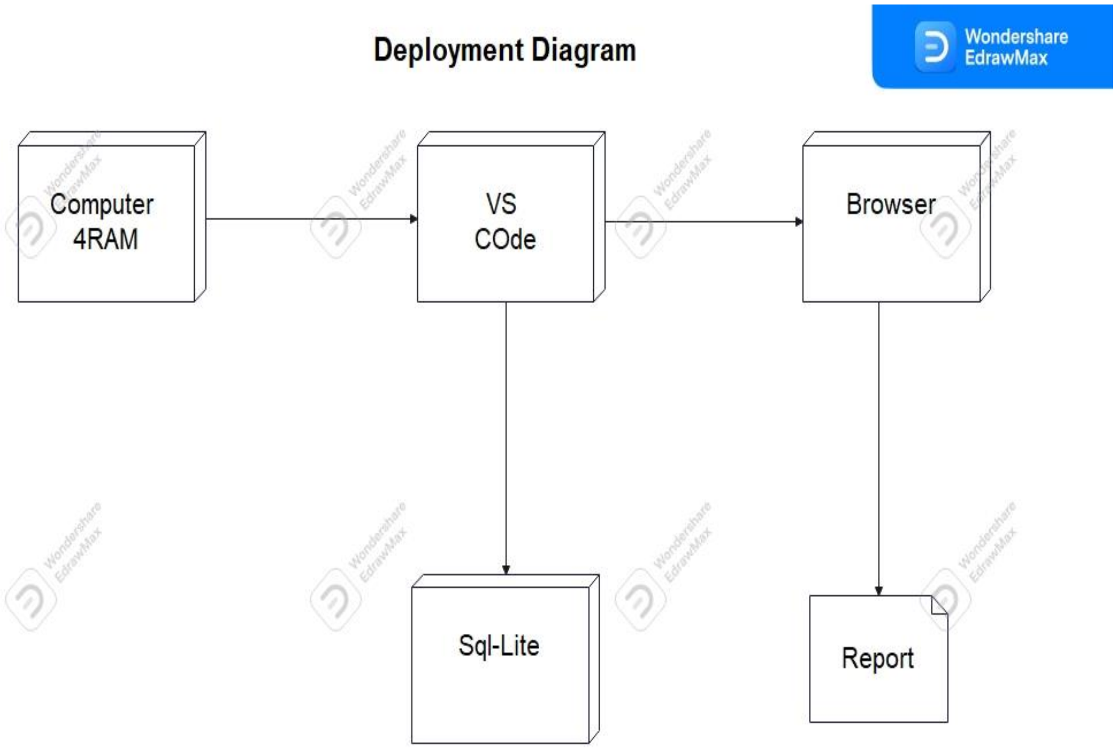
3.8.Object Diagram



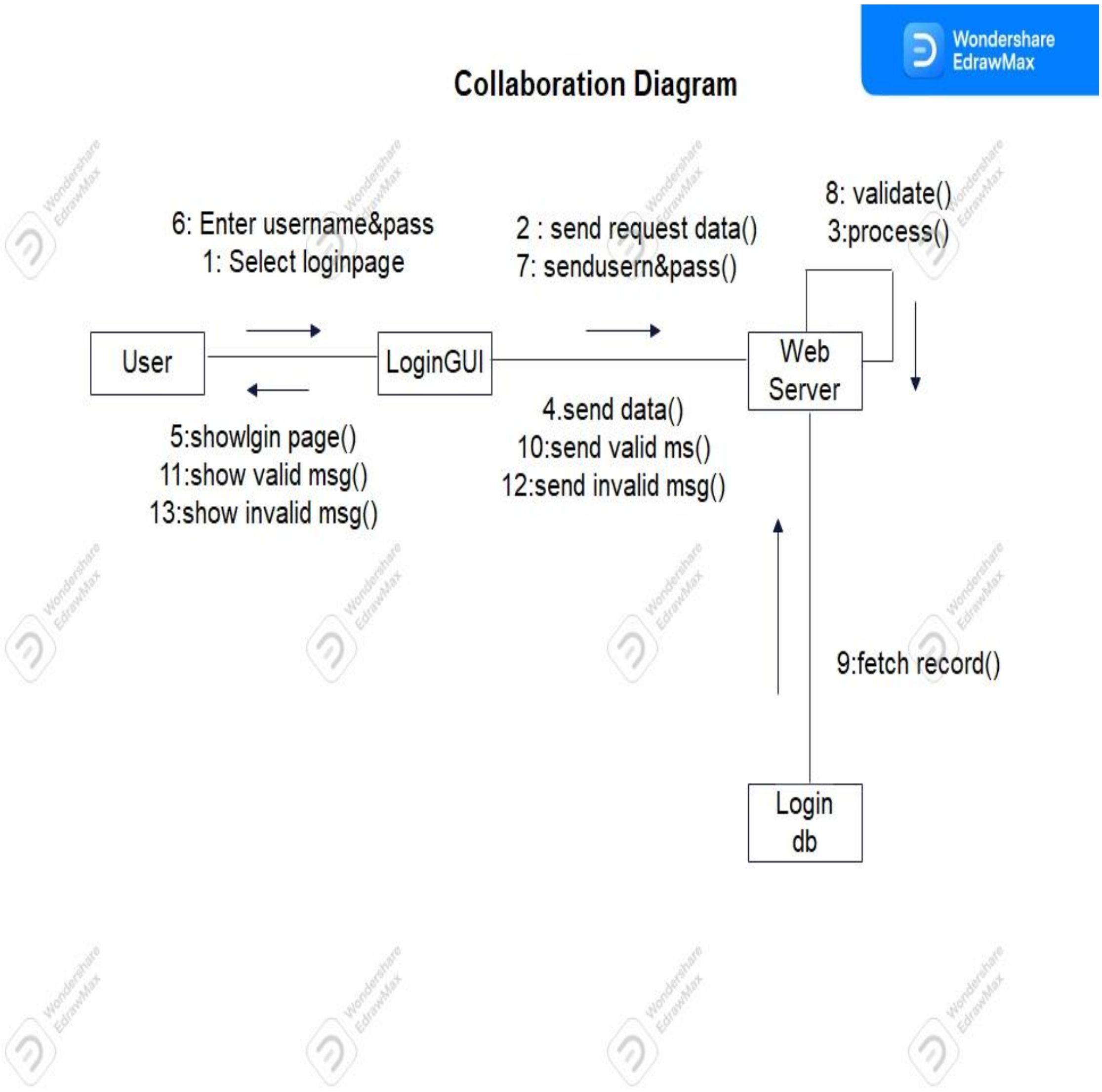
3.9. Component Diagram



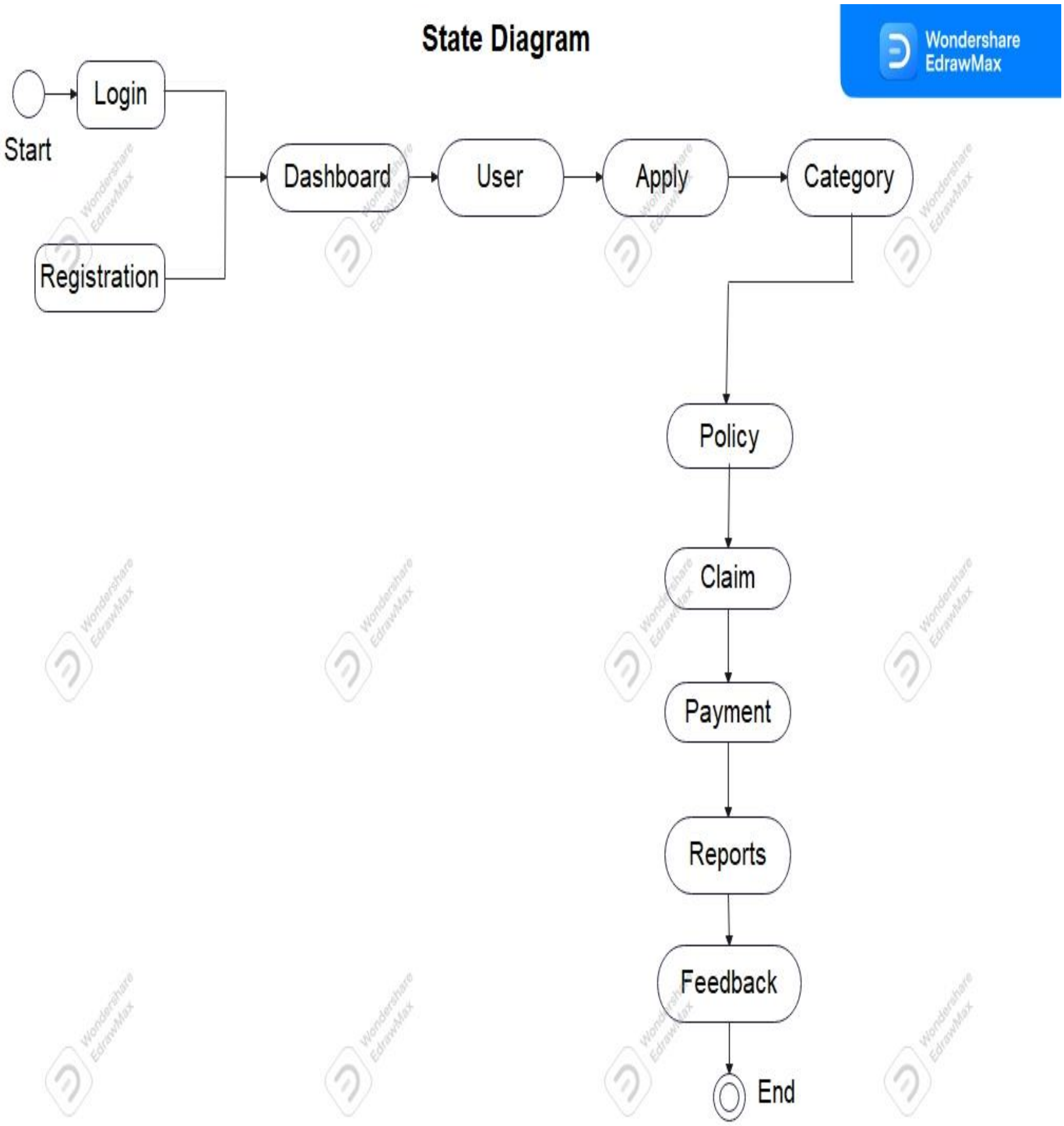
3.10.Deployment Digram



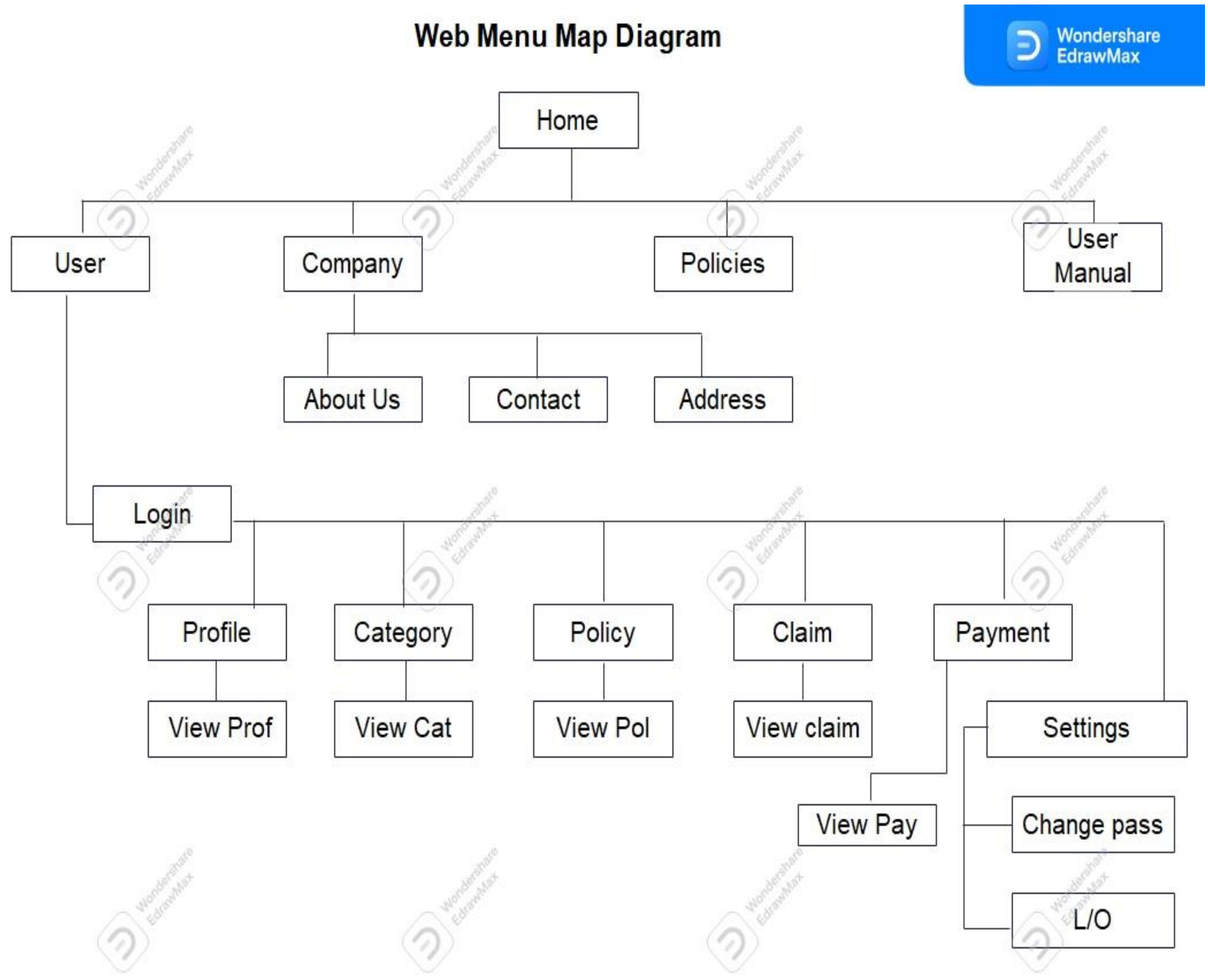
3.11.Collaboration Diagram



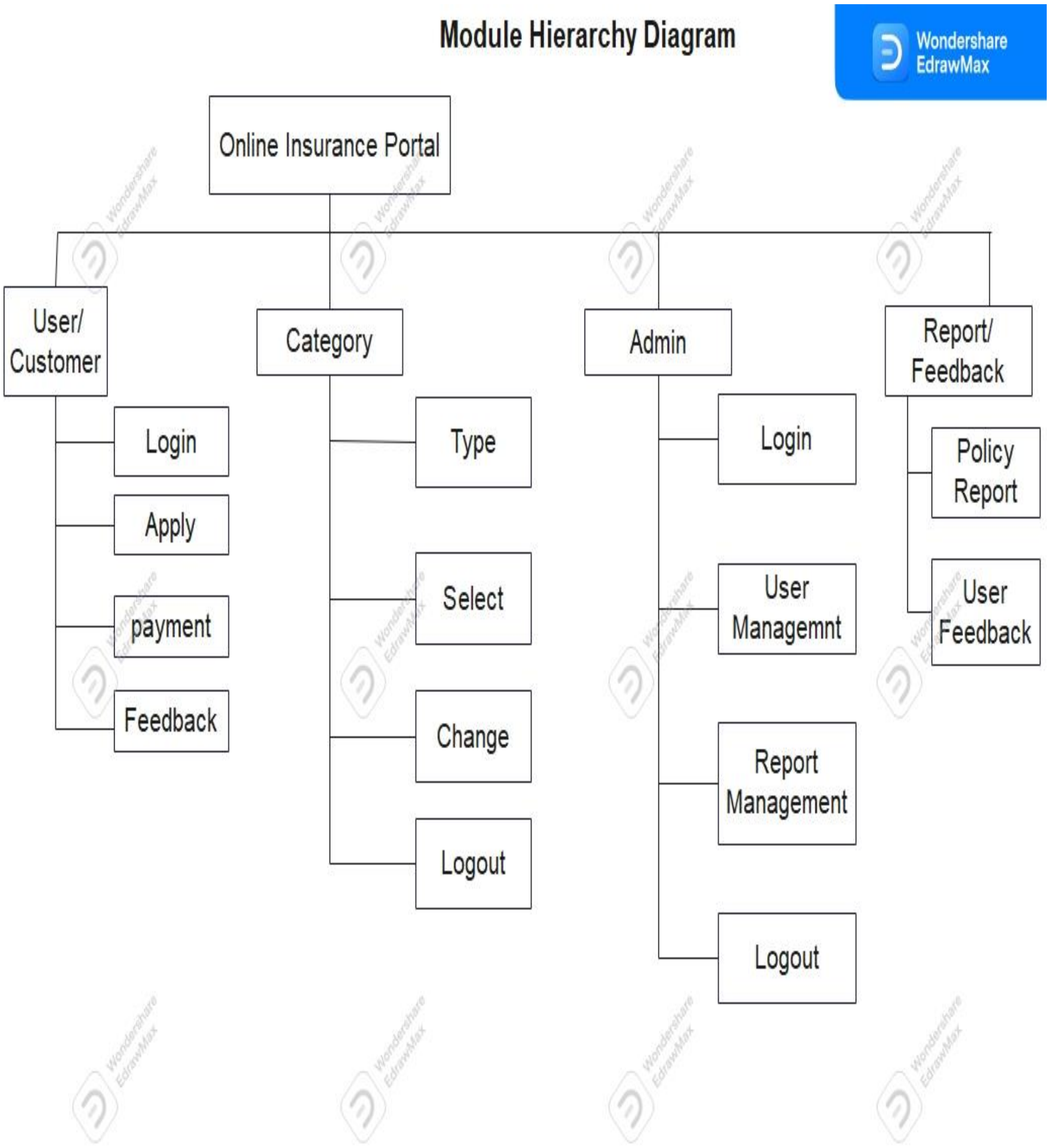
3.12.State Transaction Diagram



3.13.Web Menu Map Diagram



3.14. Module Hierarchy Diagram



3.15.Table Design/Structure

3.15.1 User Table

Table Name		User_table		
Description		This table contains the details of Users		
Constraints		Primary key (U_ID).		
Field name	**Field type**	**Field Size**	**Constraints**	**Description**
U_id	int	5	Primary Key	It represent umique identification
U_name	varchar	10	Not Null	It represent name of user
U_email	varchar	15	Not Null	It represent mail of user
U_pass	varchar	10	Not Null	It represent pass of user
U_address	varchar	20	Not Null	It represent address
U_mobile	int	10	Not Null	It represent mobile of user

3.15.2 Admin Table :

Table Name	Admin_table
Description	This table contains the details of admin
Constraints	Primary key (a_ID) Foreign key(u_ID)

Field name	**Field type**	**Field Size**	**Constraints**	**Description**
a_id	int	5	Primar Key	It represent unique identification
a_username	varchar	10	Not Null	User name of admin
a_email	varchar	15	Not Null	It represent mail of admin
a_password	varchar	10	Not Null	It represent pass of admin
U_id	int	20	Foreign Key	Unique identification of user

3.15.3 Migraints Table :

Table Name	migraints_table
Description	This table contains the details of migraints
Constraints	Primary key (m_ID).

Field name	**Data type**	**Field Size**	**Constraints**	**Description**
m_id	int	5	Primary Key	It represent id of migrants
m_name	varchar	10	Not Null	It represent name of migrants
m_applied_date	date	10	Not Null	It shows Applied date

3.15.4 Payment Table :

Table Name	payment_table
Description	This table contains the details of payment
Constraints	Primary key (P_ID) Foreign key(U_ID)

Field name	**Data type**	**Field Size**	**Constraints**	**Description**
p_id	int	5	Primary Key	It represent id of payment
u_name	varchar	10	Foreign Key	It represent name of user
p_date_time	timestamp	10	Not Null	It shows date & time
P_type	varchar	10	Not Null	It shows type of payment

3.15.5 Feedback Table:

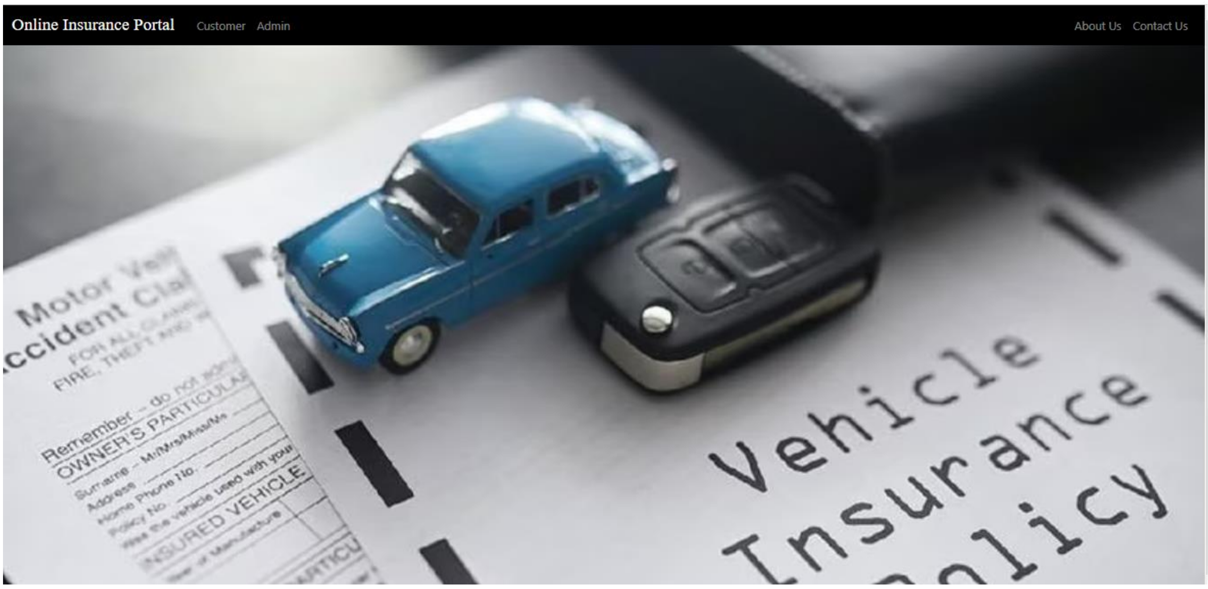
Table Name	feedback_table
Description	This table contains the details of users feedback
Constraints	Primary key (F_ID) Foreign key(U_ID)

Field name	**Data type**	**Field Size**	**Constraints**	**Description**
F_id	**int**	**10**	**Primary key**	**It represent no of feedback**
u_id	**varchar**	**15**	**Foreign Key**	**It represent id of user**
F_date_time	**timestamp**	**10**	**Not Null**	**It shows Applied date**
U_name	**varchar**	**10**	**Not Null**	**It represent user name**

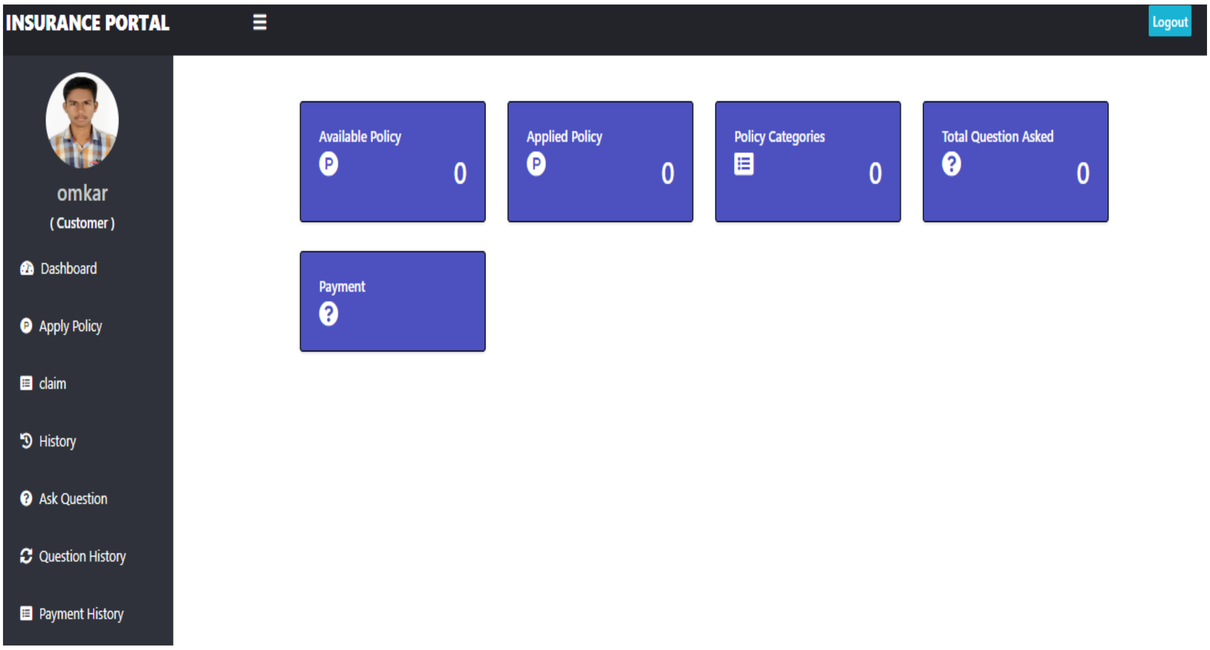
3.16.Data Dictionary

Field name	Constraints	Table Name	Referenced in table name	Description
a_id	Primary key	Admin_table	-----	It represent unique identification
a_username	Not Null	Admin_table	-----	User name of admin
a_email	Not Null	Admin_table	-----	It represent mail of admin
a_password	Not Null	Admin_table	-----	It represent pass of admin
F_no	Not Null	Feedback_table	-----	It represent no of feedback
F_date_time	Not Null	Feedback_table	-----	It shows applied date
m_id	Not Null	Migrants_table	-----	It represent id of migrants
m_name	Not Null	Migrants_table	-----	It represent name of migrants
m_applied_date	Not Null	Migrants_table	-----	It shows Applied date
p_id	Primary key	Payment_table	-----	It represent id of payment
p_date_time	Not Null	Payment_table	-----	It shows date & time
P_type	Not Null	Payment_table	-----	It shows type of payment
U_id	Primary key	User_table	Feedback_table/ admin_table	It represent unique identification
U_name	Not Null	User_table	Payment_table	It represent name of user
U_email	Not Null	User_table	-----	It represent mail of user
U_pass	Not Null	User_table	------	It represent pass of user
U_address	Not Null	User_table	------	It represent address
U_mobile	Not Null	User_table	-----	It represent mobile of user

3.17 Sample Input and Output Screens



Screen 1 : Home Page



Available Policy 0

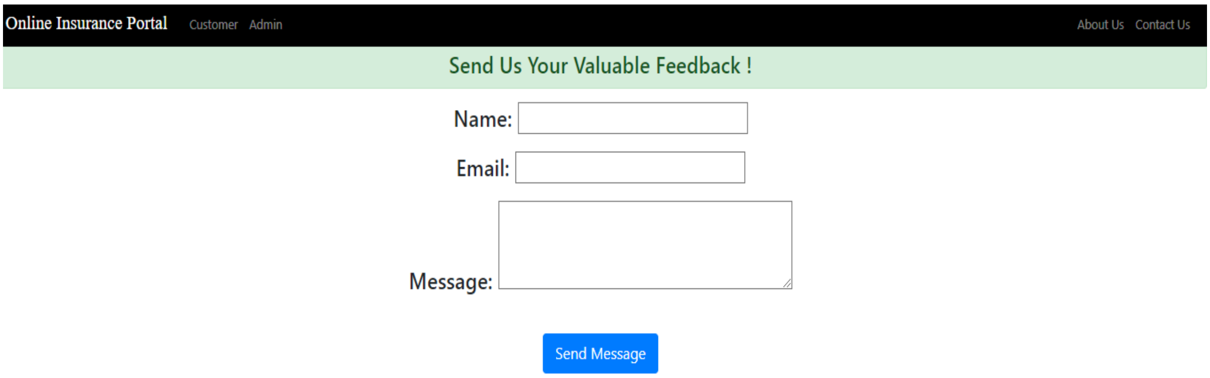
Applied Policy 0

Policy Categories 0

Total Question Asked 0

Payment

Screen 2 : User Page



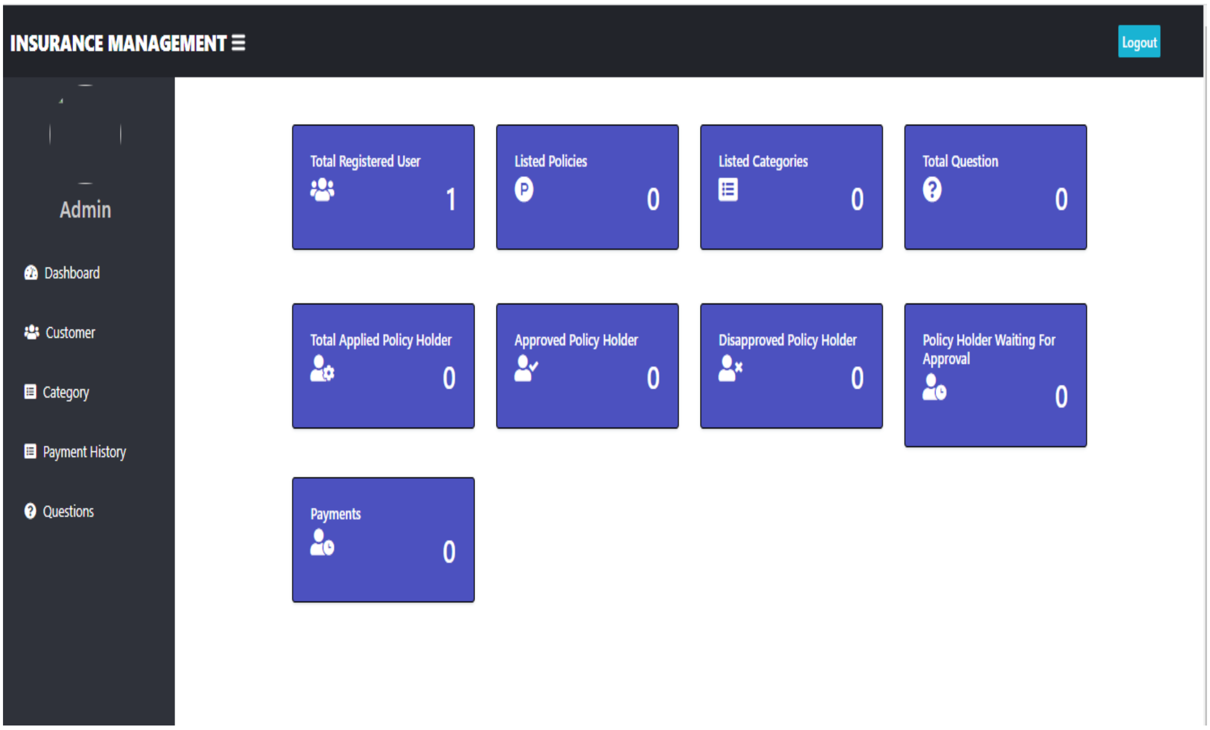
Send Us Your Valuable Feedback !

Name:

Email:

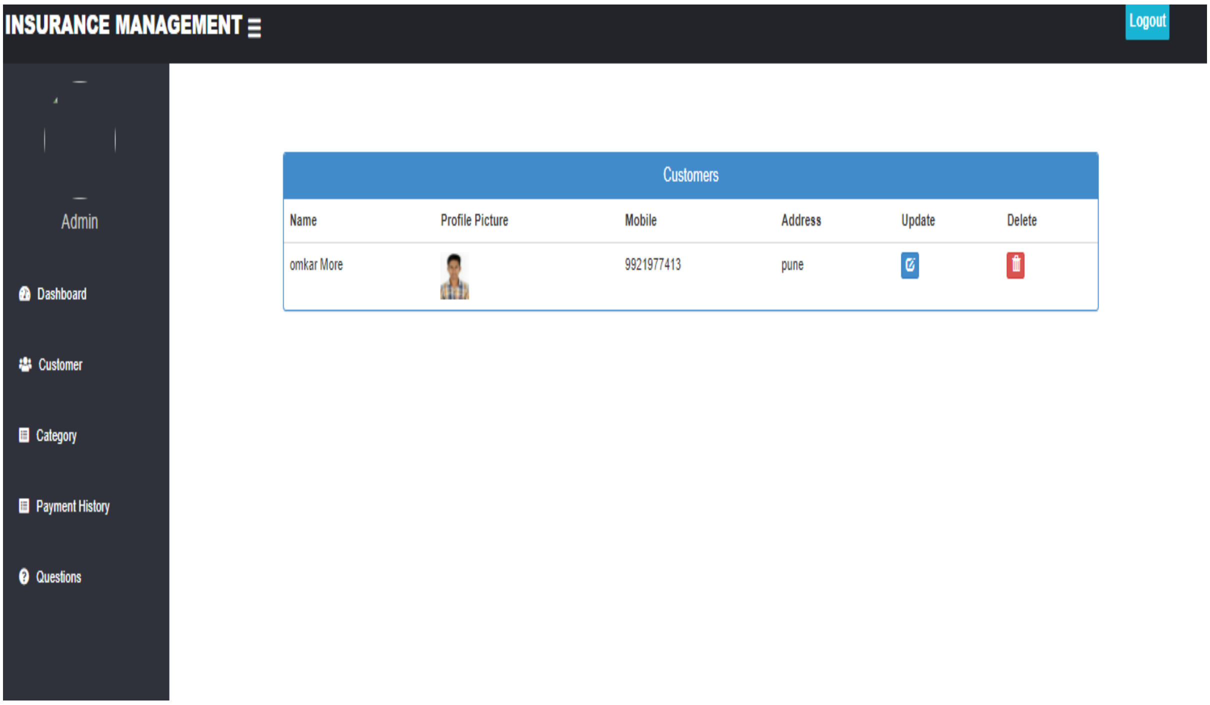
Message:

Screen 3 : Feedback page



Total Registered User	1
Listed Policies	0
Listed Categories	0
Total Question	0
Total Applied Policy Holder	0
Approved Policy Holder	0
Disapproved Policy Holder	0
Policy Holder Waiting For Approval	0
Payments	0

Screen 4 : Admin Page

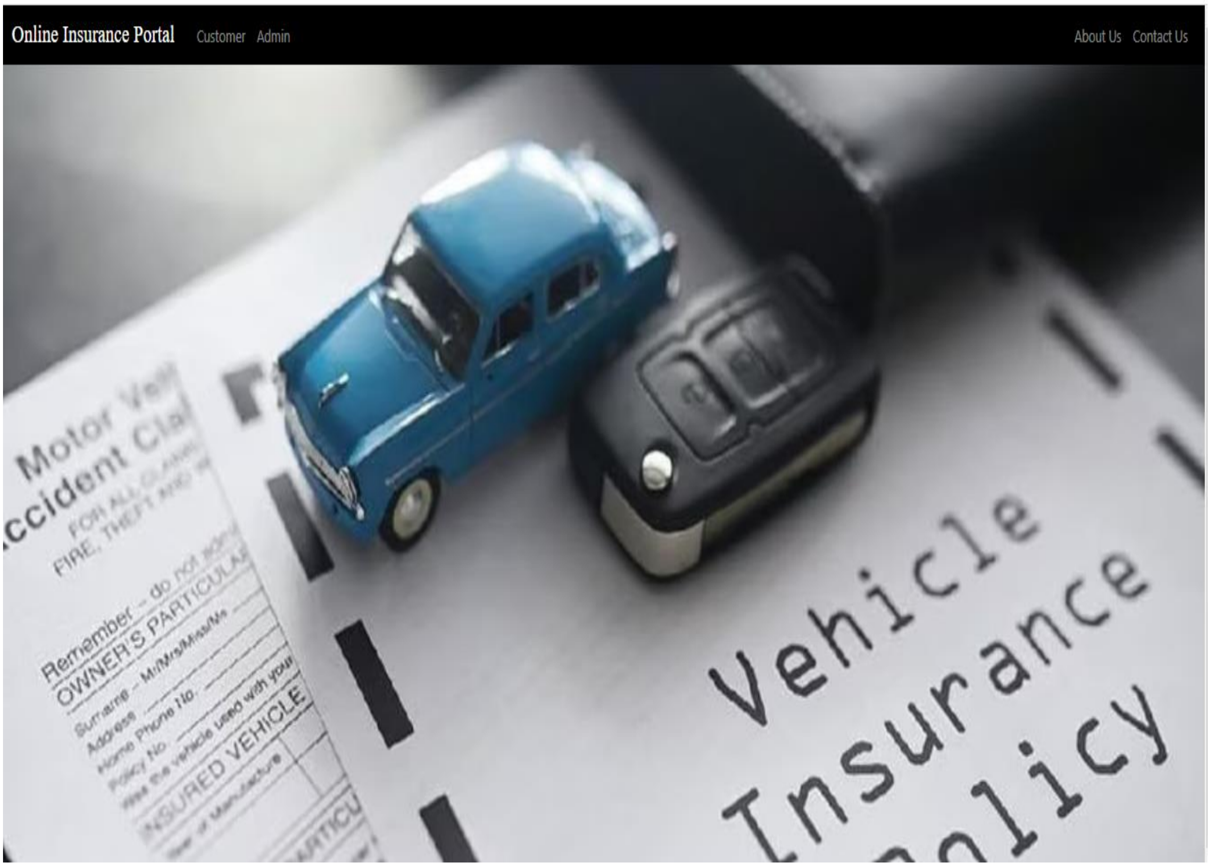


Customers

Name	Profile Picture	Mobile	Address	Update	Delete
omkar More		9921977413	pune		

Screen 5 : Customer Details Page

CHAPTER 4: CODING Sample code



Screen : 1. Index.html file

1.index.html

```
  <!DOCTYPE html> {% load static %}
<html lang="en" dir="ltr">


<style>
    .xyz {
        background-image: url(https://english.mathrubhumi.com/image/contentid/policy:1.8118452:1670553414/Insurance.jpg?$p=8c194f0&f=16x10&w=852&q=0.8);
        background-size: 100%, 100%;
        background-repeat: no-repeat;
    }
</style>


<body>
    {% include "insurance/navbar.html" %}
    <br>
    <section id="section-jumbotron" style="margin-bottom: 0px;" class="jumbotron jumbotron-fluid d-flex justify-content-center align-items-center xyz">
        <div class="container text-center">
            <br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>


        </div>
    </section>



    {% include "insurance/footer.html" %}


</body>


</html>
```

2.Customer(views.py) file

```
def customerclick_view(request):
if request.user.is_authenticated:
return HttpResponseRedirect('afterlogin')
return render(request,'customer/customerclick.html')

def customer_signup_view(request):
userForm=forms.CustomerUserForm()
customerForm=forms.CustomerForm()
mydict={'userForm':userForm,'customerForm':customerForm}
if request.method=='POST':
userForm=forms.CustomerUserForm(request.POST)
customerForm=forms.CustomerForm(request.POST,request.FILES)
if userForm.is_valid() and customerForm.is_valid():
user=userForm.save()
user.set_password(user.password)
user.save()
customer=customerForm.save(commit=False)
customer.user=user
customer.save()
my_customer_group = Group.objects.get_or_create(name='CUSTOMER')
my_customer_group[0].user_set.add(user)
return HttpResponseRedirect('customerlogin')
return render(request,'customer/customersignup.html',context=mydict)

def is_customer(user):
return user.groups.filter(name='CUSTOMER').exists()

@login_required(login_url='customerlogin')
def customer_dashboard_view(request):
```

```
dict={
'customer':models.Customer.objects.get(user_id=request.user.id),
'available_policy':CMODEL.Policy.objects.all().count(),
'applied_policy':CMODEL.PolicyRecord.objects.all().filter(customer=models.Customer.objects.get(user_id=request.user.id)).count(),
'total_category':CMODEL.Category.objects.all().count(),
'total_question':CMODEL.Question.objects.all().filter(customer=models.Customer.objects.get(user_id=request.user.id)).count(),

}
return render(request,'customer/customer_dashboard.html',context=dict)

def apply_policy_view(request):
customer = models.Customer.objects.get(user_id=request.user.id)
policies = CMODEL.Policy.objects.all()
return render(request,'customer/apply_policy.html',{'policies':policies,'customer':customer})
```

CHAPTER 5: LIMITATONS OF SYSTEM

Some of the limitations of a telegram chat bot system for a college project can include:

- Limited functionality: The systems are limited in functionality compared to web or mobile applications. They may not be able to perform complex tasks.

- Technical limitations: The system's capabilities are limited by the available technology and infrastructure. The system may not be able to handle large volumes of data or users, leading to performance issues.

- Dependence on internet connectivity: The system relies on internet connectivity to function. Poor connectivity can affect the system's performance, resulting in delayed or inaccurate information.

- Security concerns: The system may be vulnerable to cyber threats such as hacking or data breaches. It is crucial to implement adequate security measures to safeguard user data and prevent unauthorized access. Overall, while a system can provide several benefits, it is essential to consider its limitations and address them adequately to ensure that users have a positive experience

CHAPTER 6: PROPOSED ENHACEMENTS

Some proposed enhancements of system for a college project could include:

- Integration with other systems: This integration can provide more comprehensive and accurate information to users.
- Personalization: The system can be enhanced by incorporating personalization features, such as user profiles and preferences. This can help provide more customized and relevant information to users.
- Natural language processing: The system can be enhanced by incorporating natural language processing capabilities. This can help the system understand user queries better and provide more accurate and relevant responses.
- Multilingual support: The system can be enhanced by incorporating multilingual support, allowing users to interact with the system in their preferred language. Overall, these enhancements can help improve the functionality, usability, etc

CHAPTER 7: CONCLUSION

all the accumulated effort invested, there are reasons to believe that at the end of the project for “ Online Insurance Portal” for any user finds it self in a better way.

The implementation of the Online Insurance System project offers significant advantages over traditional manual processes in the insurance industry. The system automates policy issuance, streamlines claim processing, enhances user experience, and ensures data security and privacy. Through the utilization of technologies such as Python, web frameworks (Django/Flask), and libraries like SQL-lite the project provides a robust and scalable solution.

The feasibility study demonstrated the technical feasibility of using Python and related technologies for the development of the online insurance system. It also highlighted the economic benefits of increased operational efficiency, cost savings, and improved customer satisfaction. The system's operational feasibility was assessed by gathering feedback from potential users and evaluating the integration with existing systems.

The proposed system's objectives align with industry requirements, aiming to automate insurance processes, enhance user experience, enable online policy purchases, streamline claim processing, improve policyholder management, ensure data security, and provide reporting and analytics capabilities. By achieving these objectives, the system empowers policyholders, insurance agents, and administrators to efficiently manage insurance policies and streamline claim settlements.

CHAPTER 8: BIBLIOGRAPHY

1. www.w3schools.com

2.in.php.net3.en.wikipedia.or g/wiki/PHP

4.www.hotscripts.c om/category/php /

5.www.apache.org/

6. www.mysql .com/click.php? e=35050