COSC6376 - Final Report
Virtual Healthcare Assistant: A Chatbot Integrated with AWS for Medical Information Retrieval

Somaia Alhazmi, Sheema Anush , Sai Divya Sree
University of Houston

December 12, 2023

Contents

Chapter 1

Abstract

The Virtual Healthcare Assistant is an innovative chatbot developed to enhance patient experience by facilitating consistent interactions with healthcare services. This intelligent chatbot is specifically designed to assist users in scheduling medical appointments, retrieving pertinent health information, delivering timely medication reminders, and offering real-time responses to fundamental health-related inquiries. Leveraging the power of Amazon Web Services (AWS), the chatbot is integrated with various AWS services, ensuring efficient data processing, scalability to accommodate varying workloads, and reliable storage of sensitive health information. This project's goal is to establish a robust, scalable, and efficient chatbot system that seamlessly integrates into healthcare operations, ultimately contributing to an improved and streamlined healthcare communication ecosystem.

Chapter 2

Introduction

In the evolving landscape of healthcare, the integration of advanced technologies plays a substantial role in enhancing patient care and accessibility. This project, titled "Virtual Healthcare Assistant: A Chatbot Integrated with AWS for Medical Information Retrieval," responds to the growing need for innovative solutions that bridge the gap between patients and healthcare services. It is designed to be a adaptable and user-centric tool, the Virtual Healthcare Assistant employs the capabilities of Amazon Web Services (AWS) to ensure efficiency, scalability, and secure data management. The primary focus of this project is to create a robust and scalable chatbot system that seamlessly integrates with healthcare operations. By addressing fundamental challenges such as medical appointment scheduling, health information retrieval, and feedback mechanisms. The virtual healthcare assistant aims to redefine patient engagement and extend healthcare services to remote or inaccessible locations. This introduction provides a view into the overarching goals and objectives of the project, emphasizing the significance of leveraging AWS technologies for the development of an intelligent and responsive healthcare Chatbot. As we delve deeper into the subsequent sections. The project discusses the methodologies, and experiments by two sides from patients' and doctors' perspectives, and technological choices will be explored, highlighting its potential to revolutionize the way patients interact with healthcare services.

Chapter 3

Methodologies

3.1 Amazon Lex

These technologies effectively served as a perfect way, easing the healthcare system's load by monitoring and providing consultations to the public[5] Amazon Lex, a key component of the Virtual Healthcare Assistant project, serves as the natural language processing engine for patient interactions. Leveraging Lex, the chatbot can comprehend and interpret user inputs in a conversational manner. Through the definition of intents, slot types, and speech, Lex facilitates communication that allow patients to input their details in a natural and intuitive way. The integration of Lex enhances user experience, providing a responsive and interactive interface for gathering diverse information from patients. The inputs is collected to assess the viability, significance, and acceptance of technology for a specific use case. There is a remarkable changes in chatbot designs, transitioning from text-based interactions to voice-based interactions[4].

3.2 Amazon DynamoDB

The project employs Amazon DynamoDB as the NoSQL database for secure and efficient storage of patient data. DynamoDB offers a scalable, high-performance solution, ensuring the system can handle varying workloads while maintaining quick and reliable data retrieval. Patient details, including personal information, medical history, and appointment records, are securely stored in DynamoDB tables. This NoSQL database solution not only ensures the integrity and confidentiality of patient information, but also enables rapid data access, crucial for providing timely and accurate healthcare services[2].

3.3 Amazon Lambda

Amazon Lambda, a serverless computing service, plays a vital role in executing back-end functions for the Virtual Healthcare Assistant. By leveraging Lambda, the project benefits from a scalable and cost-effective architecture. Back-end operations such as querying databases, processing user input, and interacting with other AWS services are handled by Lambda functions. This serverless approach eliminates the need for traditional server maintenance, allowing the system to dynamically scale based on demand. The use of Lambda enhances the overall efficiency of the system, ensuring that back-end processes are executed swiftly and reliably. AWS Lambda function enables the solution to execute back-end logic code without the need to provision any servers[3].

3.4 Amazon S3

Amazon S3, or Simple Storage Service, is a highly scalable and secure object storage solution offered by Amazon Web Services (AWS). It provides developers with a durable and reliable platform to store and retrieve any amount of data at any time. S3 is designed for ease of use, offers a pay-as-you-go pricing model, and supports various storage classes, enabling users to optimize costs based on their specific data access patterns. With features such as data encryption, versioning, and robust access controls, Amazon S3 is a fundamental component for building scalable and resilient cloud-based applications.

Chapter 4

Integration of AWS Services

The integration Amazon Lex, DynamoDB, and Lambda creates a comprehensive system. Lex starts the natural language understanding process, DynamoDB provides a secure repository for storing patient data, and Lambda executes the necessary back-end operations. The integration ensures that the Virtual Healthcare Assistant operates cohesively, offering a seamless experience for both patients and doctors. It enables the chatbot to efficiently process user inputs, retrieve and store data securely, and execute necessary functions without the need for manual intervention.

Chapter 5

Scalability and Security

The scalability of the system is a crucial aspect addressed by the combined use of these AWS services. DynamoDB's scalability ensures that the database can handle the growth of patient records, while Lambda's serverless architecture allows the system to scale dynamically in response to varying workloads. Security measures, such as proper access control and encryption, are implemented to protect patient data throughout the entire process. The integrated AWS services not only contribute to the project's efficiency but also reinforce its commitment to maintaining the confidentiality and integrity of important healthcare information. In summary, the utilization of Amazon Lex, DynamoDB, and Lambda in the Virtual Healthcare Assistant project showcases the power of AWS services in creating a robust, scalable, and secure healthcare chatbot system. The integration of these technologies enhances user experience, ensures efficient data processing, and establishes a foundation for future advancements in healthcare technology.

Chapter 6

System Architecture

The system architecture demonstrates the flow of data between the patient and doctor interfaces, AWS Lex for conversation processing, AWS Lambda for back-end operations, and DynamoDB for secure data storage as shown in figure 6.1

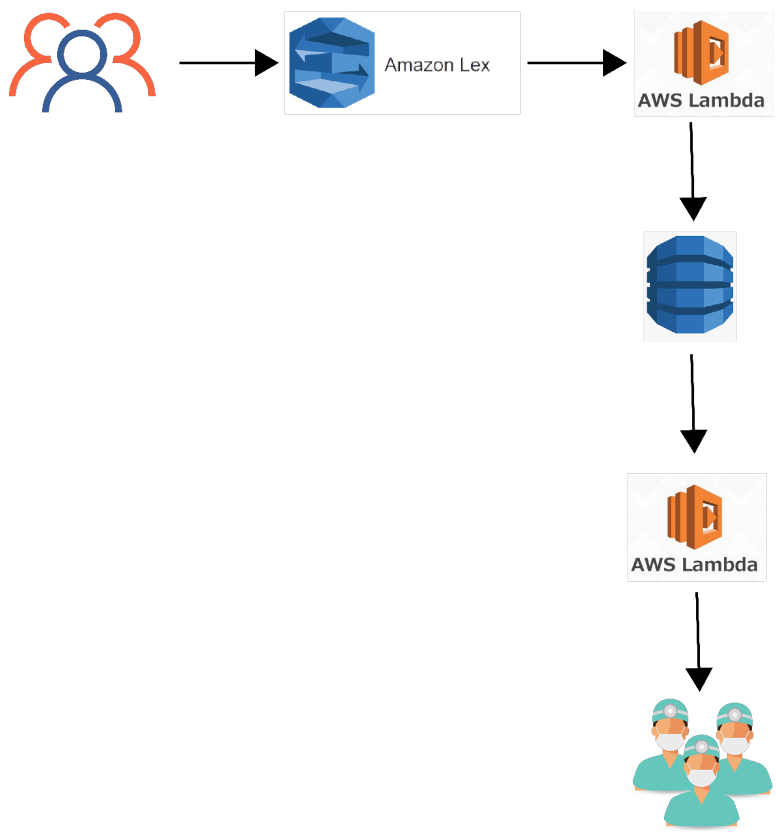


Figure 6.1: The System Architecture

Chapter 7

Experiments

7.0.1 Patient Interface Testing

In the testing phase for the patient interface, usability and the ability of AWS Lex to process diverse patient inputs were thoroughly examined. The aim was to ensure a consistent and intuitive experience for users while efficiently capturing essential health-related information. The testing process as shown in figure 7.5

Ease of Use Evaluation

First, we implemented a user-friendly interface allowing patients to input personal details such as name, contact information, and address effortlessly. secondly, employed a straightforward form layout for capturing essential information, optimizing the user experience.

Natural Language Processing Validation

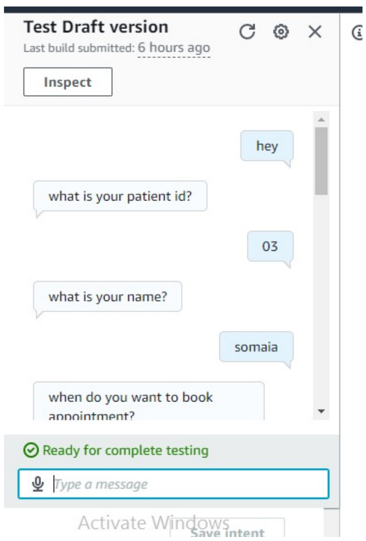
we tested AWS Lex's natural language understanding capabilities by allowing patients to input appointment details using conversational language. then, validated Lex's ability to interpret and extract relevant information, including appointment date, time, and specific health parameters like blood pressure and blood glucose levels.

Appointment Information Gathering

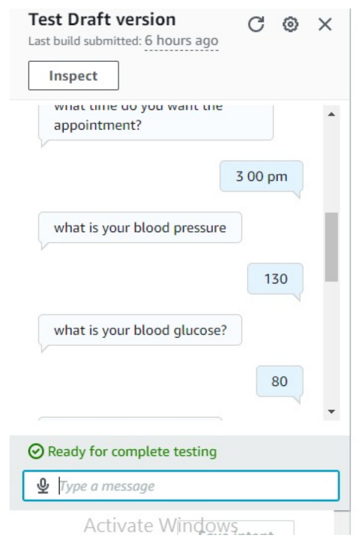
Designed conversation flows in Lex to prompt users for appointment details, extracting information in a conversational manner. Utilized Lex slots to capture specific data points, such as appointment date and time, ensuring accuracy in scheduling.

Health Parameters Input

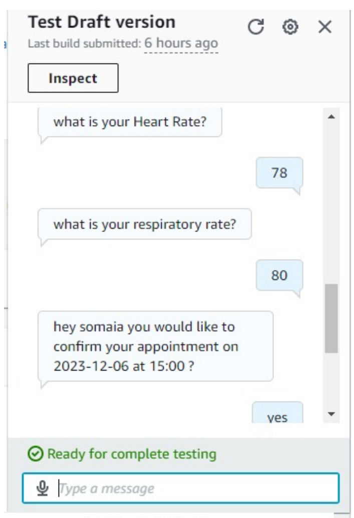
Integrated Lex for capturing health parameters like blood pressure and blood glucose levels. Then, defining a specific intents and slots in Lex to recognize



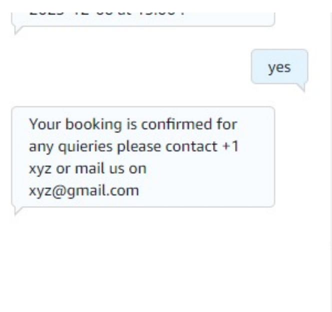
(a)



(b)

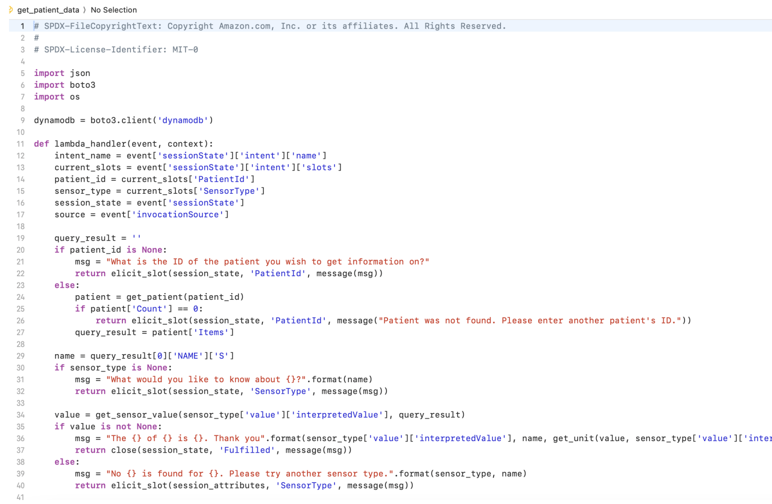


(c)



(d)

Figure 7.1: All Four Figures for Patient Interface Testing



get_patient_data 〉 No Selection

```python
# SPDX-FileCopyrightText: Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
#
# SPDX-License-Identifier: MIT-0

import json
import boto3
import os

dynamodb = boto3.client('dynamodb')

def lambda_handler(event, context):
    intent_name = event['sessionState']['intent']['name']
    current_slots = event['sessionState']['intent']['slots']
    patient_id = current_slots['PatientId']
    sensor_type = current_slots['SensorType']
    session_state = event['sessionState']
    source = event['invocationSource']

    query_result = ''
    if patient_id is None:
        msg = "What is the ID of the patient you wish to get information on?"
        return elicit_slot(session_state, 'PatientId', message(msg))
    else:
        patient = get_patient(patient_id)
        if patient['Count'] == 0:
            return elicit_slot(session_state, 'PatientId', message("Patient was not found. Please enter another patient's ID."))
        query_result = patient['Items']

    name = query_result[0]['NAME']['S']
    if sensor_type is None:
        msg = "What would you like to know about {}?".format(name)
        return elicit_slot(session_state, 'SensorType', message(msg))

    value = get_sensor_value(sensor_type['value']['interpretedValue'], query_result)
    if value is not None:
        msg = "The {} of {} is {}. Thank you".format(sensor_type['value']['interpretedValue'], name, get_unit(value, sensor_type['value']['interp
        return close(session_state, 'Fulfilled', message(msg))
    else:
        msg = "No {} is found for {}. Please try another sensor type.".format(sensor_type, name)
        return elicit_slot(session_attributes, 'SensorType', message(msg))
```

Figure 7.2: Get Patient Data

and extract numeric values, allowing for the accurate recording of vital health metrics.

User Feedback Mechanism

Incorporated a feedback mechanism within the interface to gather user opinions on the ease of data input. Adjusted the interface based on user feedback to enhance overall usability.

Validation of Data Accuracy

Conducted tests to ensure the accuracy of data input, particularly for critical health metrics. Leveraged Lex's validation capabilities to confirm that users provided information in a format that aligns with medical standards.

Error Handling Assessment

We tested Lex's error handling mechanisms to manage situations where users input incorrect or incomplete information. Also,we implemented informative prompts to guide users in providing the necessary details. By conducting extensive usability testing and validating AWS Lex's natural language processing capabilities, the patient interface was refined to guarantee an intuitive and effective means of capturing diverse patient data, including personal details, appointment information, and vital health parameters. This iterative testing process ensures that the patient interface not only meets but exceeds user expectations for ease of use and data accuracy.

7.0.2 Doctor Interface Results

7.0.3 Appointment Details

The time and date of scheduled appointments are prominent displayed, allowing doctors to manage their schedules efficiently and prioritize patient interactions.

7.0.4 Medical History

In this phase, Accessing to a patient's medical history, including past diagnoses, treatments, and medications, provides critical context for informed decision-making during consultations as shown in figure 7.3

7.0.5 Vital Signs Monitoring

Real-time monitoring of vital signs such as blood pressure and blood glucose levels is integrated into the interface, enabling doctors to assess and respond promptly to any concerning health trends.

7.0.6 Diagnostic Results

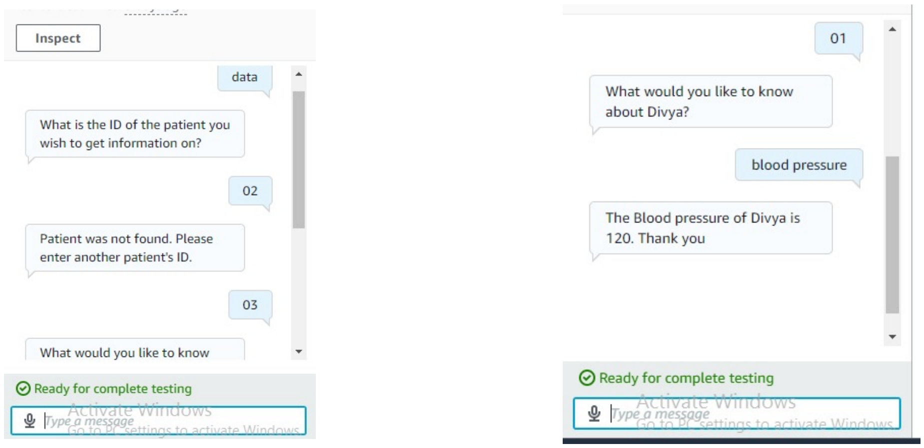
Diagnostic test results, including laboratory reports and imaging studies, are presented in a structured format for quick and easy interpretation by healthcare professionals.

7.0.7 Medication Records

A comprehensive list of prescribed medications, dosage information, and adherence records is available to support medication management and patient compliance. In summary, the doctor interface is designed to authorize healthcare professionals with efficient and secure access to a patient's comprehensive health profile. This is can be achieved by combining intuitive data retrieval functionalities with robust security measures, the system ensures that doctors can make well-informed decisions while maintaining the most confidentiality and integrity of patient data. It has been added to the DynamoDB the data from the patients. The doctors' information is shown in figure 7.4

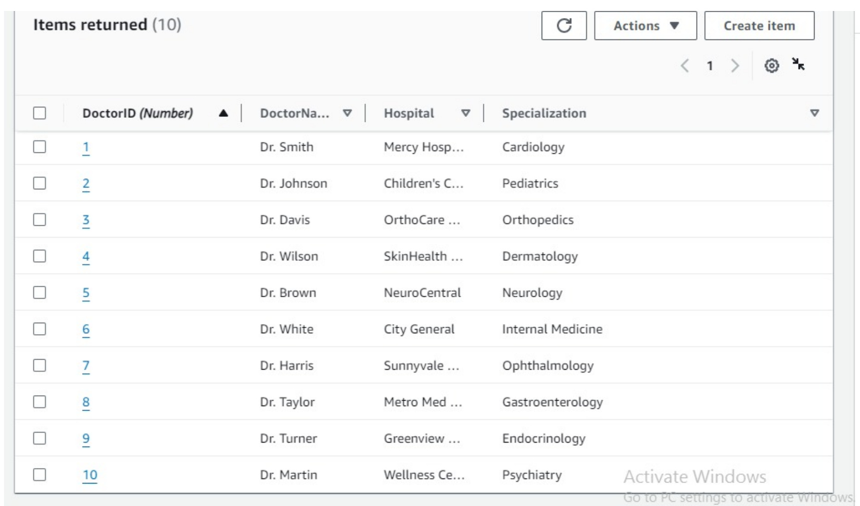
7.1 Impact on Patient Experience

The patient interface, empowered by AWS Lex, stands out as a user-friendly gateway for individuals to input their personal and medical details. This feature not only efficient the process of scheduling, rescheduling, and canceling medical appointments but also empowers patients with continuous access to relevant health information. The utilization of natural language processing through AWS Lex ensures a seamless and intuitive interaction, reducing barriers for users of varying technical proficiencies. The incorporation of Amazon DynamoDB as the



(a) (b)

Figure 7.3: Doctors Interface



Items returned (10)

DoctorID (Number)	DoctorNa...	Hospital	Specialization
1	Dr. Smith	Mercy Hosp...	Cardiology
2	Dr. Johnson	Children's C...	Pediatrics
3	Dr. Davis	OrthoCare ...	Orthopedics
4	Dr. Wilson	SkinHealth ...	Dermatology
5	Dr. Brown	NeuroCentral	Neurology
6	Dr. White	City General	Internal Medicine
7	Dr. Harris	Sunnyvale ...	Ophthalmology
8	Dr. Taylor	Metro Med ...	Gastroenterology
9	Dr. Turner	Greenview ...	Endocrinology
10	Dr. Martin	Wellness Ce...	Psychiatry

Figure 7.4: Doctors Data

underlying database solution brings forth a paradigm shift in data management within healthcare. The secure, scalable, and NoSQL nature of DynamoDB ensures not only the confidentiality of patient information but also provides a foundation for accommodating the growing volume of healthcare data. This scalability becomes particularly crucial in the context of an expanding patient base and the increasing complexity of medical records.

7.2 Empowering Healthcare Providers

The doctor interface, facilitated by AWS Lambda for serverless computing, empowers healthcare providers with a comprehensive view of patient records. This interface is designed to not only provide access to historical medical data but also to support real-time decision-making. The deployment of RESTful APIs through Amazon API Gateway ensures secure and streamlined interactions, fostering efficient communication channels between healthcare professionals and their patients. Moreover, the inclusion of Amazon S3 for storing Electronic Health Records (EHR) and other unstructured data adds a layer of versatility to the project. This feature not only enhances data storage capabilities but also enables the secure archiving of diverse healthcare-related files, such as images and documents. The result is a unified platform that consolidates patient information, supporting holistic healthcare services.

7.3 Performance

Amazon CloudWatch is AWS's comprehensive monitoring and performance testing service, offering real-time insights into resource utilization, application performance, and operational health. With customizable dashboards, CloudWatch provides visibility into key metrics, enabling proactive resource optimization. Alarms and notifications trigger timely responses to performance deviations, ensuring rapid issue resolution. CloudWatch Logs centralize log data, facilitating analysis and troubleshooting, while CloudWatch Metrics cover a broad spectrum, including compute, storage, and networking metrics, contributing to a holistic performance monitoring solution in the AWS cloud. As shown in the figures 7.5

7.4 Achievements and Future Prospects

The successful implementation of this project stands as a testament to the potential of cloud-based solutions in revolutionizing healthcare delivery. The project has not only met its initial objectives of modernizing medical appointments and enhancing health information accessibility but has also opened avenues for continuous improvement through user feedback. As we look to the future, the Virtual Healthcare Assistant project is assured for further enhancements and collaborations. Integration with additional AWS services, exploration of machine learning algorithms for predictive health analysis, and collaboration with healthcare institutions are among the avenues for future development. These efforts align with our commitment to delivering a comprehensive and forward-thinking solution that adapts to the evolving view of healthcare technology. In conclusion, the Virtual Healthcare Assistant project, with its integration of AWS technologies, represents a significant stride towards a more connected, efficient, and patient-centric healthcare system. As we navigate the evolving healthcare

(a) (b) (c) (d) (e) (f)

Figure 7.5: The Performance Figures

landscape, this project stands as a guide of innovation, promising benefits for patients, healthcare providers, and the broader healthcare community.

7.5 Conclusion

The completion of the Virtual Healthcare Assistant project marks a significant milestone in the integration of innovative technologies to address critical challenges in the healthcare sector. By leveraging AWS services such as Lex, DynamoDB, Lambda, and S3, we have successfully developed a comprehensive and adaptable system that not only enhances patient-doctor interactions but also lays the essential for a more connected and efficient healthcare ecosystem. Chatbot is available on the web at[1]

References

[1] Amazon. *Amazon AWS*. 2023. URL: https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1 (visited on 12/06/2023).

[2] Shreya Bhutada et al. "Ru-Urb IoT-AI powered Healthcare Kit". In: *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE. 2021, pp. 417–422.

[3] Stanley Mierzwa et al. "On the potential, feasibility, and effectiveness of chat bots in public health research going forward". In: *Online journal of public health informatics* 11.2 (2019).

[4] Sanket Sanjay Sadavarte and Eliane Bodanese. "Pregnancy companion chatbot using alexa and amazon web services". In: *2019 IEEE Pune Section International Conference (PuneCon)*. IEEE. 2019, pp. 1–5.

[5] Daniel Shu Wei Ting et al. "Digital technology and COVID-19". In: *Nature medicine* 26.4 (2020), pp. 459–461.