2581-4575
International Journal For Recent Developments in Science & Technology
Crossref
A Peer Reviewed Research Journal

# VEHICARE CLOUD APPOINTMENT

**[1,2,3,4] Siddhartha Vallapoju, Revoju Charan, Minnikanti Jai Sriharsha Vardhan, Siripuram Prudhvinath,[5]V. Anil Kumar**

[1,2,3,4] Ug scholars, MallaReddy college Of Engineering , Hyderabad - 500100
[5] Assistant Professor, MallaReddy college Of Engineering , Hyderabad - 500100

## ABSTRACT

High-speed networks and ubiquitous Internet access become available to users for access anywhere at any time. Cloud computing is a concept that treats the resources on the Internet as a unified entity, a cloud.

This paper presents the development of an intelligent chatbot system utilizing AWS Lex for efficient appointment scheduling in the context of Vehicle Servicing. The proposed system aims to streamline the appointment booking process by engaging users in a natural language conversation, collecting necessary details, and seamlessly integrating with backend systems to confirm and schedule appointments.

The system further enhances user experience by sending acknowledgments through SMS or email, providing real-time updates and confirmation to both users and the service provider.

# CHAPTER - 1 INTRODUCTION

## 1.1 INTRODUCTION

In an era of increasing automation and digital transformation, our project is dedicated to revolutionizing the vehicle servicing experience. Our objective is to introduce an innovative solution by leveraging the power of AWS LEX chat bot technology, which will seamlessly facilitate and automate the booking of vehicle servicing appointments. This project not only aims to enhance user convenience but also optimize operational efficiency within the automotive service industry."

## 1.2 OBJECTIVE

The objectives of a Vehicle Service Appointment Booking System typically revolve around improving service efficiency, customer experience, and operational management within a service centre.

## 1.3 METHODOLOGY ADOPTED

The methodology adopted for developing a Vehicle Service Appointment Booking System could depend on various factors such as team expertise, project scope, timelines, and specific requirements. However, a common methodology might include:

Agile Methodology

# CHAPTER - 2

# LITERATURE SURVEY

## 2.1 LITERATURE SURVEY

**Literature survey 1: -**

**Title: -** "Bot-Based Cloud Management: Challenges and Opportunities"

**Year: -**2019

**Authors: -** M. Rehman, et al.

**Abstract: -** This paper discusses the use of chatbots and conversational agents for cloud management, addressing challenges and opportunities in automating cloud-related tasks and support.

**Literature survey 2: -**

**Title: -** "CloudBot: An Intelligent Cloud Management Chatbot"

**Year: -** 2018

**Authors: -** P. Halder, et al.

**Abstract: -** This research paper presents CloudBot, an intelligent chatbot designed for cloud management tasks and demonstrates its effectiveness.

# CHAPTER - 3

# SYSTEM ANALYSIS

## 3.1 EXISTING SYSTEM

**phone calls:**
Many vehicle servicing centers still rely on traditional phone calls for appointment booking.

**Online Booking Portals:**
Some servicing centers offer online booking portals on their websites. Customers can book their appointment through online

**Service Center Management Software:**
Internally, servicing centers may use management software that includes appointment scheduling features.

## 3.2 DRAWBACKS

**Limited availability:**
Phone-based booking systems and in-person scheduling may have limited availability during certain hours, leading to potential delays and inconvenience for customer.

**Human errors:**
Manual booking systems, whether over the phone or in person, can be prone to human errors such as incorrect appointment times, service etc.

## 3.3 PROPOSED SYSTEM

The proposed system of project, which involves building an AWS Lex chatbot to automate vehicle servicing appointment bookings for customers and service providers

SYSTEM OVERVIEW:

Chatbot Interface: The system will feature an intuitive and user-friendly chatbot interface that can be accessed through various channels, such as a website, mobile app, or messaging platform

# CHAPTER - 4

# SYSTEM DESIGN

## 4.1 SYSTEM ARCHITECTURE

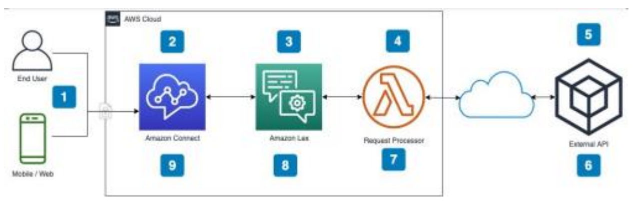


Fig4.1.1.Chatbot Architecture

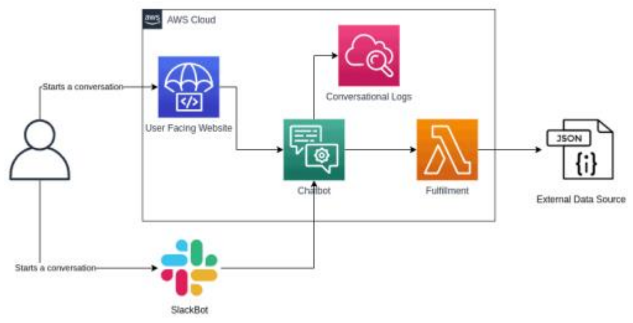


Fig4.1.2.Integrated With third Party Source

## 4.2 MODULES

1. User Interface Module
2. Backend Integration Module

**1.User Interface Module:**

- Develop the chatbot's user interface, enabling users to interact naturally.

- Implement a text or voice-based interface that understands user requests.

- Utilize Amazon Lex and Amazon Polly for natural language processing and speech synthesis.

**2. Backend Integration Module:**

- Establish connections with the backend systems of vehicle service providers.

## 4.3 UML DIAGRAMS

**DATA FLOW DIAGRAM:**

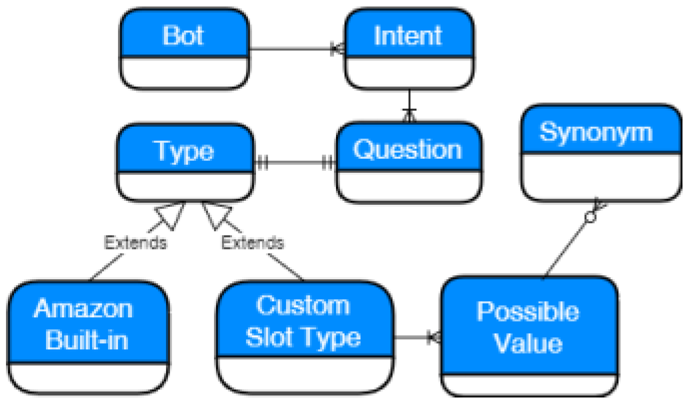


Fig4.3.1

**SEQUENCE DIAGRAM:**

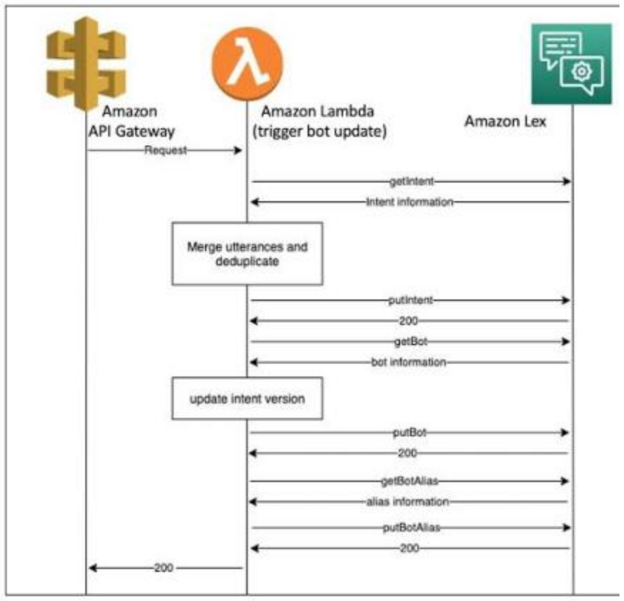


Fig4.3.5. Sequence diagram

## USE CASE DIAGRAM

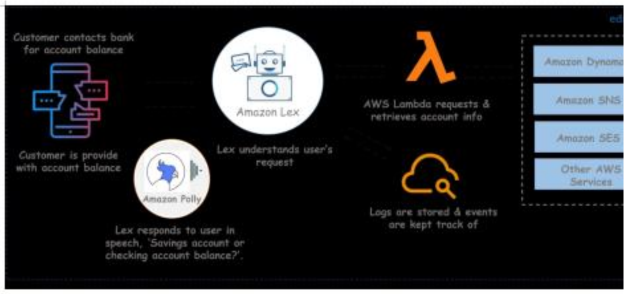

Customer contacts bank for account balance
Customer is provide with account balance
Amazon Lex
Lex understands user's request
Amazon Polly
Lex responds to user in speech, 'Savings account or checking account balance?'.
AWS Lambda requests & retrieves account info
Logs are stored & events are kept track of
Amazon Dynamo
Amazon SNS
Amazon SES
Other AWS Services

Fig.4.3.7 use case diagram

# CHAPTER - 5

# SYSTEM IMPLEMENTATION

## 5.1 CLOUD COMPUTING:

Cloud computing is a technology that enables access to a shared pool of computing resources over the internet. Instead of owning physical hardware or running software on a local server, users can access applications, storage, and processing power hosted by a third-party provider.

Here are some key components and concepts:

**Models of Cloud Computing:**

1. **Infrastructure as a Service (IaaS):** Provides virtualized computing resources over the internet. Users can rent virtual machines, storage, and networking resources on a pay-as-you-go basis.

## 5.2 SOURCE CODE

**Language used: Json Format**

1.MANIFEST:

```
 {"metaData":{"schemaVersion":"1","fileFormat":"LexJson","resourceType":"BOT"}}
```

2.BOT:

```

{"description":null,"identifier":"KJCPX5WDA9","name":"VEHICARE","version":"DRAFT",
"dataPrivacy":{"childDirected":false},"idleSessionTTLInSeconds":600}
```

3.BOTLOCALE:

4.INTENTS:

4.1 BIKESERVICEAPPOINTMENT:

```
{"name":"Bikeserviceappointment","identifier":"FA8BJ4OYNN","description":null,"parentInt
entSignature":null,"sampleUtterances":[{"utterance":"hi"},{"utterance":"hello"},{"uttera
nce":"Hi"},{"utterance":"Hello"}],"intentConfirmationSetting":{"isActive":true,"codeHook
":{"isActive":true,"enableCodeHookInvocation":true,"invocationLabel":null,"postCodeHookS
pecification":{"failureResponse":null,"failureNextStep":{"intent":{"name":null,"slots":n
ull},"sessionAttributes":null,"dialogAction":{"type":"EndConversation","slotToElicit":nu
ll,"suppressNextMessage":null}},"failureConditional":null,"successResponse":null,"succes
sNextStep":{"intent":{"name":null,"slots":null},"sessionAttributes":null,"dialogAction":
{"type":"FulfillIntent","slotToElicit":null,"suppressNextMessage":null}},"successConditi
onal":null,"timeoutResponse":null,"timeoutNextStep":{"intent":{"name":null,"slots":null},
"sessionAttributes":null,"dialogAction":{"type":"EndConversation","slotToElicit":null,"s
```

6.LAMBDA CODE FOR DATABASE:

python

import boto3

```
def lambda_handler(event,
context):
    # Extract relevant
information from the Lex
event
    user_id = event['userId']
    input_text =
event['inputTranscript']

    # Connect to DynamoDB
    dynamodb =
boto3.resource('dynamodb')
    table =
dynamodb.Table('YourDyna
moDBTableName')

    # Perform some
DynamoDB operation (e.g.,
put item)
    response =
table.put_item(
        Item={
            'UserId': user_id,
            'InputText':
input_text
        }
    )

    # You can customize the
response based on the
DynamoDB operation result
    if
response['ResponseMetadat
a']['HTTPStatusCode'] ==
200:
        return {
            'dialogAction': {
                'type': 'Close',
                'fulfillmentState':
'Fulfilled',
                'message': {
                    'contentType':
'PlainText',
                    'content': 'Your
DynamoDB operation was
successful.'
                }
            }
        }
    else:
        return {
            'dialogAction': {
                'type': 'Close',
                'fulfillmentState':
'Failed',
                'message': {
                    'contentType':
'PlainText',
```

'content': 'There was an issue with the DynamoDB operation.'
            }
        }
    }

# CHAPTER - 6

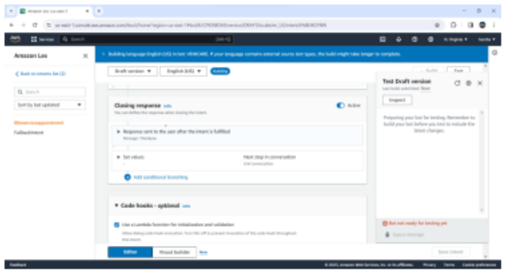
# TESTING

## 1.1 TEST THE BOT

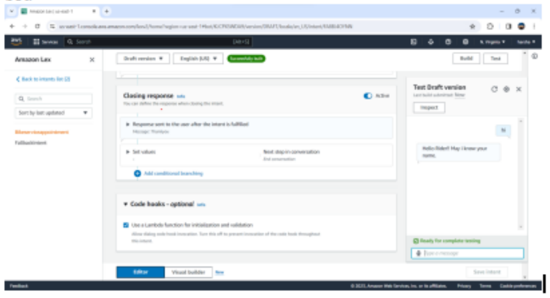
To build the VEHICARE bot, choose **Build**.

Amazon Lex builds a machine learning model for the bot. When you test the bot, the console uses the runtime API to send the user input back to Amazon Lex. Amazon Lex then uses the machine learning model to interpret the user input.

It can take some time to complete the build.

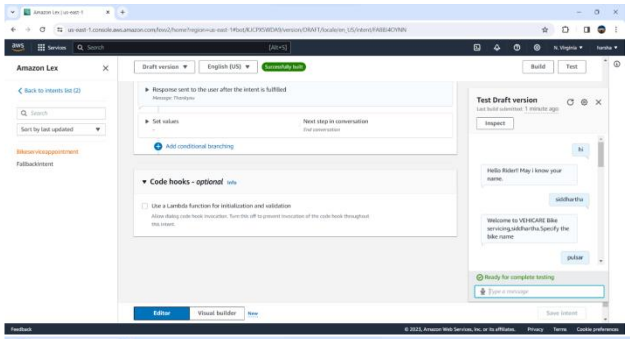


To test the bot, in the **Test Bot** window, start communicating with your Amazon Lex bot.



hi

Hello Rider!! May i know your name.



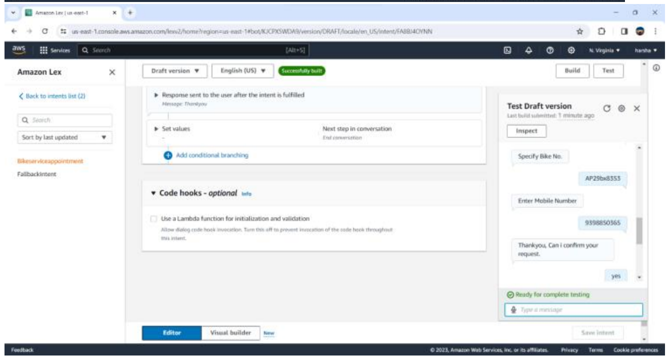
hi

Hello Rider!! May i know your name.

siddhartha

Welcome to VEHICARE Bike servicing,siddhartha.Specify the bike name

pulsar



Specify Bike No.

AP29bx8353

Enter Mobile Number

9398850365

Thankyou, Can i confirm your request.

yes

## 6.2 TESTING

**The various levels of testing are**

6.2.1 White Box Testing

6.2.2 Black Box Testing

### 6.2.1 White Box Testing

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

### 6.2.2 Black Box Testing

Black-box testing is a method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white-box testing). This method of test can be applied to virtually every level of software testing: unit, integration,

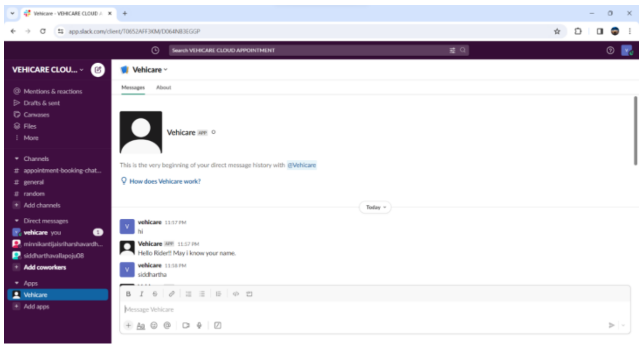
system and acceptance. It typically comprises most if not all higher-level testing, but can also dominate unit testing as well

Test procedures

# CHAPTER - 7

# RESULTS

## 7.1 SCREENSHOTS



Vehicare APP

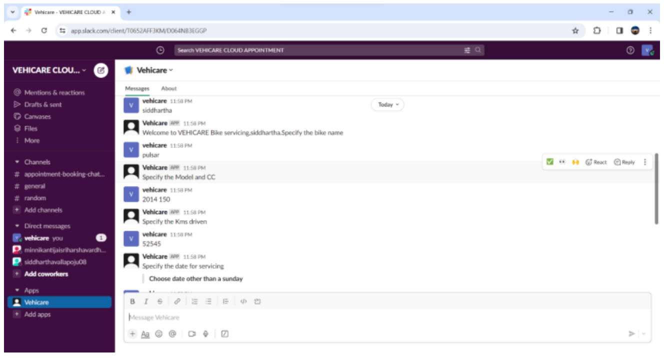
This is the very beginning of your direct message history with @Vehicare

How does Vehicare work?

vehicare 11:57 PM
hi

Vehicare APP 11:57 PM
Hello Rider!! May i know your name.

vehicare 11:58 PM
siddhartha



vehicare 11:58 PM
siddhartha

Vehicare APP 11:58 PM
Welcome to VEHICARE Bike servicing,siddhartha.Specify the bike name

vehicare 11:58 PM
pulsar

Vehicare APP 11:58 PM
Specify the Model and CC

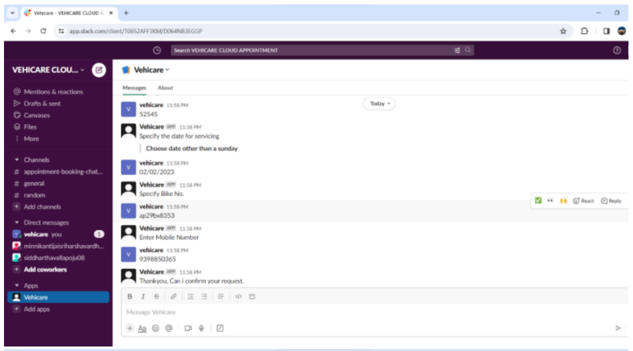
vehicare 11:58 PM
2014 150

Vehicare APP 11:58 PM
Specify the Kms driven

vehicare 11:58 PM
52545

Vehicare APP 11:58 PM
Specify the date for servicing

Choose date other than a sunday



vehicare 11:58 PM
52545

Vehicare APP 11:58 PM
Specify the date for servicing

Choose date other than a sunday

vehicare 11:58 PM
02/02/2023

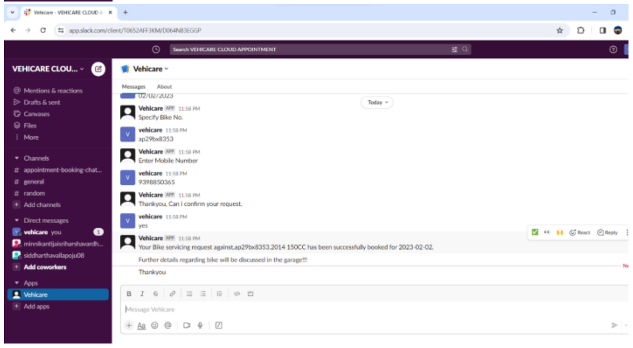
Vehicare APP 11:58 PM
Specify Bike No.

vehicare 11:58 PM
ap29bx8353

Vehicare APP 11:58 PM
Enter Mobile Number

vehicare 11:58 PM
9398850365

Vehicare APP 11:58 PM
Thankyou, Can i confirm your request.



Vehicare APP 11:58 PM
Specify Bike No.

vehicare 11:58 PM
ap29bx8353

Vehicare APP 11:58 PM
Enter Mobile Number

vehicare 11:58 PM
9398850365

Vehicare APP 11:58 PM
Thankyou, Can i confirm your request.

vehicare 11:58 PM
yes

Vehicare APP 11:58 PM
Your Bike servicing request against.ap29bx8353.2014 150CC has been successfully booked for 2023-02-02.
Further details regarding bike will be discussed in the garage!!!
Thankyou

# CHAPTER - 8

# CONCLUSION

## 8.1 CONCLUSION

The development and implementation of the Vehicle Service Appointment Booking System have significantly transformed the way customers engage with our service center. By providing a streamlined and user-friendly platform for scheduling vehicle maintenance, the system has enhanced customer satisfaction and operational efficiency.

**Key Achievements:**

- Improved User Experience: The system offers a convenient and intuitive interface, allowing customers to effortlessly book appointments, select services, and receive timely reminders.

# CHAPTER - 9

# FUTURE ENHANCEMENTS

## 9.1 FUTURE ENHANCEMENTS

- AI Integration: While you mentioned your current system isn't AI-powered, integrating AI could offer features like predictive maintenance scheduling based on vehicle usage patterns or suggesting optimal service packages based on vehicle history.
- Mobile App Development: Creating a dedicated mobile app for the service center could enhance user experience, allowing convenient booking, real-time notifications, and service tracking.

## REFERENCES

1. AWS Documentations

2.Amanpreet Kaur sandhu,"Big data with cloud computing: Discussions and challenges"-2022

3.Ishu Gupta;Ashutosh Kumar Singh;Chung-Nan Lee;Rajkumar Buyya,

"Secure Data Storage and Sharing Techniques for Data Protection in Cloud Environments: A Systematic Review, Analysis, and Future Directions
",2022
4.Garima Sinha;Rekha Chapagain;Apsara Budhathoki;Kunal Sarkar;Anjali Kumari Mandal;Owk Manorishik."Infrastructure as a Code Chatbot using Natural Language Processing",2023
5.Rrezarta Krasniqi;Hyunsook Do,"Generalizability of NLP-based Models for Modern Software Development Cross-Domain Environments
",2023
6.Ranci Ren;Sara Pérez-soler;John W. Castro;Oscar Dieste;Silvia T. Acuña,"Using the SOCIO Chatbot for UML Modeling: A Second Family of Experiments on Usability in Academic Settings",2022
7.Giovanni Almeida Santos;Guilherme Guy de Andrade;Geovana Ramos Sousa Silva;Francisco Carlos Molina Duarte;João Paulo Javidi Da Costa;Rafael Timóteo de Sousa,"A Conversation-Driven Approach for Chatbot Management",2022