**PROJECT REPORT ON**

ONLINE GAS BOOKING

**SUBMITTED BY**

PRATIK ANIL BHOSALE.(7802)

**UNDER THE GUIDENCE**

Mrs. Edith Michael.

**ACKNOWLEDGEMENT**

I would like to express my thanks to the people who have helped me most throughout my project.

I sincerely thank to my project guide teacher “Edith Ma’am” for guidance and encouragement in carrying out this project work.

A special thanks of mine goes to my colleagues who helped me out in testing/debugging the project and with whom I exchanged lot of interesting ideas, thoughts and which made it possible to complete my project.

I am highly indebted to Ruia College for their guidance and constant supervision as well as for keeping me track regarding the project & also for their support in completing the project.

At last but not least I wish to avail myself of this opportunity, express a sense of gratitude and love to my friends and my beloved parents for their manual support, strength and help for everything.

Pratik A. Bhosale.

**PRELIMINARY INVESTIGATION**

* **Topics Ahead : -**
* Organizational Overview
* Description of Existing System
* Limitation of current System
* Proposed System
* Advantages Of Proposed System
* Feasibility Study
* Stakeholders
* Technology Used
* Gantt Chart
* **ORGANIZATIONAL OVERVIEW**

This is an online gas booking system for customers that allows them to order gas cylinders whenever required. Users can now directly order gas cylinder and can also make payment instead of using conventional approach of ordering through phone.

The conventional system takes long time for processing request and users have no other option but to wait. This project eliminates all these drawbacks. There will be customer login and admin login. Admin has the control of all the users account and their details. There will be 12 cylinders initially allotted to all customer accounts along with their cost that are valid only for one year, and would require more then he will get notification.

The final part of the project is for generating the report of Customer details and his requirements. This includes the total evaluation of **ONLINE GAS BOOKING.**

* **DESCRIPTION OF SYSTEM:-**
* In a Current System User can book gas through only register phone number.
* They did not get notification about subsidy.
* Users have to wait long time for processing request.
* Low speed of websites.
* **LIMITATION OF CURRENT SYSTEM:-**
* Booking can be done only through register mobile number.
* User cannot identify the exact amount for each cylinder.
* User is not aware of how much cylinders are remaining.
* User did not get subsidy notification.

* **PROPOSED SYSTEM:-**
* This website open 24\*7 to receive booking.
* In this system online booking can be done.
* No hassle of travelling to gas agency or constant follow with the distributor.
* There is no need for users and agency to maintain gas records manually. They can view the details through their account.
* This system saves time, efforts and cost.
* **ADVANTEGES OF PROPOSED SYSTEM:-**
* Safe and convenient method of booking .
* Simply good Customer Service.
* In this System User can get subsidy notification.
* In this System can easily maintain the record.
* This system saves time, efforts and cost.
* Simplified login.
* User Friendliness.
* Data security.
* Online payment can be done.

* **FEASIBILITY STUDY:-**

Feasibility study consideration of all possible ways to provide solution to given problem . The proposed should satisfy all the user requirements and should be flexible enough so that future changes can be easily done by future upcoming requirements.

1. **Economic Feasibility:**

* The project doesn’t need much money for development. The maintenance of the system is also economically feasible.

1. **Technical Feasibility:**

* The project is technically feasible as there isn’t any difficulty to get required resources. All the technology used is simple.

1. **Operational Feasibility:**

* The system requires minimum knowledge and understanding of the computer. The system can be easily used by the user.

* **STAKEHOLDERS:-**

1. **ADMIN:-**

* Admin can update data in system.
* Admin updates the websites and checks the feedback and registration from the user.

1. **CUSTOMER:-**

* User can view information about cylinder subsidy.
* User can read the info about gas Services & feel the feedback form.

* **PROJECT REQUIREMENTS:-**
* Hardware requirements.
* No Specific Hardware Requirement.
* Software Requirements
* Browser:-Mozilla Firefox ,Google chrome .
* **Front End:-ASP.NET**
* **Back End:-SQL SERVER 2012.**
* **GANTT CHART:-**

**SYSTEM ANALYSIS**

* **TOPICS AHEAD:-**
* Fact Finding Techniques
* Prototypes
* Event Table
* Use Case Diagram
* Use Case Scenario
* Use case Description
* ERD
* Activity Diagram
* Class Diagram
* Sequence Diagram /Collaboration Diagram
* State Diagram
* **FACT FINDING USING QUESTIONAIRES:**

1. Currently, how the data is stored?
2. When the data is updated?
3. What kind of data is stored in the system?
4. Who can use the system?
5. How the system works?

* **EVENT TABLE :-**

A table that lists events in rows and key places of information about each event in column.

* **Event:-**

It causes a system to do something.

* **Source:-**

An external Agent or Actor that supplies Data to the system.

* **Trigger:-**

An occurrence that tells the system that an event has occurred, either the arrival of data needing processing or of a point in time.

* **Activity:-**

Behavior that the system performs when an event occurs.

* **Response:-**

An output , produced by the system that goes to a destination.

* **Destination:-**

An external Agent or Actor that receives data from the system .

* **Event table :- CUSTOMER**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Event** | **Trigger** | **Source** | **Activity** | **Response** | **Destination** |
| 1. | Customer create an account | Creating new account | customer | Created new account | Display account details | customer |
| 2. | Customer wants to login | Request to login | customer | Verify authentication | Login | customer |
| 3. | Apply for new connection | Request to new connection | customer | Created new connection | Display the new connection Details | System |
| 4. | Gas Booking | Request for gas Booking | customer | Gas Booked | Gas Booked Details | customer |
| 5. | Transaction Details | Request for transaction | customer | Transaction successful | Display Receipt | customer |
| 6. | Subsidy Notification | Get subsidy | customer | View Subsidy details | Display Subsidy Details | customer |
| 7. | Cancel Booking | Cancellation of Booking | customer | Booking Canceled | Confirmation about Cancellation | customer |
| 8. | Feedback Form | Request for Feedback form | customer | Display feedback form | Feedback details | customer |
| 9. | Changes in personal data | Request for updating information | customer | Updating information | Update details | customer |
| 10. | Customer Logout | Request logout | customer | Logout successfully | logout | system |

* **EVENT TABLE:- ADMIN**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Event** | **Trigger** | **Source** | **Activity** | **Response** | **Destination** |
| 1. | Admin Login | Request for login | Admin | Verify authentication | Login | Admin |
| 2. | Add new connection Details | Adding new connection | Admin | Assign new ID to customer | New connection Added | customer |
| 3. | View gas Booking Details | Request to check details | Admin | Display Booking | Booking details | customer |
| 4. | Checks customer subsidy | Request to check subsidy | Admin | Sending notification about subsidy to customer | Subsidy details | Admin |
| 5. | Update customer account | Request to customer account  details | Admin | Display account details | Account details | Admin |
| 6. | Admin views customer details | Request to view details | Admin | Display customer information | Customer details | Admin |
| 7. | Admin feedback form | Request to display feedback | Admin | Display feedback | Feedback  details | Admin |
| 8. | Admin wants to delete feedback | Request to delete feedback | Admin | Delete feedback | Feedback deletion details | Admin |
| 9. | Admin wants to logout | Request to logout | Admin | Logout | Logout | Admin |

* **Use Case Diagram:-**

Use Case diagram is used to identify the “uses” or uses Use cases of the new system -- in other words , to identify how the system will be used . The use case diagram is essentially an extension of the event Table.

* **USE CASE:-**

It describes an activity the system carries out in response to an event.

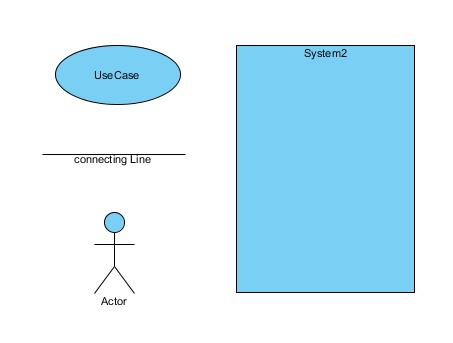
* **ACTOR:-**

In UML ,the person involved is called an Actor .An Actor is always outside of the Automation Boundary of the system.

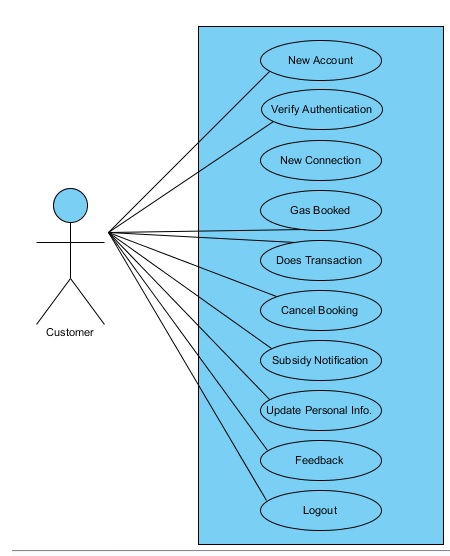
* **CONNECTION LINE:-**

The arrow is used to show which Actors participate in which Use cases.

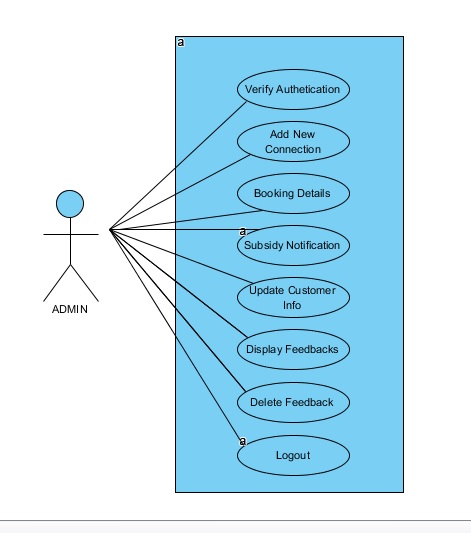
* **USE CASE NOTATION:-**

****

* **USE CASE DIAGRAM:-CUSTOMER(ACTOR WISE)**



* **USE CASE DIAGRAM :-ADMIN(ACTOR WISE)**

****

* **USE CASE SCENARIO:-**

1. **Use case name:** New account

**Scenario:** Customer wants to create a new account

**Actors:** Customer

**Triggering event:** New account

**Flow of activities:**

1. Customer wants to create a new account.
2. Customer enters details.

**Pre-Condition:** Customer do not have an account

**Post-Condition:** Customer successfully created new account.

1. **Use case name:** Verify authentication.

**Scenario:** Customer wants to login.

**Actors:** Customer

**Triggering event:** Request to login

**Flow of activities:**

1. Customer enters login Details.
2. Customer logs in to system.

**Pre-Condition:** Customer is not logged in.

**Post-Condition:** Customer successfully logged in.

1. **Use case name:** New Connection.

**Scenario:** Customer wants a new gas connection.

**Actors:** Customer

**Triggering event:** Request to new connection.

**Flow of activities:**

1. Customer enters new connection Details.

2. Customer gets new connection.

**Pre-Condition:** Customer do not have new connection.

**Post-Condition:** Customer successfully get new connection details.

1. **Use case name:** Gas Booking .

**Scenario:** Customer wants to book a gas .

**Actors:** Customer

**Triggering event:** Request to gas booking.

**Flow of activities:**

1. Customer enters Gas booking Details.

2. Customer Book gas.

**Pre-Condition:** Customer is not book gas .

**Post-Condition:** Customer successfully booked gas.

1. **Use case name:** Transaction Details.

**Scenario:** Customer wants to make payment.

**Actors:** Customer

**Triggering event:** Request for transaction

**Flow of activities:**

1. Customer enters Transaction Details.

2.Customer make payment.

**Pre-Condition:** Customer does not make payment.

**Post-Condition:** Customer successfully make payment.

1. **Use case name:** Subsidy Notification.

**Scenario:** Customer gets subsidy notification.

**Actors:** Customer

**Triggering event:** get subsidy notification.

**Flow of activities:**

1**.** Customer view subsidy notification details.

**Pre-Condition:** Customer book gas.

**Post-Condition:** Customer gets subsidy notification and view subsidy details.

1. **Use case name:** Cancel Booking.

**Scenario:** Customer wants to cancel booking.

**Actors:** Customer

**Triggering event:** Request for cancel booking.

**Flow of activities:**

1. Customer enters cancel booking Details.
2. Customer gets confirmation about cancel booking.

**Pre-Condition:** Customer Book gas.

**Post-Condition:** Customer successfully canceled a booking.

1. **Use case name:** Feedback form.

**Scenario:** Customer wants to give feedback.

**Actors:** Customer

**Triggering event:** Request for feedback.

**Flow of activities:**

1. Customer submits feedback form.

**Pre-Condition:** Customer logged in account.

**Post-Condition:** Customer successfully give feedback.

1. **Use case name:** update account information .

**Scenario:** Customer wants to change in personal details.

**Actors:** Customer

**Triggering event:** Request for updating information.

**Flow of activities:**

1. Customer update personal info.

**Pre-Condition:** Customer logged in system.

**Post-Condition:** Customer successfully updated account.

**10) Use case name:** customer logout.

**Scenario:** Customer wants to logout.

**Actors:** Customer

**Triggering event:** Request to logout.

**Flow of activities:**

1. Customer logs out from system.

**Pre-Condition:** Customer logged in system.

**Post-Condition:** Customer successfully log out from system.

**11)Use case name:** Verify Authentication.

**Scenario:** Admin wants to login.

**Actors:** Admin

**Triggering event:** Request for login.

**Flow of activities:**

1. Admin enters login details.

2. Admin logs into system.

**Pre-Condition:** Admin is not logged in.

**Post-Condition:** Admin successfully logged in.

**12) Use case name:** Add new connection Details.

**Scenario:** Admin wants to add new connection details.

**Actors:** Admin

**Triggering event:** Adding new connection details.

**Flow of activities:**

1. Admin verify the customer details.

2. Admin add new connection and assign id to the customer.

**Pre-Condition:** Admin logged in system.

**Post-Condition:** Admin successfully add new connection details.

**13) Use case name:** Gas Booking Details.

**Scenario:** Admin wants to view gas booking details.

**Actors:** Admin

**Triggering event:** Request to check details.

**Flow of activities:**

1. Admin view the customer gas booking details.

**Pre-Condition:** Admin logged in system.

**Post-Condition:** Admin successfully view the customer booking details.

**14)Use case name:** Subsidy notification.

**Scenario:** Admin sends subsidy notification details to customer.

**Actors:** Admin

**Triggering event:** Request to check subsidy details.

**Flow of activities:**

1. Admin checks subsidy details.

2. Admin successfully sends a subsidy notification to the customer.

**Pre-Condition:** Admin checks subsidy details .

**Post-Condition:** Admin successfully sends a subsidy notification .

**15) Use case name:** Update account .

**Scenario:** Admin wants to update account .

**Actors:** Admin

**Triggering event:** Request to the customer account details.

**Flow of activities:**

1. Admin update account.

**Pre-Condition:** Admin select a customer account .

**Post-Condition:** Admin successfully update account.

**16) Use case name:** Display Feedbacks

**Scenario:** Admin wants to check outs feedback.

**Actors:** Admin

**Triggering event:** Request to display feedback details.

**Flow of activities:**

1. Admin view feedbacks .

**Pre-Condition:** Admin logged in system.

**Post-Condition:** Admin successfully view feedbacks.

**17) Use case name:** Delete feedback .

**Scenario:** Admin wants to delete feedback.

**Actors:** Admin

**Triggering event:** Request to delete feedback.

**Flow of activities:**

1. Admin selects feedback.

2. Admin deletes feedback.

**Pre-Condition:** Admin views feedback.

**Post-Condition:** Admin successfully deletes selected feedbacks.

**18) Use case name:** Admin logout .

**Scenario:** Admin wants to logout from system.

**Actors:** Admin

**Triggering event:** Request to logout .

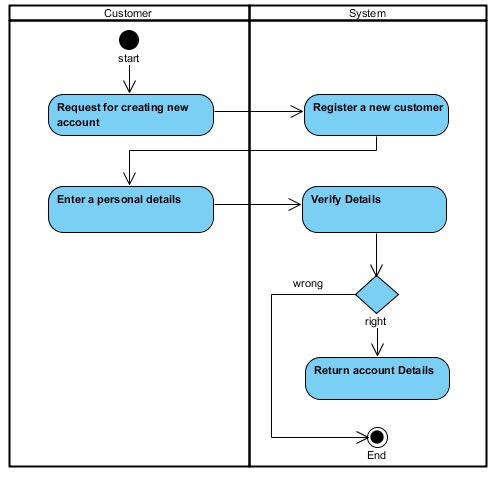
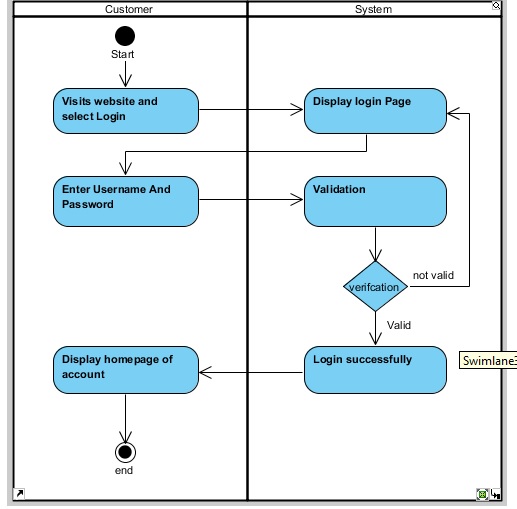
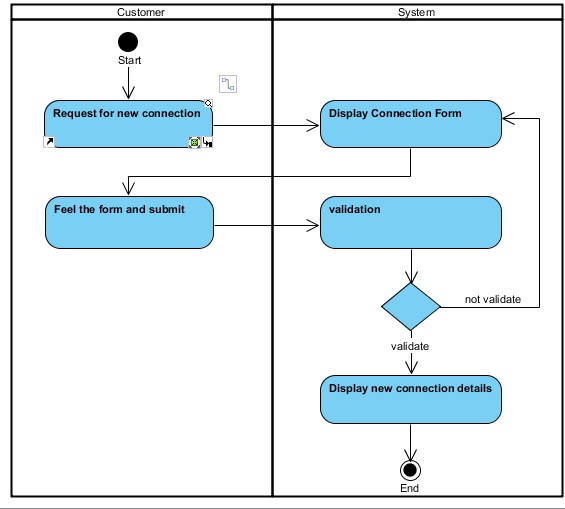
**Flow of activities:**

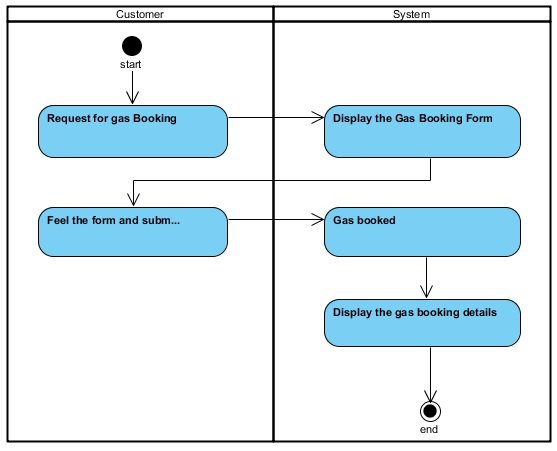
1. Admin logs out from system .

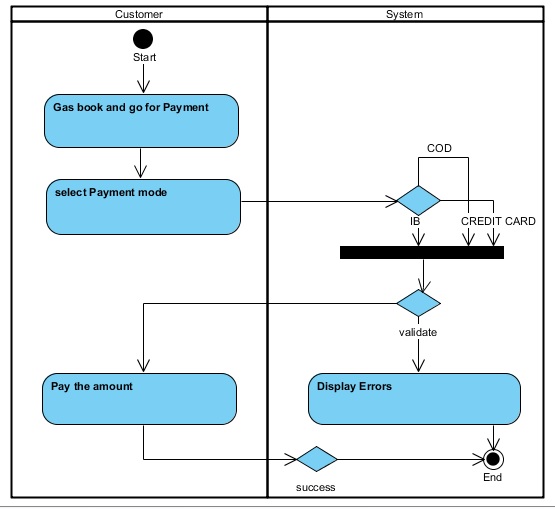
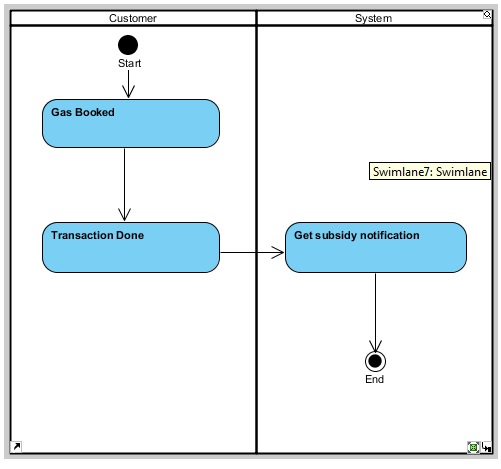
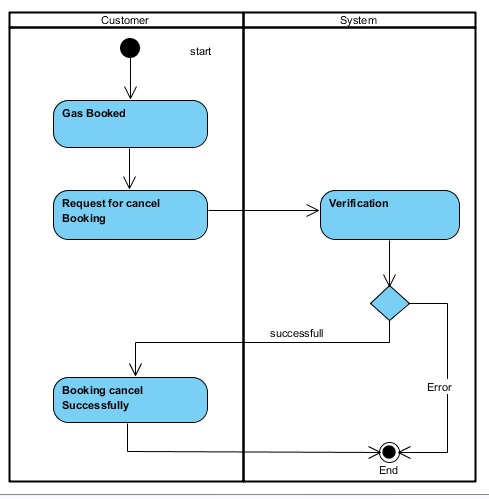
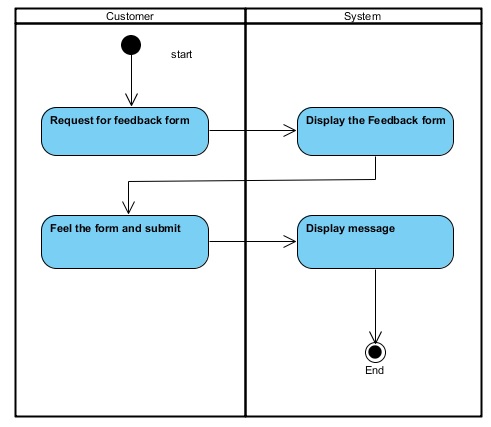
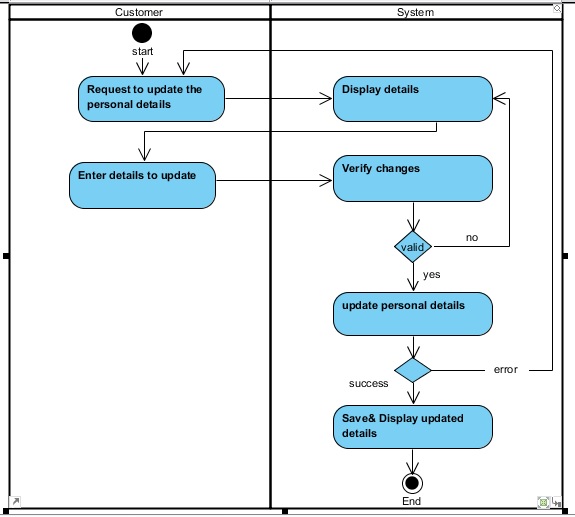
**Pre-Condition:** Admin logged in system.

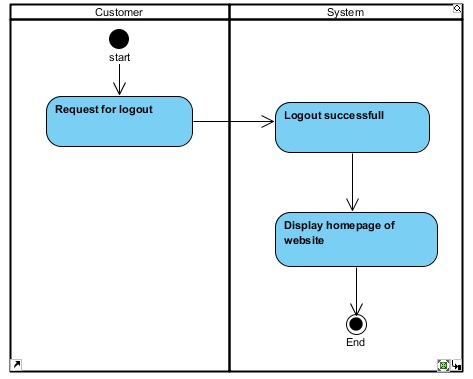
**Post-Condition:** Admin successfully logout from system .

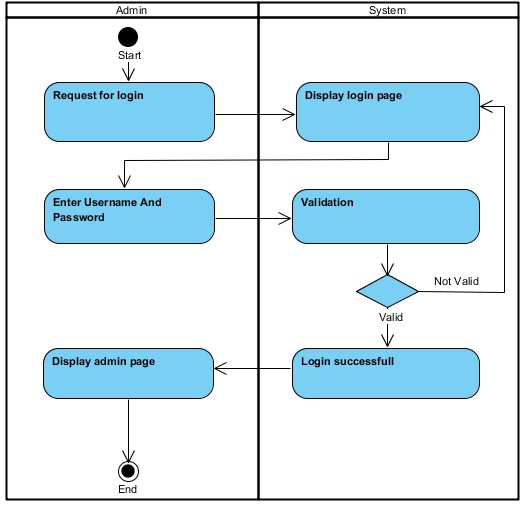
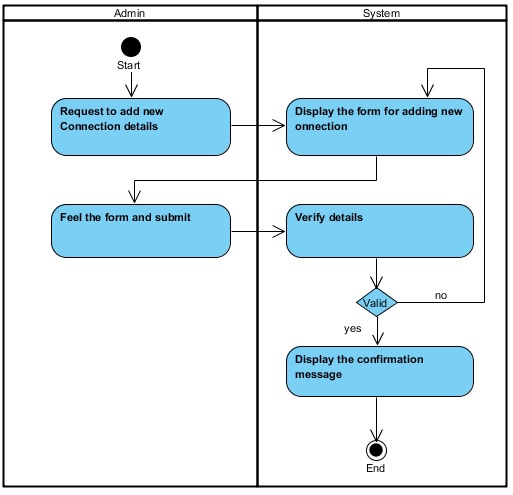
* **ACTIVITY DIGRAM (Event wise):-**

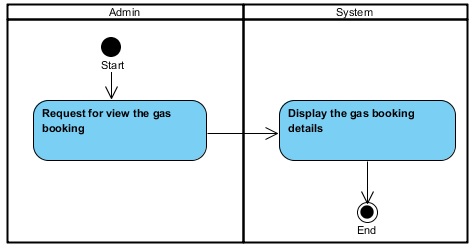
1. **Customer create an account :-** 
2. **Customer wants to login:- **
3. **Apply for new connection:- **
4. **Gas Booking:-**

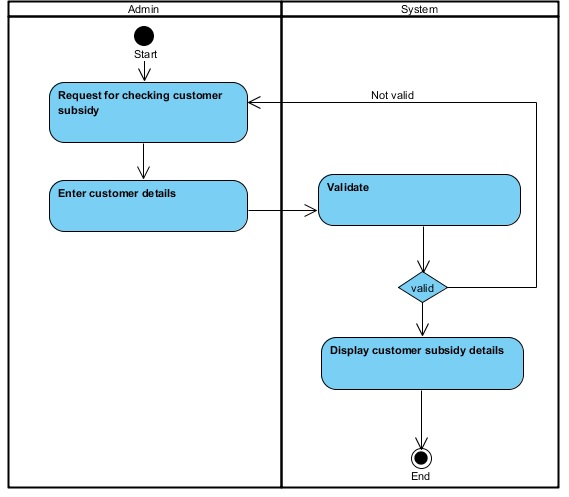
****

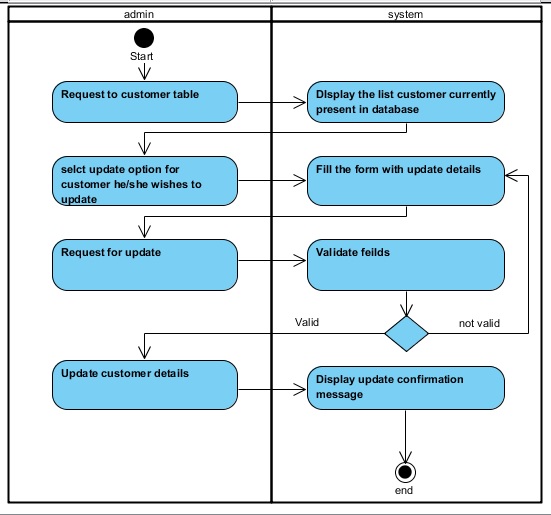
1. **Transaction Details:- **
2. **Get subsidy notification:-**
3. **Cancel Booking:-** 
4. **Feedback Form:-** 
5. **Change in personal details:- **

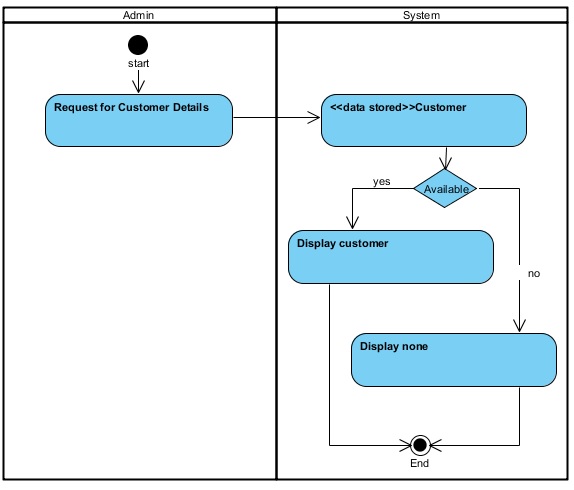
**10.Customer Logout:-** 

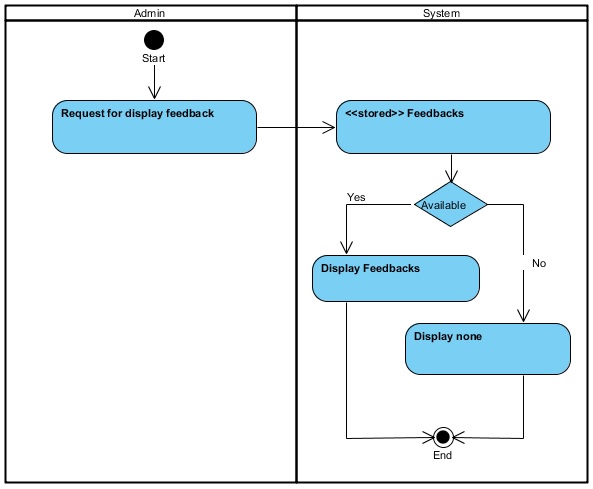
**11.Admin Login:- 12.Add New Connection Details:- **

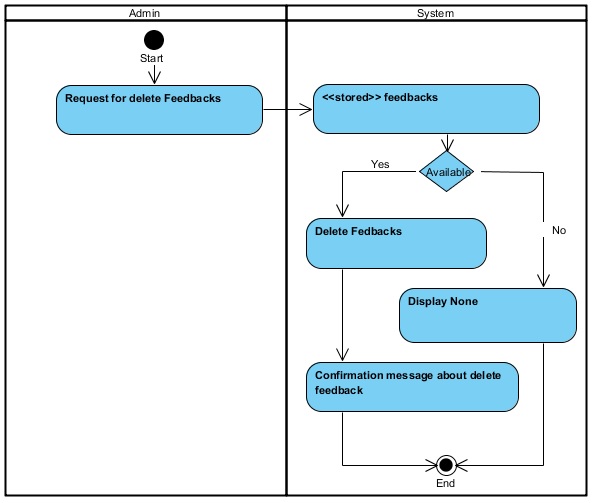
**13.View the Gas Booking Details:-**

**14.Checks Customer Subsidy:-**

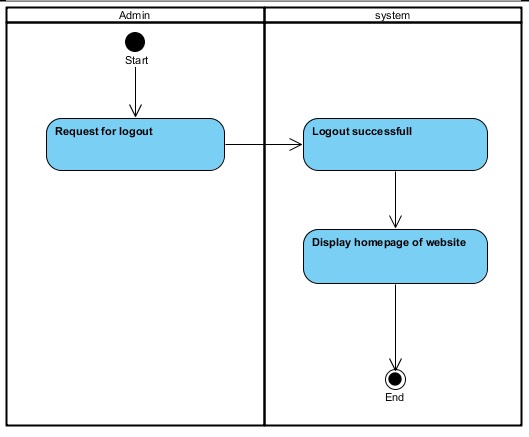
**15.Update Customer Account:-**

**16.Admin Views Customer Details:-**

**17.Admin Views Feedbacks:- **

**18.Admin wants to delete feedbacks:-**

**19.Admin Logout:-**

****

* **CLASS DIAGRAM:-**

The class diagram are used to identify and classify the objects which constitute a system. It also includes the important attributes of the objects which must be capture.

* **CLASS:-**

It is a collection of objects of same type.

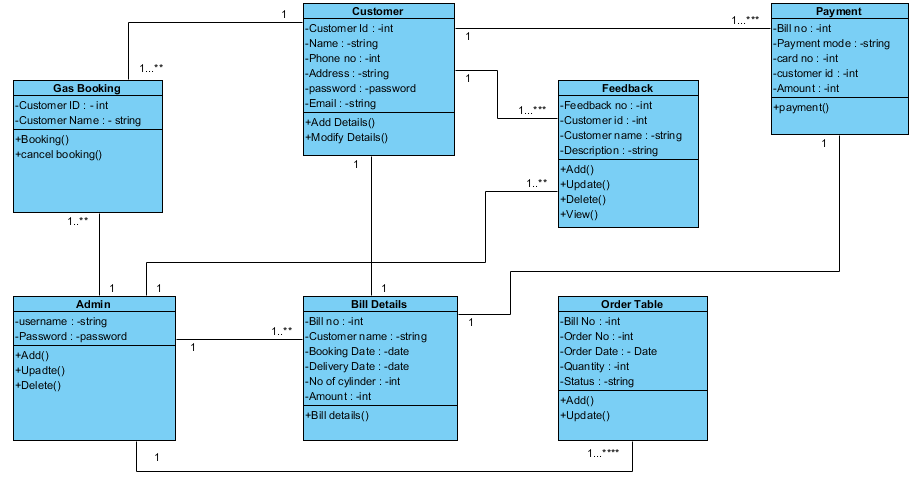
* **RELATIONSHIP:-**

A naturally occurring association among specific things.

* **DESCRIPTION:-**

It is model which is used to show the classes constituting a system and their interrelationship .It is based on UML . Only the important attributes and methods are shown in class Diagrams. In the initial period of analysis , the important attributes of the classes, which must be captured and functionalities provided by the class may not be very clear . As the analysis progresses, the attributes and methods may be added. If more focus is on interrelationship of classes, then the attributes and methods may not be shown in the class diagram.

* **CLASS DIAGRAM:-**

****

* **SEQUENCE DIAGRAM:-**

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time.  A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

* **Class Roles or Participants**

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

* **Activation or Execution Occurrence**

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.

* **Messages**

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

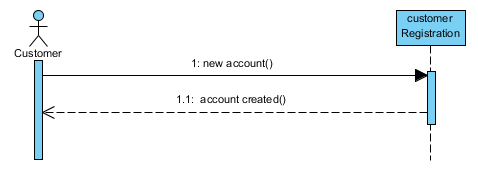
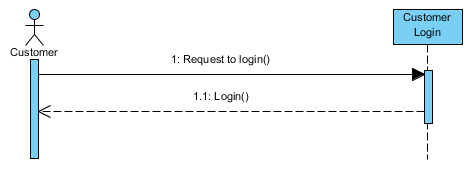
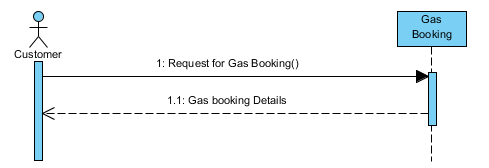
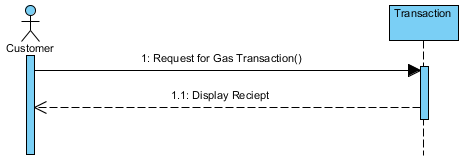
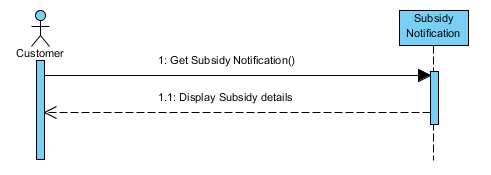
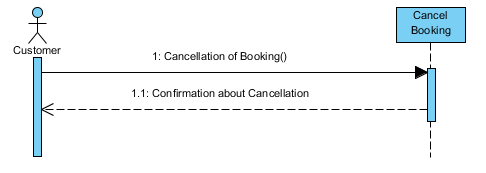
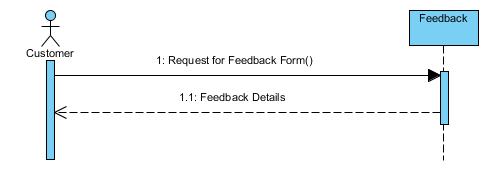
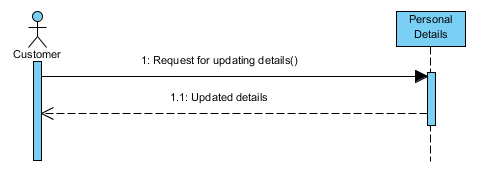
* **Lifelines**

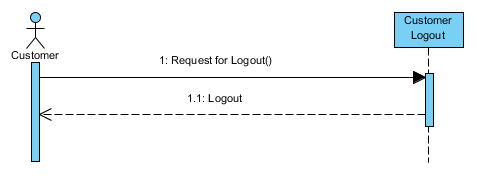
Lifelines are vertical dashed lines that indicate the object's presence over time.

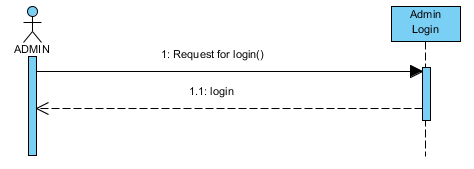
* **Destroying Objects**

Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

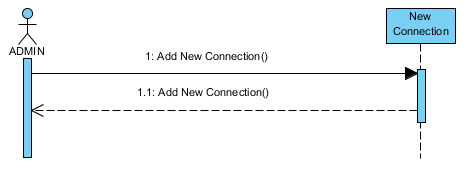
* **SEQUENCE DIAGRAM(Actor wise):-**

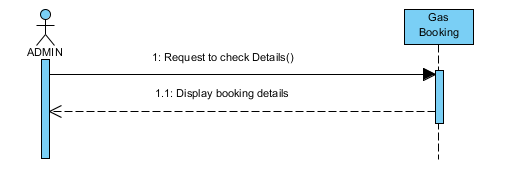
1. Customer creating new account:-
2. Verify Authentication(customer login):-
3. Customer Apply for new connection:-
4. Gas Booking:-
5. Transaction details:-
6. Get subsidy Notification:-
7. Cancel Booking:-
8. Feedback Form:-
9. Change in Personal Details:-
10. Logout:-

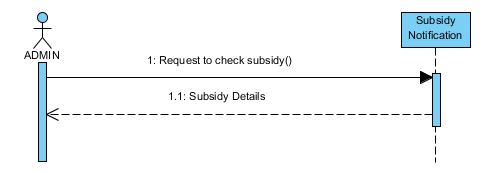
11. Verify Authentication (Admin Login):-

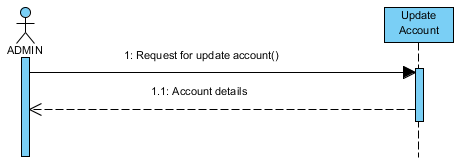


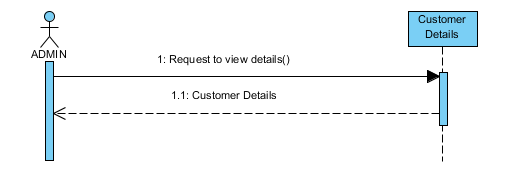
12.New Connection Details:-

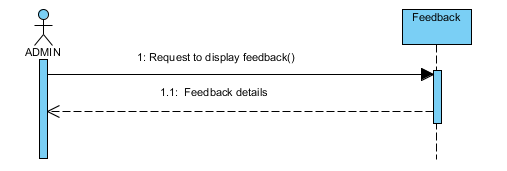


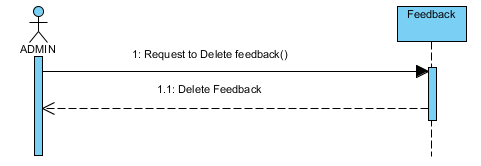
13.View Gas Booking Details:-

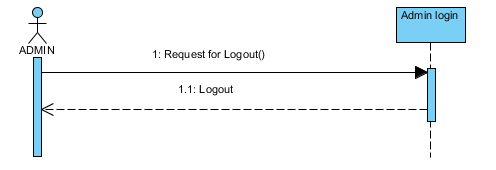
14. Checks Customer subsidy:-

15. Update customer account:-

16. View Customer Details:-

17. Feedback Form:-

18. Delete Feedback:-

19. Admin Logout:-

* **STATE DIAGRAM:-**

A state diagram shows the behavior of classes in response to external stimuli. Specifically a state diagram describes the behavior of a single object in response to a series of events in a system

State diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State diagram is to model lifetime of an object from creation to termination.

* **States**

States represent situations during the life of an object. You can easily illustrate a state in smart draw by using a rectangle with rounded corners.

* **Transition**

A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it. A state can have a transition that points back to itself.

* **Initial State**

A filled circle followed by an arrow represents the object's initial state.

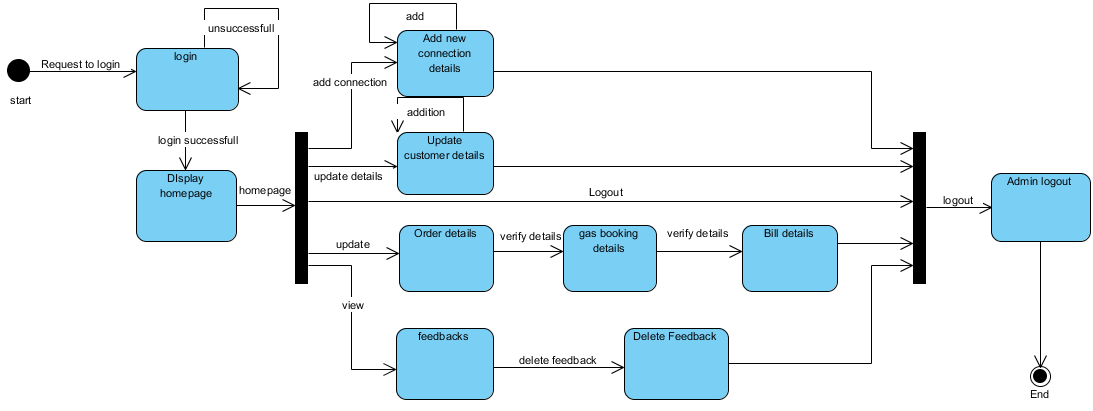
* **Final State**

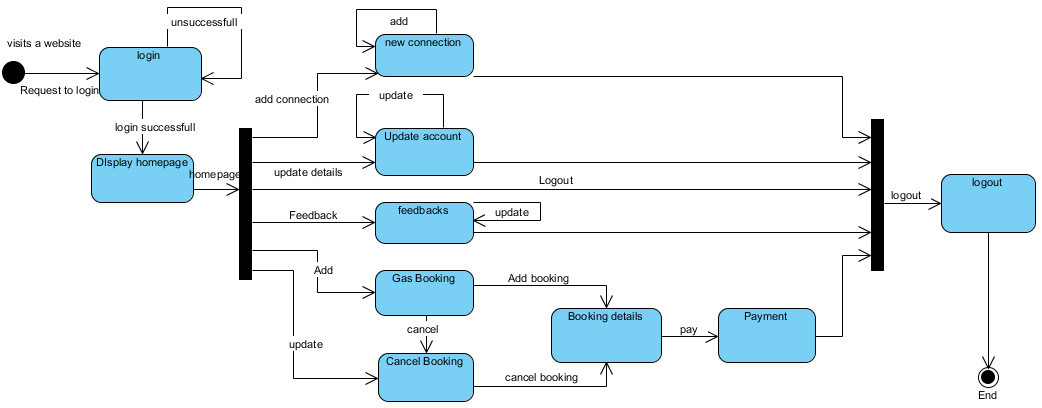
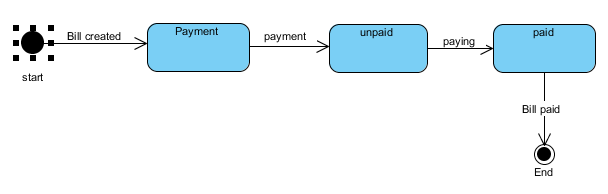
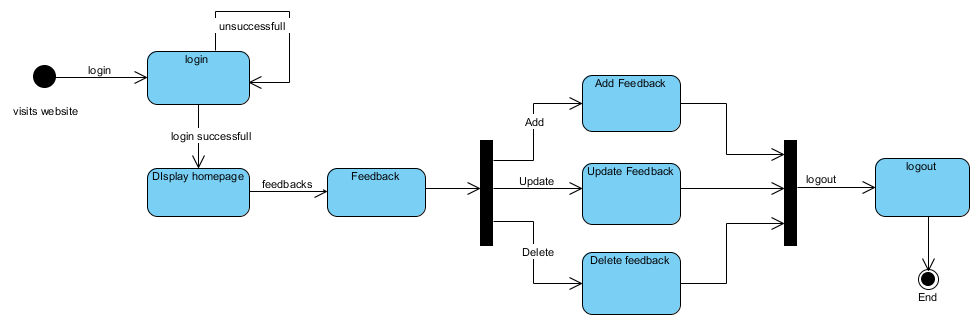
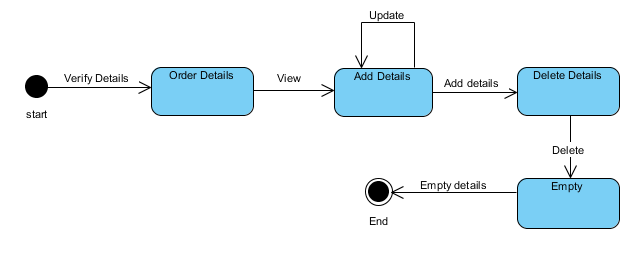
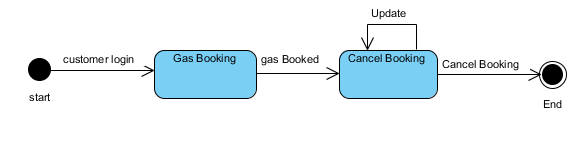
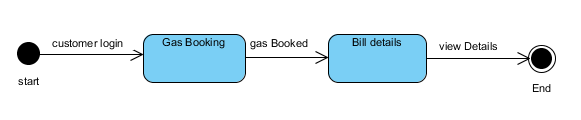
An arrow pointing to a filled circle nested inside another circle represents the object's final state.

* **Synchronization and Splitting of Control**

A short heavy bar with two transitions entering it represents a synchronization of control.

* **STATE DIGRAM FOR ADMIN:-**

****

* **STATE DIAGRAM FOR CUSTOMER:-**
* **STATE DIGRAM FOR PAYMENT:-**
* **STATE DIGRAM FOR FEEDBACK:-**
* **STATE DIAGRAM FOR ORDER DETAILS:-**
* **STATE DIAGRAM FOR GAS BOOKNG:-**
* **STATE DIGRAM FOR BILL DETAILS:-**

**THANK YOU**