

# **BHARTIYA GAS RESIDENCY**

## **SOFTWARE ENGINEERING PROJECT REPORT**

[Submitted in Partial Fulfillment]

As a part of the curriculum of

B.Sc.(H)Computer Science

From



Shyama Prasad Mukherji College, New Delhi

University of Delhi

**SUBMITTED BY**

AnkitaKumari- 16075570005

KhushbooGoel- 16075570027

Khushboo Grover-1607550028

Radhika-16075570050

B.Sc.(H) Computer Science

IV Semester

Shyama Prasad Mukherji College

Punjabi Bagh(West), New Delhi-110026

## Acknowledgement

In performing our project, we had to take the help and guideline of some respected persons, who deserve our greatest gratitude. The completion of this project gives us much pleasure. We would like to show our gratitude to Dr. Baljeet Kaur for giving us a good guideline for this project throughout the numerous consultations. We would like to expand our deepest gratitude to all those who directly and indirectly guided us in writing this project.

We would also like to thank Ms Pratibha Yadav, Mr. Ravinder, Mr. Uday for giving his valuable suggestions and ideas when we were in need.

Many people, especially our classmates and team members itself have made valuable comments and suggestions on this proposal which gave us an inspiration to improve our work. We are immensely grateful to all who are involved in this project as without their inspiration and valuable suggestions it would not have been possible to develop the project within the prescribed time.

Ankita Kumari- 16075570005 *Ankita*

Khushboo Goel- 16075570027 *Khushboo*

Khushboo Grover-1607550028 *Khushboo*

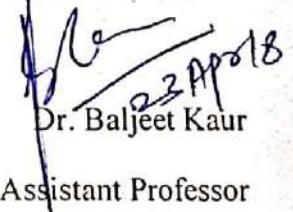
Radhika-16075570050 *Radhika*

## CERTIFICATE

This is to certify that the content of this project entitled, "Bhartiya Gas Residency" by Ankita Kumari, Khushboo Goel, Khushboo Grover and Radhika is the bona fide work of them submitted to Shyama Prasad Mukherji College, New Delhi for consideration in partial fulfillment of the requirement of Delhi University as a part of the curriculum of B.Sc.(H) Computer Science.

The original research work was carried out by them under my supervision in the academic year 2018. On the basis of the declaration made by them I recommend this project for evaluation.

Certified by:

  
Dr. Baljeet Kaur  
Assistant Professor

Shyama Prasad Mukherji College

University of Delhi

New Delhi

## **Introduction**

Bhartiya gas residency is an online gas booking software in which the user can book cylinders, pay for gas pipeline and request for extra cylinder. The objective of this project is to create the system where the customer can easily book LPG gas cylinder through online system and agency can track the record of its customer and the delivery of the cylinder.

The system will help the customer by providing a user interactive interface for booking the gas cylinder and by providing the bill payment option for gas pipeline through online which will save their time and money. Through this system, we are solving the customer's problem by providing the facility for them to book cylinders from home.

Bhartiya gas residency is widely accepted model for the following reasons :

- Customer record is maintained
- Customer can cancel their booking within 24 hours
- Customer can request for extra cylinder in case of any event
- Customer can pay bill for gas pipeline
- Online booking and bill payment through internet from any point
- Check if customer is valid to book a cylinder

## TABLE OF CONTENT

CONTENTS	PAGE NO.
1.PROBLEM STATEMENT	1
2.SOFTWARE LIFECYCLE MOODEL	3
3.PROJECT SCHEDULING	5
4.SOFTWARE REQUIREMENT SPECIFICATION	7
5 ENTITY RELATIONSHIP DIAGRAM (ERD)	28
6.DATA DICTIONARY	29
7.CONTEXT LEVEL DIAGRAM	32
8.DATA FLOW DIAGRAM (LEVEL-1)	33
9.DATA FLOW DIAGRAM (LEVEL-2)	34
10.USE CASE DIAGRAM	36
11.USE CASE DESCRIPTION	37
12.FUNCTION POINT METRICS	44
13.EFFORT ESTIMATION USING COCOMO MODEL	46
14.RISK ANALYSIS	48
15.TESTING	51
16. BIBLIOGRAPHY	

## PROBLEM STATEMENT

BHARTIYA GAS RESIDENCY is a web based application to book gas cylinders online. The software provides a way to enroll new customers or login to their already registered portal. The software will provide the functionality as explained below:

- **Login**

To login, customer needs to provide a valid E-mail/Adhaar number/PAN number and password. Admin needs to login by providing agency name and password in order to keep track of the various orders made by the customers.

- **Registration**

The software will require customer's details like name, address, adhaar number, PAN number, and phone number.

- **Edit details**

Existing user may change his/her address and phone number.

- **Make payment**

The software will also provide billing options for users having gas pipeline service. Customer needs to provide meter number and address.

- **Cylinder payment**

The customer may pay for cylinder by COD(cash on delivery)or net banking.

- **Booking cylinder**

For booking cylinders, the software provides three options. The customer can book cylinder online or via email/message.

- **Cancel Booking**

The software will provide option to cancel booking.

- **Request Extra cylinder**

The software will provide an option to book at most two extra cylinders in case of any event like birthday, parties. Extra charges will be applied.

- **Request subsidy**

Customer will have the facility to avail or give up subsidy

- **Request automatic booking**

It's up to the customer to avail this option or not. If customer enables this feature, the software will automatically book cylinder after 21 days time period and will notify customer via text message and email.

- **Customer's feedback/Complaint**

The customer can share his experience of service. Customer can share any complaint regarding the product and service.

## **SOFTWARE LIFECYCLE MODEL**

The waterfall model is best suitable for our project.

The requirements of the project are well known and understood. As all the requirements are well defined and stable this will lead to high quality SRS. Revisiting the earlier complete phases is not required.

A high quality SRS ensures a high quality product.

### **What is Waterfall Model?**

Waterfall Model is also referred to as linear-sequential life cycle model. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of software development model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model software testing starts only after the development is complete. In waterfall model phases do not overlap. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of software development model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model software testing starts only after the development is complete. In waterfall model phases do not overlap.

Very less customer interaction is involved during the development of the product. Once the product is ready then only it can be demoed to the end users. Once the product is developed and if any failure occurs then the cost of fixing such issues are very high, because we need to update everywhere from document till the logic.

### **Advantages of waterfall model:**

1. This model is simple and easy to understand and use.
2. It is easy to manage due to the rigidity of the model-each phase has specific deliverables and a review process.
3. In this model phases are processed and completed one at a time. Phases do not overlap.
4. Waterfall model works well for smaller projects where requirements are very well understood.

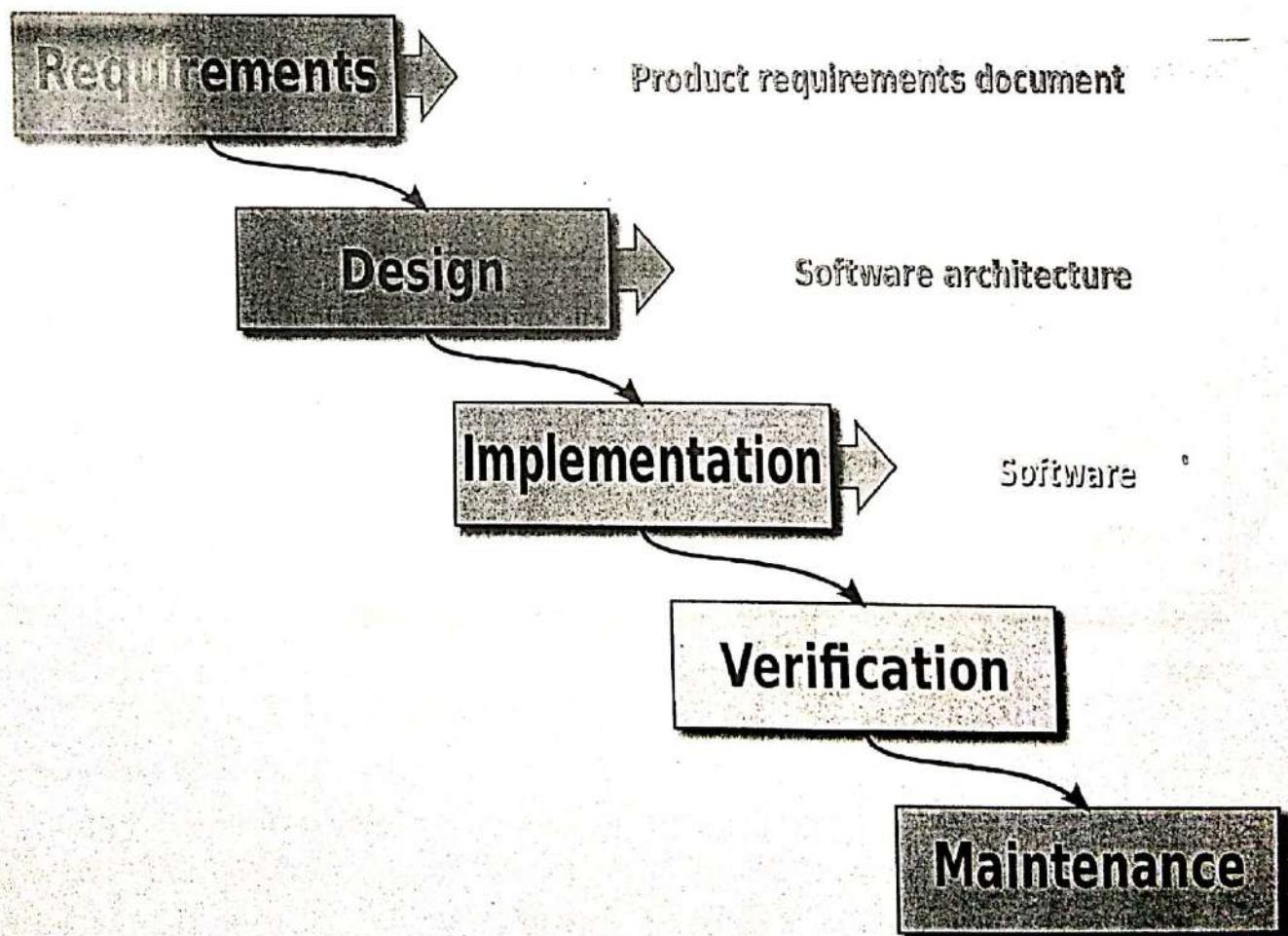
### **Disadvantages of waterfall model:**

1. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
2. No working software is produced until late during the life cycle.
3. High amounts of risk and uncertainty.
4. Not a good model for complex and object-oriented projects.

5. Poor model for long and ongoing projects.
6. Not suitable for the projects where requirements are at a moderate to high risk of changing.

#### When to use the waterfall model:

1. This model is used only when the requirements are very well known, clear and fixed.
2. Product definition is stable.
3. Technology is understood.
4. There are no ambiguous requirements
5. Ample resources with required expertise are available freely
6. The project is short.



## PROJECT SCHEDULING

**Time Line Chart**

Work tasks	January				February				March				April			
	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4
Problem statement	[Start]		◆													
Software Lifecycle model		[Start]														
Project scheduling			[Start]													
Software requirement specification				[Start]												
Entity Relationship Diagram					[Start]			◆								
Data Dictionary							[Start]									
Context Level Diagram								[Start]								
Data Flow Diagram(level 1)									[Start]							
Data flow Diagram(level 2)										[Start]			◆			
User case Description											[Start]					
Function point metrics												[Start]				
Effort estimation using COCOMO model												[Start]		◆		
Risk Analysis					[Start]											◆
Testing													[Start]		◆	

### Project Table

Work	Planned start	Actual Start	Planned Complete	Actual complete	Assigned Person	Effort Allocated
Problem Statement	Jan,w1	Jan,w1	Jan,w2	Jan,w2	Khushboo Grover KhushbooGoel	2 p-w
Software Lifecycle model	Jan,w2	Jan,w3	Jan,w3	Jan,w4	Khushboo Grover KhushbooGoel	2 p-w
Project Scheduling	Jan,w4	Jan,w4	Jan,w4	Jan,w4	AnkitaKumari	1 p-w
Software Requirement specification	Jan,w4	Feb w1	Feb,w2	Feb,w2	Radhika Khushboo Grover	2 p-w
Entity Relationship Diagram	Feb,w2	Feb,w2	Feb,w3	Feb,w3	Khushboo Goel Khushboo Grover	2 p-w
Data Dictionary	Feb,w3	Feb,w4	Feb, w4	Feb,w4	AnkitaKumari	1 p-w
Context Level Diagram	Feb,w3	Feb,w4	Feb,w4	March, w1	KhushbooGoel Khushboo Grover	2p-w
Data Flow Diagram(level 1)	March, w1	March,w2	March, W2	March,w3	KhushbooGoel Khushboo Grover	2p-w
Data Flow Diagram(Level 2)	March,w2	March,w4	March, w3	March, w4	KhushbooGoel Khushboo Grover	2p-w
User case Description	March,w4	April, w1	March, w4	April, w1	Radhika Rajput	1 p-w
Functional points	April,w1	April,w1	April,w1	April,w2	Khushboo Goel Ankita Kumari	2 p-w
Effort estimation using COCOMO model	April,w2	April,w2	April,w2	April,w2	Radhika Rajput	1 p-w
Risk analysis	Jan w4	Jan w4	April, w3	April, w3	Ankita Kumari	1 p-w
Testing	April,w2	April,w3	April,w3	April,w3	Khushboo Grover	1 p-w

## SOFTWARE REQUIREMENT SPECIFICATION

### Introduction

#### **Purpose:**

This is software requirements specification (SRS) for the "BHARTIYA GAS RESIDENCY". The purpose of this document is to provide

1. A description of the environment in which the software is expected to operate.
2. Definition of the software capabilities.

The SRS will serve to establish a basis of agreement between the client and development team about the functionality to be provided by the software.

#### **Scope:**

The "BHARTIYA GAS RESIDENCY" is a web based application, which helps the customer to book, borrow and pay for cylinders. The customer has to fill the various forms and number of copies of the forms can be easily generated at a time.

A customer can also pay bills for gas pipeline system with the help of this system by providing meter number and passwords. The software will provide customer with the fee receipt.

The software will also provide the facility of automatic booking. In this option, the software will automatically book cylinders after 21 days and will notify customer via text messages and e-mail.

The customer can avail or give up subsidy with the help of this software. Software will also have option to cancel order within 24 hours.

#### **Definition, Acronyms and Abbreviations:**

User-someone who interacts with the software

Stakeholder- person who is not a developer but interacts with a system.

#### **References:**

IEEE Software Engineering Standards Committee "IEEE Standard 29148:2011, IEEE Recommended Practice for Software Requirements Specifications".

#### **Overview:**

The remainder of this document provides an overview of the system functionality and system

interaction with other systems. The document also mentions the system constraints and assumptions about the product.

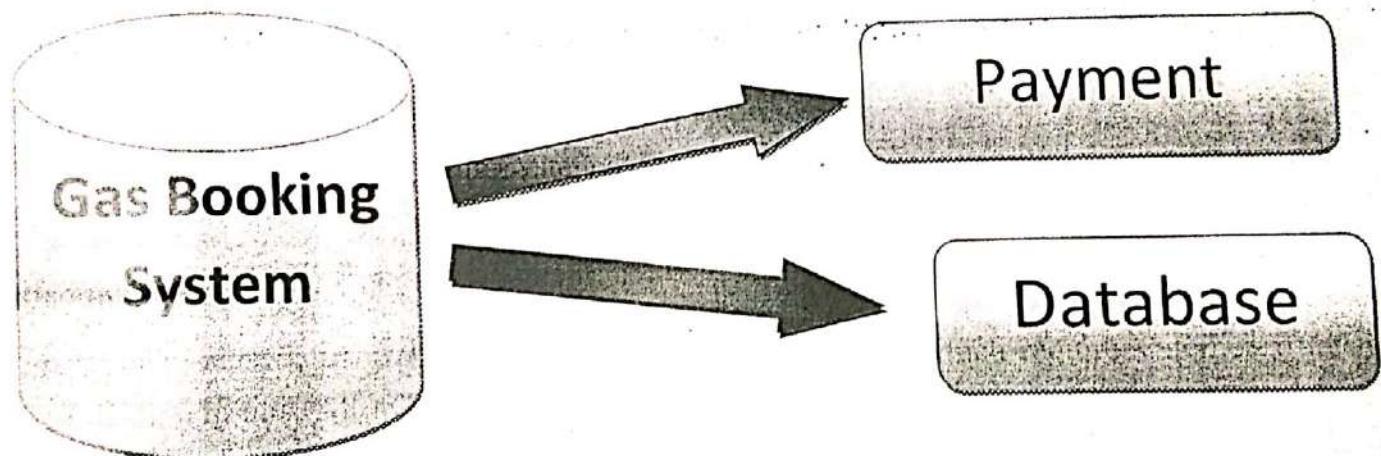
### **The overall Description:**

This section will show how the system interacts with other systems and introduce the basic functionality of it.

#### **Product Perspective:**

"BHARTIYA GAS RESIDENCY" is a web based application which requires internet to fetch and display results. All system information is maintained in a database, which is located on a web server.

The system is self contained. However it is possible to exchange data with other system through external interfaces.



The software's database will store the user details. It includes user name, PAN number /adhaar number, phone number, address, customer's ID and meter number (optional).The information will be used for keeping records of booked cylinder and billing of gas pipeline.

#### **Product function:**

The major functions of this software are the following:

- 1.**Login:** To login, customer needs to provide a valid email/PAN number/adhaar number and password.
- 2.**Registration:** In order to make account, customer needs to give details like name, adhaar/PAN and contact info.
- 3.**Edit Details:** Existing user can edit his or her details.
- 4.**Make Payment:** Payment option for billing of gas pipeline or cylinders.
- 5.**Cancel booking:** The software will provide facility to cancel their order.
- 6.**Request extra cylinder:** Customer can book extra cylinders in case of any event.

- 7.Request/Cancel subsidy:** Customer can avail or give up subsidy.
- 8.Request Automatic booking:** Customer can enable this option to allow software to book cylinders automatically after 21 days.
- 9.Customer feedback /complaint:** Customer can give feedback and also can complaint against the services.

#### **User Characteristics:**

User should have basic knowledge of English language and can operate computer easily.

#### **Constraints:**

The internet connection is a constraint for the software. Since the software fetches data from the database over the internet, it is crucial that there is an internet connection for the software to function.

Memory - 4GB

#### **User Interfaces:**

First time user will see the login page when he/she opens the application. User has to first register (if new user). An existing user can edit their details. After login, user will see three booking options:

1. Get cylinders, where user can book, borrow or cancel cylinders
2. Pay for gas pipeline, when customer can pay for gas pipelines or cylinders
3. Automatic booking, where user can enable the option to book cylinders automatically.

#### **Hardware Interfaces:**

No direct hardware interfaces.

#### **Software Interfaces:**

1. **Operating System:** Windows(for best support and user friendliness)
2. **Database:** To save customers records, we have chosen SQL database.

#### **Communication Interfaces:** None.

#### **Functional Requirements:**

This software will work on windows to make it easily accessible to the customer. User need to login to his/her existing portal to book cylinders. A new user can register and provide necessary details like name, address, contact information, adhaar number, pan number. Meter number will be required if the user wishes to pay the bill for gas pipeline service.

The user can avail subsidy or give up subsidy by submitting the required document. Functionality to cancel order will be provided by the software with some terms and conditions. User can book extra cylinders in specific conditions. User can choose automatic booking option to automatically book cylinders after 21 days time span. A user can give feedback regarding the service provided by the software.

### **Performance Requirement:**

**TITLE:** The response time of a search.

**DESCRIPTION:** The response time of a search is the overall time beginning with the initial user action (click on the search button) on the device, the request going to server, the response received from the server, and finally the response processing by the mobile application.

**METER:** Measurements obtained from 1000 searches during testing.

**WISH:** No more than 1 second during 100% of the searches during testing.

**Design Constraints:** The software is designed for web portal; screen size limitation will be major design consideration.

**Software System Attributes** – The requirements in this section specify the required reliability, availability, security, maintainability and portability of the software system.

**Reliability** – The application is reliable product that produces fast and verified output of all its process.

**Availability** – The application will be available to use and carry out operations conveniently. Reports and Receipt are made available to the customer.

**Security** – The application will be password protected. User will have to enter correct username, password and role in order to access the patient.

**Maintainability** – The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules. It should maintain a patient's record for 6 month time span.

**Portability** – The application will be easily portable on any window-based system that has Java 1.5 or more installed.

## Screens



Number of input :0

Number of outputs:1

Number of Files:0

Numbers of enquiries:0

External Interfaces:0

**New User**

Name:

Sex: Male  Female  Others 

Date of birth:

Age:

Phone number:

Enter password:

Confirm password:

Address:

Adhaar Number:

Meter number (Optional):

Would you like to facilitate automatic booking after 21 days?

Information provided is correct?

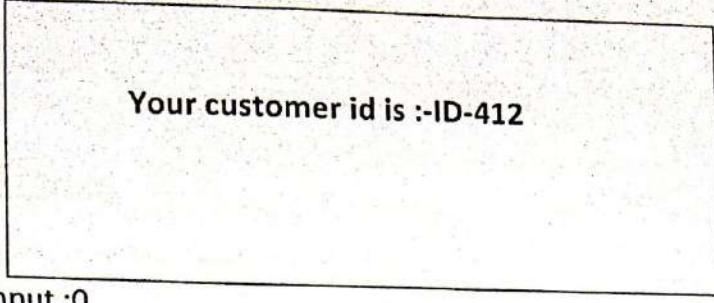
Number of input :12

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0



Your customer id is :-ID-412

Number of input :0

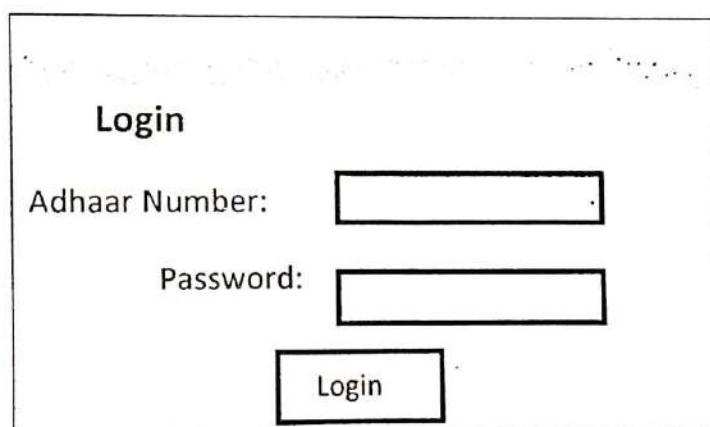
Number of outputs:1

Number of Files:0

Numbers of enquiries:0

External Interfaces:0

4.)



**Login**

Adhaar Number:

Password:

Number of input :2

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

5.

**Welcome**

\*Customer can request extra cylinder only once in a year.

\*Cylinders booking should be cancelled within 24 hours or else you have to pay charges

Number of input :1

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

6.

**Edit Details**

Name : XYZ

Phone number : 987654321

Address : Abc street.

Number of input :1

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

7.)

**Post Complaint**Booking ID: Complaint: \_\_\_\_\_  
\_\_\_\_\_

Number of input :2

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

8.)

**Your Complaint has been registered**

Your complaint ID is ID-412. We will get back to you soon.

Thank you.

OK

Number of input :0

Number of outputs:1

Number of Files:0

Numbers of enquiries:0

External Interfaces:0

**Booking Portal**

Booking id:- \_\_\_\_\_

**Payment**

Cash On delivery



e-payment

Confirm Booking

Number of input :2

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

10.)

**Cancellation Portal**

Booking id: \_\_\_\_\_

Are you sure you want to cancel the booking?

Number of input :2

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

11.)

**Requesting Extra Cylinders**

Booking id:- \_\_\_\_\_

**Reason:**

Birthday

Other

Marriage

**Number of cylinders needed:**

One

Two

**Submit****Cancel**

Number of input :3

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

**Automatic Booking**

Booking id:- \_\_\_\_\_

Are you sure you want to do automatic booking?

Yes

No

Medium of payment



Cash on delivery



e-payment

Number of input :2

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

**Automatic Booking****e-Payment**

\*Add details for the payment

Account Number:- \_\_\_\_\_

Password:- \_\_\_\_\_

OTP:- \_\_\_\_\_

(OTP is send at \*\*\*\*\*89)

Confirm

Number of input :3

Number of outputs:1

Number of Files:1

Numbers of enquiries:1

External Interfaces:0

14.)

**Automatic Booking Confirmation**

Your account number has been registered

Thank you

Number of input :0

Number of outputs:1

Number of Files:0

Numbers of enquiries:0

External Interfaces:0

15.)

**Subsidy**

Retain

Leave

Cancel

Number of input :1

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

16.)

**Update Details**Name: AbcAddress: XYZ StreetContact: 987\*\*\*\*342\*Any booking:- Yes\*Subsidy status:- Retained\*Automated Booking:- No\*Extra Cylinder:- NoNumber of input :0Number of outputs:1Number of Files:1Numbers of enquiries:0External Interfaces:0

**Payment Option**

Cash on delivery

Debit card/Credit card

Number of input :1

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

18.)

**Pay Your Bill !!**

Cylinder       Pipeline

Amount(Automatically generated):

Card Number:

CVV:

Number of input :4

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:1

19.)

27

### Welcome to ABC bank online payment!!

Customer Name:

ATM pin:

OTP:

( \*OTP send to \*\*\*\*\*45)

Pay

Number of input :3

Number of outputs:1

Number of Files:1

Numbers of enquiries:0

External Interfaces:0

20.)

### Payment Transaction Successful!

OK

Number of input :0

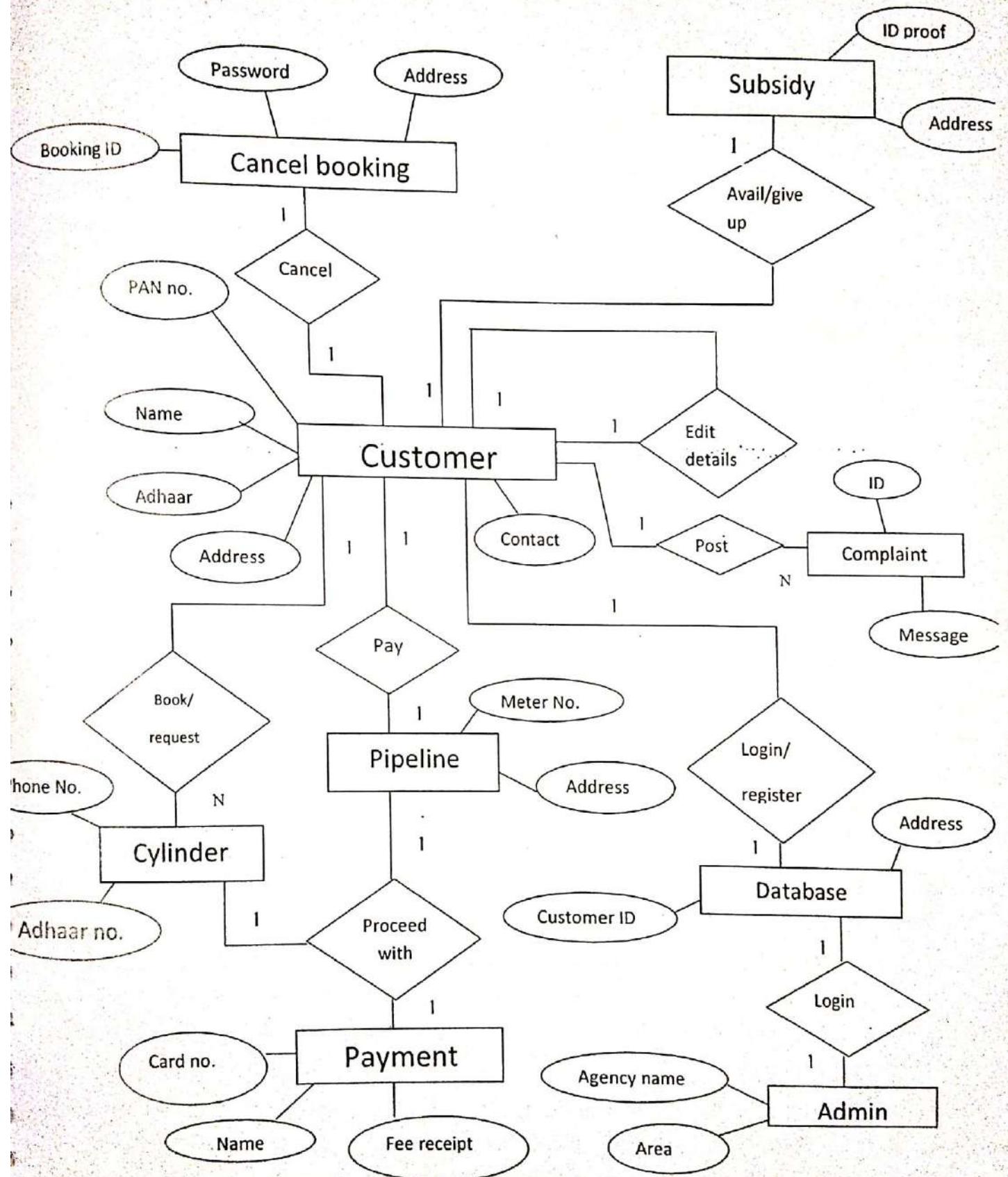
Number of outputs:1

Number of Files:0

Numbers of enquiries:0

External Interfaces:0

## ENTITY RELATIONSHIP DIAGRAM



*Pratibha*

## DATA DICTIONARY

### Data Dictionary for Customer:

S.NO	Field Type	Data type	Field length	Description
1.)	Name	Alphanumeric	20	Name of the customer
2.)	Address	Alphanumeric	30	Address of the customer
3.)	Adhaar	Numeric	12	Adhaar number of the customer
4.)	PAN	Alphanumeric	10	PAN number of the customer
5.)	Contact	Numeric	10	Contact number of the customer
6.)	Edit Details	Alphanumeric	30	Asks customer for editing his/her details.

### Data Dictionary for Cylinder Booking:

S.No	Field Type	Data Type	Field Length	Description
1.)	Phone	Numeric	10	Phone number of the customer
2.)	Adhaar	Numeric	12	Adhaar number of customer

### Data Dictionary for Payment:

S.NO	Field Type	Data Type	Field Length	Description
1.)	Card number	Numeric	12	Card number of the customer
2.)	Name	Alphanumeric	20	Name of the customer
3.)	Fee receipt	Alphanumeric	30	Fee receipt to the customer

### Data Dictionary for Admin:

S.No	Field Type	Data Type	Field Length	Description
1.)	Agency name	Alphanumeric	15	Name of the gas agency
2.)	Area	Alphanumeric	15	Name of the customer's area
3.)	Login	Alphanumeric	15	Login of customer

### Data Dictionary for Database:

S.No	Field type	Data type	Field length	Description
1.)	Customer id	Numeric	12	Id of the customer
2.)	Address	Alpha numeric	30	Address of the customer

### Data Dictionary for pipeline:

S.No	Field type	Data type	Field length	Description
1.)	Meter number	Numeric	15	Meter number of the customer
2.)	Address	Alphanumeric	30	Address of the customer

### Data Dictionary for Complaint:

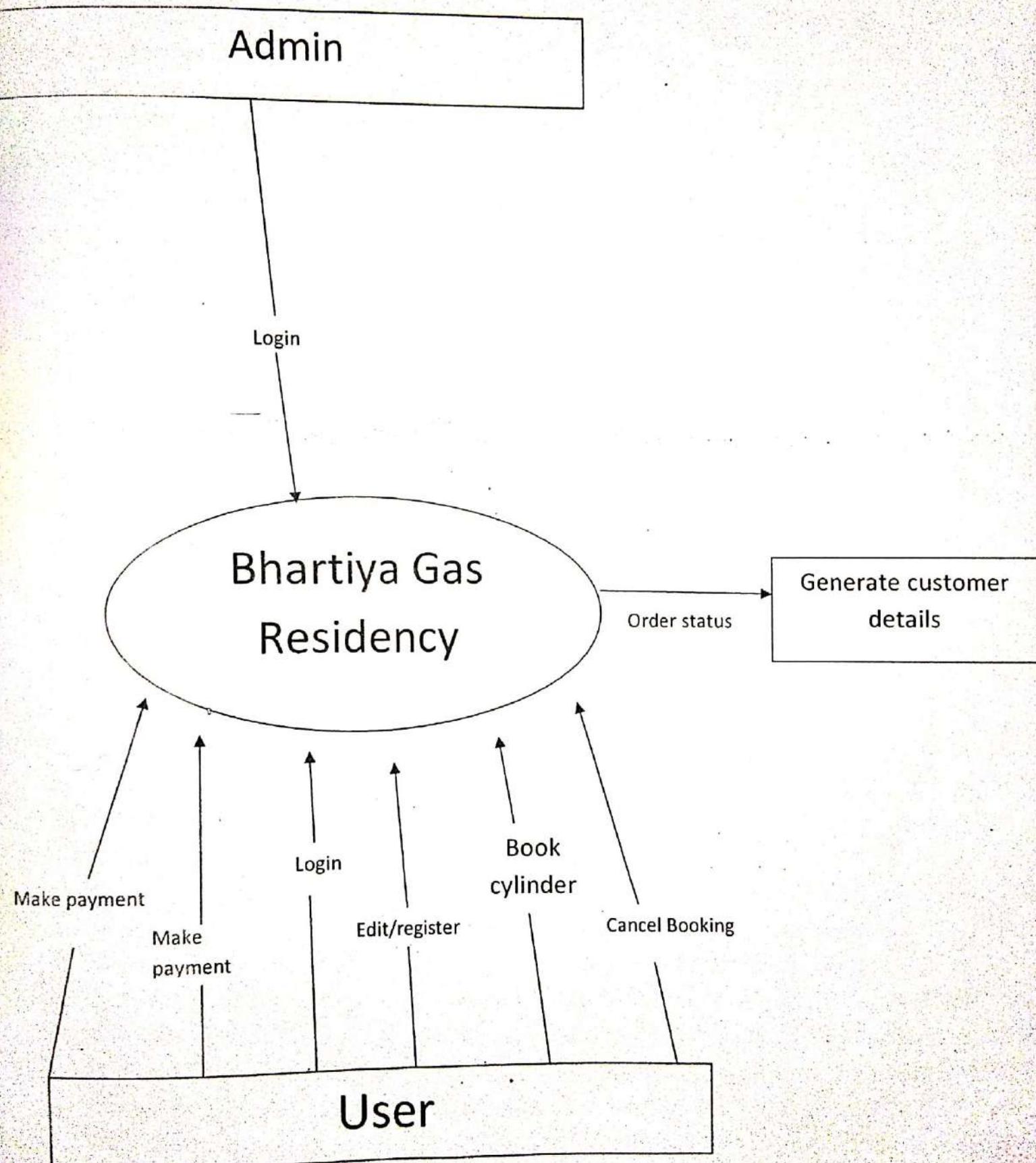
S.No	Field Type	Data type	Field length	Description
1.)	ID	Numeric	10	ID of the customer
2.)	Message	Alphanumeric	30	Message from the customer

### Data Dictionary for subsidy:

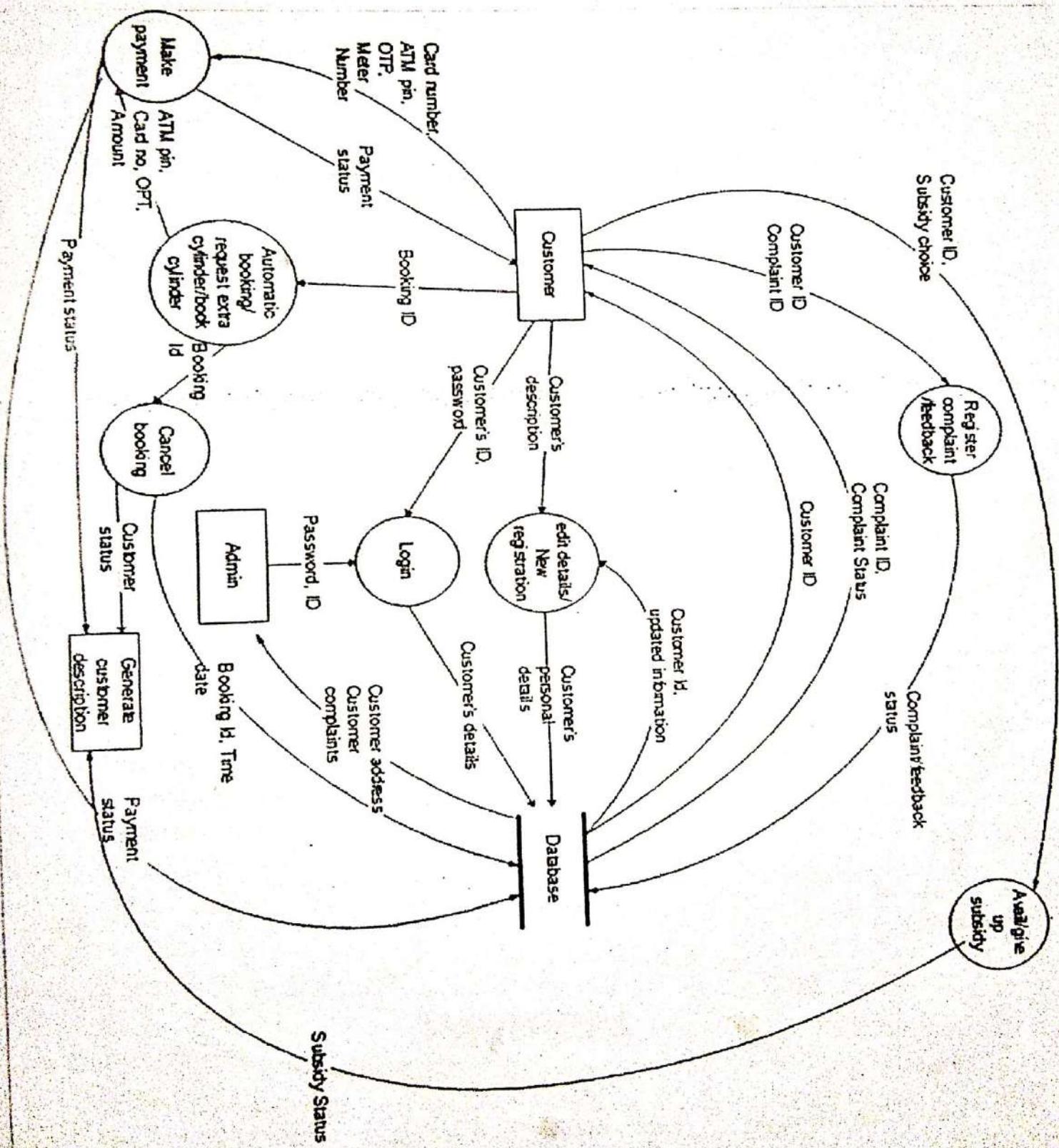
S.No	Field type	Data type	Field length	Description
1.)	ID proof	Numeric	10	Proof of the customer's id
2.)	Address	Alphanumeric	30	Address of the customer

### Data Dictionary for Cancel Booking:-

S.No	Field Type	Data Type	Field Length	Description
1.)	Booking id	Numeric	10	Booking id of the customer
2.)	Password	Numeric	7	Password of the id
3.)	Address	Alpha numeric	30	Address of the customer

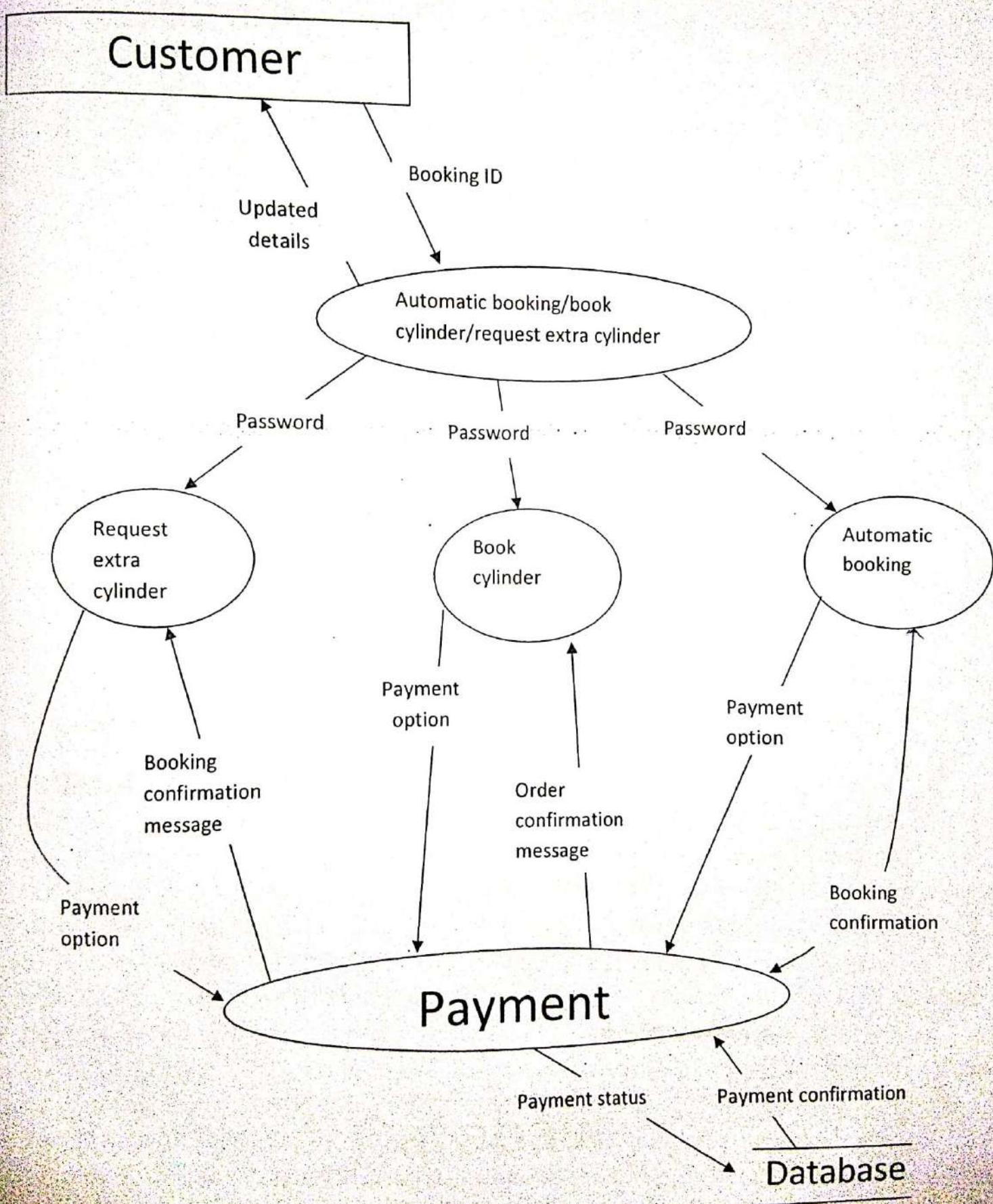
CONTEXT LEVEL DIAGRAM

## DATA FLOW DIAGRAM (level 1)

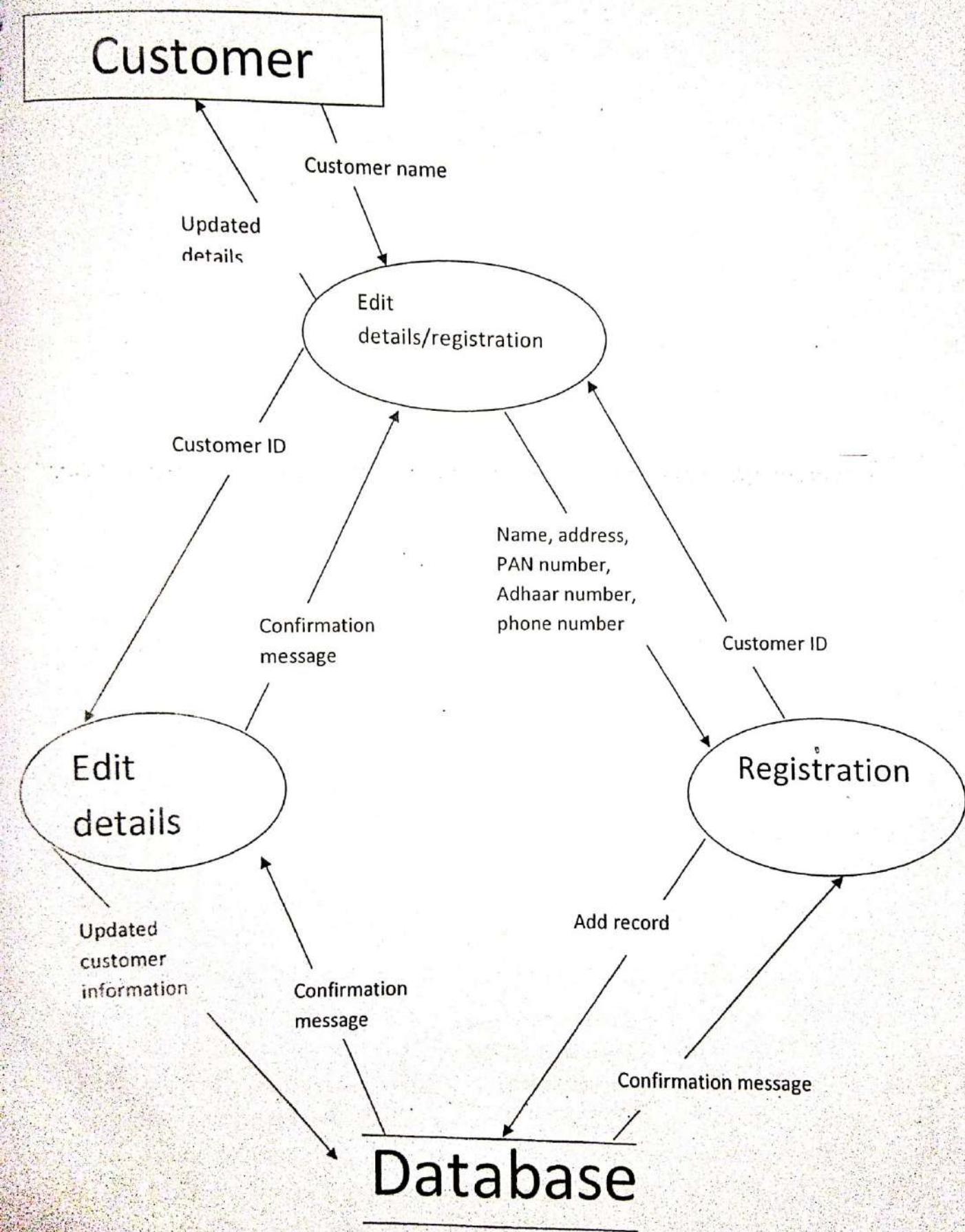


## DATA FLOW DIAGRAM Level-2

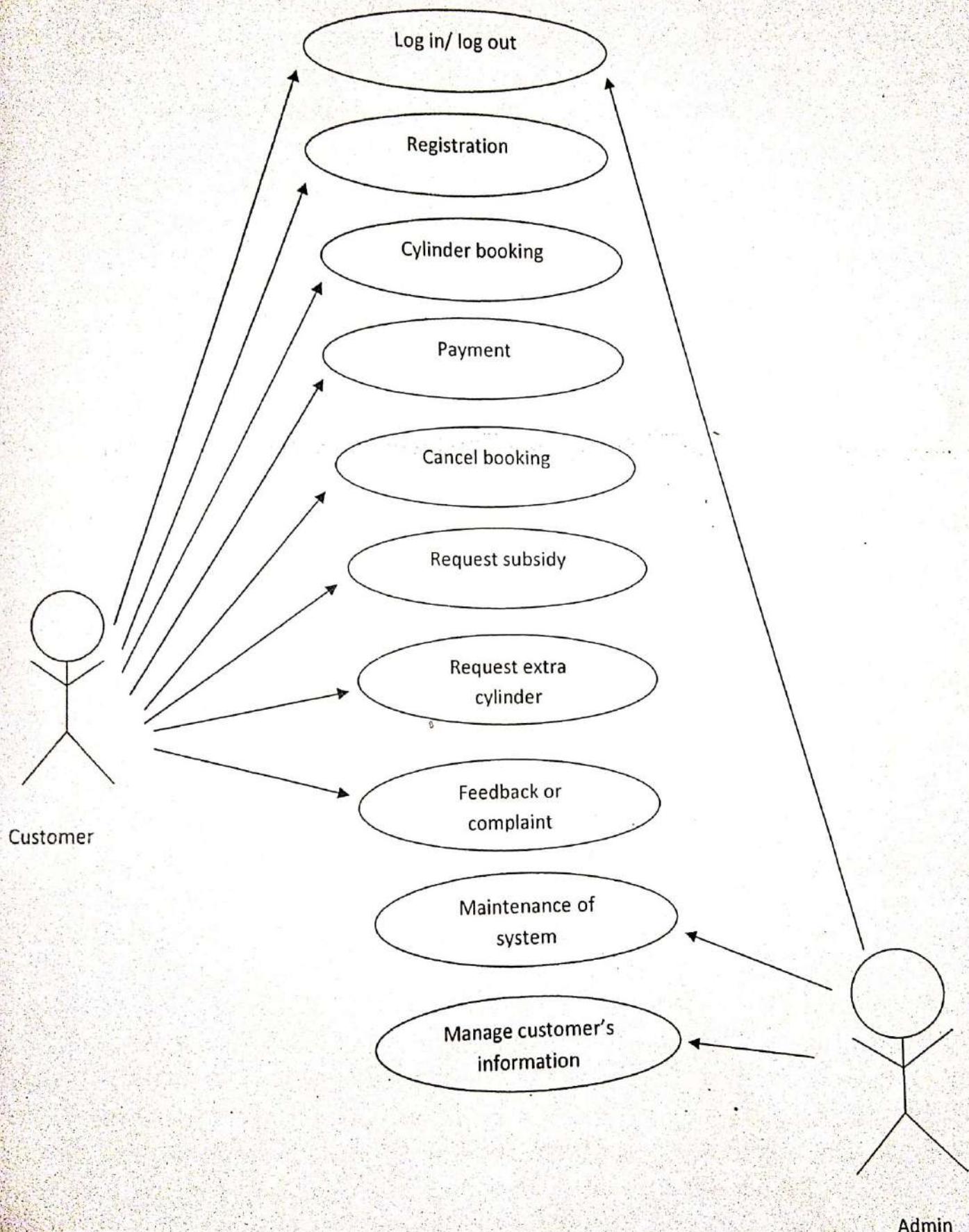
### Automatic booking/booking cylinder/request extra



## Edit details/register



## USE CASE DIAGRAM



## USE CASE DESCRIPTION

### Login

#### **Brief description**

Login use case describes how an actor Logs into system.

#### **Actors**

The following actors are engage in the activities described in the use case:

1. Customer
2. Admin

### **Flow of Events**

#### **Basic Flow**

1. This use case starts when the customer logs in for booking cylinder.
2. The Actor enters his/her details. The actor can be any of customer and admin.
3. System checks the details and role of the actor.

#### **Alternative flow**

If the actor enters wrong details the system will display a failure message. The system will cancel the login or throws the actor at the beginning of the basic flow.

#### **Special requirement**

none

#### **Pre-condition**

The actor has a account for logging into the system. Or he/she can register himself/herself.

#### **Post-condition**

If the actor was successfully logged into system, he can book cylinder, cancel cylinder if the logged in user was a customer. If the logged in user was a admin then he can maintain the system or user's information.

### Registration

#### **Brief description**

Registration use case describes how a new actor registers into system.

#### **Actors**

The following actors are engage in the activities described in the use case:

1. Customer

## **Flow of Events**

### **Basic Flow**

1. This use case starts when a new actor registers himself/herself.
2. The Actor enters his/her details. The actor is customer.
3. System checks the details entered by the actor.

### **Alternative flow**

If the new actor enters any wrong detail the system will display a failure message. The system will cancel the registration or requests the actor to reenter the details for a successful registration.

### **Special requirement**

none

### **Pre-condition**

The actor must have an account in LPG gas booking agency.

### **Post-condition**

If the actor was successfully logged into system, he can book cylinder, cancel cylinder.

## **Cylinder booking**

### **Brief description**

Cylinder booking use case describes how a actor can book a cylinder.

### **Actors**

The following actors are engage in the activities described in the use case:

1. Customer

## **Flow of Events**

### **Basic Flow**

1. This use case starts when an actor registers/ logs in himself/herself in the system successfully.
2. System checks the details entered by the actor.
3. If the actor books a cylinder after 21 days, System will accept actor's request and books a cylinder.
4. Actor will receive an email or message about the status of booking.

### **Alternative flow**

If the actor enters any wrong detail or books a cylinder before 21 days, the system will display a failure message. The system will request the actor to reenter the details for a successful booking.

### **Special requirement**

none

### **Pre-condition**

The actor must have a account in system. He/ She can only book cylinder after 21 days.

### **Post-condition**

If the use case was successful, the actor can cancel the booking or he/she can pay via any medium.

## **Payment**

### **Brief description**

Login use case describes how an actor pays for his/her cylinder booking.

### **Actors**

The following actors are engage in the activities described in the use case:

- I. Customer

## **Flow of Events**

### **Basic Flow**

1. This use case starts when the customer books a cylinder.
2. Actor can pay for cylinder via net-banking or cash on delivery. The Actors enters his/her details. System checks the details.
3. Actor can also pay for pipeline connection by providing meter number and details.

### **Alternative flow**

If the actor enters wrong details the system will display a failure message and requests the actor to pay again.

## **Special requirement**

none

### **Pre-condition**

The actor received an email or message after the booking.

### **Post-condition**

If the actor wants to cancel the cylinder booking, he can cancel cylinder booking by entering his/her detail.

## **Cancel Booking**

### **Brief description**

Cancel booking use case describes how an actor cancels his/her cylinder booking.

**Actors**

The following actors are engage in the activities described in the use case:

1. Customer

**Flow of Events****Basic Flow**

1. This use case starts when the booking status was successful and the actor wants to cancel the booking.
2. Actor can cancel the booking by entering his/her details.
3. Actor will receive a status after canceling the booking successful.

**Alternative flow**

If the actor enters wrong details the system will display a failure message and requests the actor to enter the details again.

**Special requirement**

none

**Pre-condition**

The actor received an email or message after the booking and net-banking payment

**Post-condition**

none

**Request extra cylinder****Brief description**

Request extra cylinder use case describes how a actor can book a extra cylinder in case of emergency or any occasion.

**Actors**

The following actors are engage in the activities described in the use case:

1. Customer

**Flow of Events****Basic Flow**

1. This use case starts when a actor registers/ logs in himself/herself in the system successfully..
2. System checks the details entered by the actor.
3. If the actor requests a extra cylinder before 21 days, System will check the availability of cylinder.
4. Actor will receive an email or message about the status of booking.

### **Alternative flow**

If the actor enters any wrong detail, the system will display a failure message. The system will request the actor to reenter the details for a successful booking.

### **Special requirement**

none

### **Pre-condition**

The actor must have an account in system.

### **Post-condition**

If the use case was successful, the actor can cancel the booking or he/she can pay via any medium.

## **Request Subsidy**

### **Brief description**

Login use case describes how an actor pays for his/her cylinder booking.

### **Actors**

The following actors are engage in the activities described in the use case:

1. Customer

### **Flow of Events**

#### **Basic Flow**

1. This use case starts when the customer books a cylinder.
2. Actor can pay for cylinder via net-banking or cash on delivery. The Actors enters his/her details. System checks the details.
3. Actor can also pay for pipeline connection by providing meter number and details.

#### **Alternative flow**

If the actor enters wrong details the system will display a failure message and requests the actor to pay again.

### **Special requirement**

none

### **Pre-condition**

The actor received an email or message after the booking.

### **Post-condition**

If the actor wants to cancel the cylinder booking, he can cancel cylinder booking by entering his/her detail.

## Feedback/complaint

### **Brief description**

Feedback/complaint use case describes how an actor can share his/her experience of service provided by the system.

### **Actors**

The following actors are engage in the activities described in the use case:

1. Customer

### **Flow of Events**

#### **Basic Flow**

1. This use case starts when the actor is a user of the system.
2. Actor can share his/her experience of service provided by the system.

#### **Alternative flow**

none

#### **Special requirement**

none

#### **Pre-condition**

The actor is a user of system.

#### **Post-condition**

none

## Maintenance of system

### **Brief description**

Maintenance of system use case describes how an actor maintains the system.

### **Actors**

The following actors are engage in the activities described in the use case:

1. Admin

### **Flow of Events**

#### **Basic Flow**

1. This use case starts when the admin is successfully logged into the account.
2. Admin will maintain the system.

#### **Alternative flow**

If the actor enters wrong details the system will display a failure message. The system will cancel the login or throws the actor at the beginning of the basic flow.

### **Special requirement**

none

### **Pre-condition**

The actor has an account for logging into the system.

### **Post-condition**

If the actor was successfully logged into system, he can manage user's information.

## **Manage customer's information**

### **Brief description**

Manage customer's information use case describes how an actor can share his/her experience of service provided by the system.

### **Actors**

The following actors are engage in the activities described in the use case:

1. Admin

## **Flow of Events**

### **Basic Flow**

1. This use case starts when the admin is successfully logged into the account.
2. Admin will manage customer's information.

### **Alternative flow**

If the actor enters wrong details the system will display a failure message. The system will cancel the login or throws the actor at the beginning of the basic flow.

### **Special requirement**

none

### **Pre-condition**

The actor has an account for logging into the system.

### **Post-condition**

If the actor was successfully logged into system, he can manage user's information.

## PROJECT METRICS

Project Metrics are used to control and coordinate software engineering process and to improve quality of the software to be produced. Project specific metrics provide indication of productivity and insight into the technical activities. Project metrics are used by a project manager and a software team to adapt project work flow and technical activities.

### **Function Oriented Metrics**

Function Oriented Metrics use function point as normalization value. Function points are derived using an empirical relationship based on countable (direct) measure of software's information domain and assessments of software complexity.

Calculation of Value Adjustment Factors (VAF) based on responses to the following:-

Factor	Value
• Does the system require reliable backup and recovery?	1
• Are specialized data communications required to transfer information to or from the application	2
• Are there distributed processing functions?	4
• Is performance critical?	4
• Will the system run in an existing, heavily, utilized operational environment?	4
• Does the system require online data entry?	5
• Does the online data entry require the input transaction to be built over multiple screens or operations?	1
• Are the ILFs updated online?	5
• All the inputs, outputs, files, or inquiries complex?	3
• Is the internal processing complex?	1
• Is the code designed to be reusable?	3
• Are conversion and installation included in the design?	4
• Is the system designed for multiple installations in different organizations?	4
• Is the application designed to facilitate change and ease of use by the user?	4
$\sum(f_i)$	45

### Computing Function Points

<u>Information Domain Value</u>	<u>Count</u>	<u>Weighing Factor</u>	
External Inputs (EIs)	46	*	4 = 184
External Outputs (EOs)	20	*	5 = 100
External Inquiries (EQs)	0	*	4 = 0
Internal Logic Files (ILFs)	15	*	10 = 150
External Interface Files (EIFs)	1	*	7 = 7
Count Total			→ 441

$$FP = \text{Count Total} * [0.65 + 0.01 * \sum(f_i)]$$

$$\text{Function point} = 441[0.65 + 0.01 * 45]$$

$$\text{Function point} = 441[0.65 + 0.45] = 485.1$$

Therefore, 485 FP per person month

## EFFORT ESTIMATION USING COCOMO MODEL

### COCOMO Model

Barry Boehm introduced a hierarchy of software estimation models bearing the name COCOMO, for Construction Cost Model. The original COCOMO model became one of the most widely used and discussed software cost estimation models in the Industry. It has evolved into a more comprehensive estimation model, called COCOMO II. COCOMO II is actually a hierarchy of estimation models that address the following areas:

- Application Composition Model
- Early Design Stage Model
- Post- architecture- stage model

The project is based on Application Composition Model as this is used during the early stages of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance and evaluation of technology maturity are paramount.

- The COCOMO II application composition model uses object points.
- The object point is an indirect software measure that is computed using counts of number of
  1. Screens (at the user interface)
  2. Reports
  3. 3 GL component (component likely to be required to build the application)
- Each object instance is classified into one of three complexity levels i.e., simple, medium or difficult.

<b>OBJECT TYPES</b>				<b>COMPLEXITY WEIGHT</b>
	<b>SIMPLE</b>	<b>MEDIUM</b>	<b>DIFFICULT</b>	
SCREEN	1	2	3	
REPORT	2	5	8	
3GL COMPONENT			10	

- The object count is determined by multiplying the original no. of object instances by the factor and summing to obtain a total Object Point Count.
- When component based development or general software reuse is to be applied , the percent of reuse (%reuse) is estimated and the New object point is calculated using the formula  

$$\text{NOP} = (\text{object points}) * [(100\% - \text{reuse})/100]$$
 where NOP is defined as new object point
- Following table presents the productivity rate

Developer's experience/ capability	Very low	Low	Nominal	High	Very high
Environment maturity/ capability	Very low	Low	Nominal	High	Very high
PROD	4	7	13	25	50

- To derive an estimate of effort based on the computed NOP value, a "productivity rate" can be derived using formula

$$\text{PROD} = \text{NOP}/\text{Person-month}$$

- Once the Productivity rate has been determined, estimate of project effort is computed using

$$\text{Estimated effort} = \text{NOP}/\text{PROD}$$

### COCOMO Estimation for the Project

Object point	Complexity Weight		
	Simple	Medium	Difficult
Screen	2	10	3
Report	5	0	0
3 GL component			

Object point	Complexity weight		
	Simple	Medium	Difficult
Screen	2	20	9
Report	10		
Total			41

Calculating Non-Object Point(NOP)

$$\text{NOP} = (\text{Object point}) * [(100 - \% \text{ reuse})/100]$$

$$\text{NOP} = 41 * [100 - \% 0]/100$$

$$\text{NOP} = 41$$

$$\text{PROD} = \text{NOP}/\text{Person-month} \quad (\text{where NOP}=52, \text{Person-month}=4)$$

$$\text{PROD} = 41/4 = 10.25$$

Estimate effort= NOP/PROD

$$=41/10.25 = 4 \text{ person-month}$$

## **RISK ANALYSIS**

Risk analysis is the review of the risks associated with a particular event or action. It is applied to projects, information technology, security issues and any action where risk may be analyzed on quantitative and qualitative basis. It is a component of risk management.

Risks are part of every IT project and business endeavor. As such, Risk analysis should occur on a recurring basis and be updated to accommodate new potential threats. Strategic risk analysis minimizes future risk probability and damage.

### **Risk Identification:**

**Product Size:** The customer's system should have the capability to fulfill the system's size.

**Business impact:** No risk associated with constraints imposed by management.

#### **Stakeholder characteristics:**

No risk associated with the sophistication of the stakeholders and developer's ability to communicate with the stakeholders in timely.

#### **Process definition:**

The process is well defined and is followed by development organization.

#### **Development Environment:**

No risk associated with the availability and quality of the tools to be used to build the product.

#### **Staff size and experience:**

The staff size is appropriate for the project and all the members are very experienced in their own field. They all are comfortable with the technology.

### **Assessing Overall Project Risk:**

1. Have top software and customers, managers formally committed to support the project?

YES

2. Are end users enthusiastically committed to the project and the system/product to be built?

YES

3. Are requirements fully understood by software engineering team and its customers? -

YES

4. Have customers been involved fully in the definition of requirements? -  
YES
5. Do end users have realistic expectations?  
YES
6. Is the project's scope stable?  
YES
7. Does the software engineering team have the right mix of skills?  
YES
8. Are project requirements stable?  
YES
9. Does the project team have experience with the technology to be implemented? –  
YES
10. Is the number of people on the project team adequate to do job?  
YES
11. Do all customer /user constituencies agrees on the importance of the project and on the requirements for the system/product to be built?
12. YES

### RISK PROJECTION

Risk projection, also called Risk estimation, attempts to rate each risk in two ways- (1) The likelihood or probability that the risk is real and (2) the consequences of the problems associated with the risk, should it occur. The whole team works to perform four risk projection steps:

1. Establish a scale that reflects the perceived likelihood of a risk.
2. Delineate the consequences of the risk.
3. Estimate the impact of the risk on the project and the product.
4. Assess the overall accuracy of the risk projection so that there will be no misunderstandings.

The intent of these steps is to consider risks in a manner that leads to prioritization. By prioritizing risks, we can allocate resources where they will have the most impact.

## Risk Table And Impact

RISK	CATEGORY	PROBABILITY	IMPACT
1.)Size estimation may be significantly low	PS	40%	2
2.)larger number of users than planned	PS	30%	3
3.)Less reuse than planned	PS	30%	1
4.)End user resist system	BU	20%	1
5.)Delivery dead line will be tightened	BU	60%	3
6.)Funding will be lost	CU	30%	2
7.)Technology will not meet expectations	TE	50%	2
9.)Staff turnover will be high	ST	50%	2

Impact Values:

1-catastrophic

2-critical

3-marginal

4-negligible

## TESTING

Testing presents an interesting anomaly for software engineers who by their nature are constructive people. Testing requires that the developer discard preconceived notions of the "correctness" of software just developed and then work hard to design test cases to "break" the software.

### **WHITE - BOX TESTING**

White – box testing, sometimes called glass – box testing, of a software is predicated on close examination of procedural detail. Logical paths through the software and collaboration between components are tested by exercising specific sets of conditions and/or loops.

#### **Basis -- Path Testing**

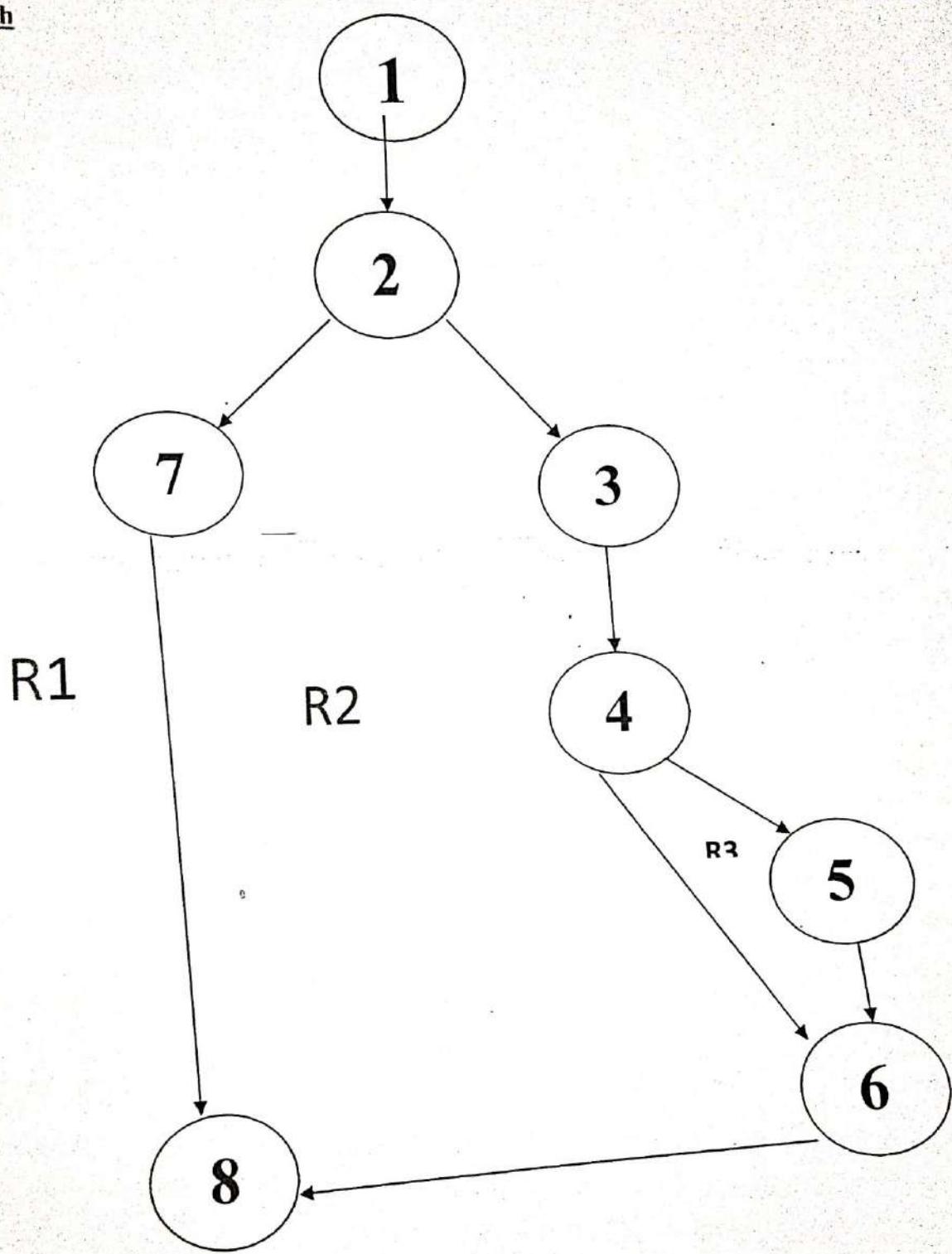
It is a white – box testing technique which enables the test case designer to derive a logical complexity measure as a guide for defining a basis set of execution paths. Test cases derived to exercise the basis set are guaranteed to execute every statement in the program at least once during testing.

#### **Whitebox testing**

We are performing whitebox testing for BOOKING screen

#### **PSEUDOCODE**

1. Enter customer ID
2. If customer ID is valid
  3.       Display booking portal
  4.       If payment mode= online
  5.       Display payment portal
  - End if
6. End if
7. Else
  - Print "invalid ID"
8. End code

Flow Graph

### CYCLOMATIC COMPLEXITY OF RESULTANT GRAPH:

$V(G) = 3$  regions

$V(G) = 9 \text{ edges} - 8 \text{ nodes} + 2 = 3$

$V(G) = 2 \text{ predicate nodes} + 1 = 3$

### LINEARLY INDEPENDENT PATHS FOR FLOW GRAPH

Path 1: 1-2-7-8

Path 2: 1-2-3-4-6-8

Path 3: 1-2-3-4-5-6-8

### WHITEBOX TESTING

TEST CASE ID	INPUT	ACTUAL OUTPUT	EXPECTED OUTPUT
1.	Customer id=valid	To be observed after the execution	Display booking portal
2.	Customer id=invalid	To be observed after the execution	Show error message "invalid ID"
3.	Payment option=online	To be observed after the execution	Go to payment portal
4.	Payment option=COD	To be observed after the execution	Confirm booking

## **BIBLIOGRAPHY**

1. R.S. Pressman, Software Engineering: A Practitioner's Approach (7<sup>th</sup> Edition), Mc Graw-Hill, 2009.
2. P.Jalote, An integrated Approach to Software Engineering (2<sup>nd</sup> Edition), Narosa Publishing House, 2003.
3. IEEE Software Engineering Standards Committee "IEEE Standard 29148:2011, IEEE Recommended Practice for Software Requirements Specifications".