“ConCraft”

A

Project Report

Submitted to the

SAGE University, Bhopal, M.P.

in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Computer Application

Specialization in

Cloud Computing

Semester-VI

By

Abhinav S
[21BCA3CCM10001]
Kopal Vishwakarma
[21BCA3CCM10005]

Under the Guidance of
Mr. Lokesh Sahu
Assistant Professor

SCHOOL OF COMPUTER APPLICATION

SANJEEV AGRAWAL GLOBAL EDUCATIONAL UNIVERSITY, BHOPAL

SPRING 2023-24

SANJEEV AGRAWAL GLOBAL EDUCATIONAL UNIVERSITY, BHOPAL

SCHOOL OF COMPUTER APPLICATION

CERTIFICATE

This is to certify that the work embodies in this project entitled “**ConCraft”** being submitted by **Abhinav S [21BCA3CCM10001], Kopal Vishwakarma [21BCA3CCM10005]** in partial fulfillment of the requirement for the award of the degree of **BCA-Cloud Computing** to School of Computer Application, Sanjeev Agrawal Global Educational University, Bhopal (M.P) during the academic year **2023-24** is a record of bonafide piece of work, undertaken by him under the supervision of the undersigned.

Mr. Lokesh Sahu	**Dr. Pinaki Ghosh**
Assistant Professor, SSOCA	**HOS, SSOCA**
(Project Guide)	

SANJEEV AGRAWAL GLOBAL EDUCATIONAL UNIVERSITY, BHOPAL

SCHOOL OF COMPUTER APPLICATION

CERTIFICATE OF APPROVAL

The Project entitled **“ConCraft”** being submitted by **Abhinav S [21BCA3CCM10001], Kopal Vishwakarma [21BCA3CCM10005]** has been examined by us and is hereby approved for the award of the degree of **BCA-Cloud Computing,** for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn there in, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner) (External Examiner)

SANJEEV AGRAWAL GLOBAL EDUCATIONAL UNIVERSITY, BHOPAL

SCHOOL OF COMPUTER APPLICATION

DECLARATION

We hereby declare that the work, which is being presented in this project entitled **"ConCraft"** for fulfillment of the requirements for the award of the degree of **BCA-Cloud Computing,** Semester-VI submitted in the School of Advanced Computing, Sanjeev Agrawal Global Educational University, Bhopal, M.P. is an authentic record of our own work carried under the guidance of **"Mr. Lokesh Sahu"**. We have not submitted the matter embodied in this report for the award of any other degree.

We also declare that "A check for Plagiarism has been carried out on this dissertation and is found within the acceptable limit and report of which is enclosed herewith."

Abhinav S
[21BCA3CCM10001]
Kopal Vishwakarma
[21BCA3CCM10005]

SANJEEV AGRAWAL GLOBAL EDUCATIONAL UNIVERSITY, BHOPAL

SCHOOL OF COMPUTER APPLICATION

ACKNOWLEDGEMENT

It is our proud privilege to present a project on **“ConCraft”**. We take this opportunity to express deep sense of gratitude and would like to give thanks to our guide, “**Mr. Lokesh Sahu”, Assistant Professor,** School of Advanced Computing, Sanjeev Agrawal Global Educational University, Bhopal, M.P. for his valuable guidance, inspiration and encouragement that has led to successful completion of this work.

We would like to express our heartfelt thanks towards **Dr. Pinaki Ghosh, HOS, School of Computer Application & Advanced Computing**. We could not have accomplished, what we actually have, without their guidance. We would like to express our heartfelt thanks to

Dr. Gourav Shrivastava, HOD, School of Computer Application, for his valuable suggestions throughout the project work.

We would like to express my heartfelt thanks and sense of gratitude to “**Mr. Lokesh Sahu”** for being a constant source of inspiration. We are also thankful to all faculty members and staff of School of Advanced Computing for their suggestions and support.

We would like to deeply thank our family and friends for all the support and encouragement they have rendered time to time.

Last but not the least, We dedicate my work to almighty God without whose wish and helping hands this work would not have taken the shape it has now and also to our family members whose support and encouragement had led us to complete this task.

Abhinav S
[21BCA3CCM10001]
Kopal Vishwakarma
[21BCA3CCM10005]

ABSTRACT

In today's dynamic work environment, missing important meetings can significantly hinder team collaboration and create information gaps, potentially leading to missed opportunities and misunderstandings. To address this critical issue, we introduce ConCraft, a groundbreaking solution leveraging Amazon Lex's intelligent chatbot technology, integrated with AWS Lex and Lambda. ConCraft serves as a vital link between team members who attend meetings and those who are unable to participate, ensuring continuous and seamless access to the vital information discussed and captured during sessions.

ConCraft's primary feature is a rich and user-friendly chatbot interface that provides fast and easy access to meeting details, selections, and updates maintained in a detailed datasheet curated by the team leader. Through simple commands, users can quickly retrieve meeting agendas, project updates, and assigned action items, effectively having a personal assistant at their disposal. This ensures that no team member is left out of the loop, promoting a more inclusive and collaborative work environment.

Beyond the convenience of accessing meeting information on-demand, ConCraft significantly supports diversity within teams. By enabling all team members to contribute productively, regardless of their availability, ConCraft fosters an inclusive culture where every member's input is valued. This inclusivity minimizes the fear of missing out (FOMO) that often accompanies missing meetings, encouraging a culture of collaboration, transparency, and accountability. With ConCraft, team members can stay connected, informed, and engaged, irrespective of their location or schedule, bridging the gap created by time zones and different work hours.

In Summary,ConCraft is designed to enhance overall team productivity. By ensuring that all team members have access to the same information, ConCraft reduces the need for repetitive briefings and follow-ups, allowing teams to focus more on their core tasks. This streamlined communication process leads to more efficient decision-making and project execution.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

In today’s fast-paced work environment, missing a meeting can feel like missing a pivotal moment. The exchange of critical information, the formulation of key decisions, and the assignment of essential tasks often happen in these collaborative settings. However, the reality of modern work means that attending every meeting isn't always feasible. This is where ConCraft steps in, an innovative solution designed to ensure that no team member is left behind.

ConCraft is an intelligent chatbot powered by Amazon Lex and integrated with AWS Lambda, designed to revolutionize how teams access and share information. This project transcends being merely a tool; it is a comprehensive solution aimed at enhancing team collaboration and ensuring that all team members, regardless of their availability, remain informed and engaged.

At the heart of ConCraft lies its ability to interpret and deliver meeting information efficiently. During meetings, the team lead meticulously documents all discussions, decisions, and updates in a datasheet. This datasheet is more than just a record; it is a repository of valuable insights, capturing everything from action items and deadlines to strategic decisions and project milestones.

However, ConCraft elevates this process. With a few simple commands, team members can quickly access any piece of information stored in these datasheets. Whether you need to review the meeting agenda, catch up on the latest project developments, or clarify specific action items assigned to you, ConCraft is your go-to assistant. This intelligent chatbot operates seamlessly, providing instant access to critical information, thereby ensuring that you are always in the loop.

Key Features of ConCraft

- Intuitive information retrieval: ConCraft is designed to understand natural language commands, making it easy for users to request information. Simply ask ConCraft about the latest project updates, upcoming deadlines, or any specific decisions made during the meeting, and it will provide the relevant details instantly.

- Seamless integration with AWS Lambda: The integration with AWS Lambda allows ConCraft to process requests efficiently, ensuring quick and accurate responses. This serverless architecture also ensures scalability, so ConCraft can handle multiple requests simultaneously without compromising on performance.

- Comprehensive data access: By pulling information directly from the datasheet created by the team lead, ConCraft ensures that all critical details are captured and accessible. This includes action items, key insights, and even nuanced discussions, ensuring that nothing is missed.

- Enhanced collaboration: ConCraft promotes a culture of transparency and accountability within the team. By making all meeting information readily available, it ensures that every team member has the same level of understanding and can contribute effectively, regardless of their attendance at the meeting.

Benefits of using ConCraft

- No more missed meetings: With ConCraft, missing a meeting doesn't mean missing out on crucial information. This tool ensures that you can stay updated and informed, eliminating the fear of missing out (FOMO).

- Increased productivity: By providing instant access to meeting information, ConCraft saves time and reduces the need for follow-up meetings or lengthy email threads. This efficiency translates to increased productivity for the entire team.

- Empowered team members: ConCraft empowers every team member by giving them access to the same information. This fosters a sense of equality and encourages proactive participation, as everyone is equipped with the knowledge they need to contribute their best.

- Improved decision-making: With all relevant information at their fingertips, team members can make informed decisions quickly. This agility is crucial in a fast-paced work environment where timely decision-making can be a significant competitive advantage.

Use cases of ConCraft

- Project management: Project managers can use ConCraft to keep track of progress, update team members on the latest developments, and ensure that everyone is aware of their responsibilities and deadlines. This streamlined communication helps in maintaining project momentum and achieving targets on time.

- Team collaboration: For teams working remotely or across different time zones, ConCraft is a game-changer. It ensures that all team members have access to the same information, fostering a cohesive and collaborative work environment.

- Employee onboarding: New hires can use ConCraft to catch up on past meetings, understand ongoing projects, and get up to speed with the team's workflow. This accelerates the onboarding process and helps new employees become productive more quickly.

- Meeting recaps: Even for those who attend meetings, ConCraft can serve as a quick reference tool to recap important points and action items. This ensures that no detail is overlooked and that follow-up actions are clear and well-documented.

Technical Overview

ConCraft leverages the power of Amazon Lex, a service for building conversational interfaces using voice and text. Amazon Lex provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text. These capabilities allow ConCraft to engage in lifelike conversations with users.

The integration with AWS Lambda further enhances ConCraft's capabilities. AWS Lambda allows ConCraft to execute backend code in response to triggers such as requests from Amazon Lex. This serverless architecture ensures that ConCraft is both scalable and cost-effective, capable of handling a large number of requests without the need for complex infrastructure management.

Services used to implement concraft chatbot

The integration of AWS services like Amazon Lex, AWS Lambda, Amazon S3, IAM, and CloudWatch into ChatOps platforms significantly enhances customer engagement by providing scalable, secure, and efficient solutions. These services enable the creation of intelligent, responsive, and reliable chatbots that can handle a wide range of customer interactions seamlessly. With advanced natural language understanding, secure data management, and real-time monitoring, businesses can ensure high-quality, engaging, and trustworthy interactions, driving customer satisfaction and loyalty.

1. **Amazon Lex:** Amazon Lex is a service for building conversational interfaces using voice and text, which is fundamental in the ChatOps industry for creating intelligent chatbots. These chatbots can handle a variety of tasks, from answering customer queries to automating routine processes.

- Natural Language Processing: Amazon Lex uses advanced NLP to understand and interpret user inputs. This capability ensures that chatbots can engage in meaningful and context-aware conversations with customers, leading to more satisfying interactions.
- Voice and text integration: By supporting both voice and text interactions, Lex offers a versatile communication channel, meeting diverse customer preferences and making it easier for users to engage with the system.
- Quick response time: Lex-powered chatbots can handle multiple queries simultaneously, providing instant responses and reducing wait times, which is critical for maintaining high levels of customer satisfaction.

2. **AWS Lambda:** AWS Lambda is a serverless compute service that runs code in response to events. It is integral in processing the logic behind chatbot interactions without the need for managing servers.

- Scalability: Lambda can automatically scale to handle increasing loads, ensuring that customer queries are processed efficiently even during peak times, maintaining a smooth user experience.

- Cost efficiency: Since Lambda only charges for actual compute time, it helps in managing operational costs effectively, allowing businesses to allocate resources towards enhancing customer engagement strategies.
- Event driven interactions: Lambda can trigger functions based on specific events, enabling real-time processing of customer requests and dynamic content delivery, enhancing the overall interaction quality.

3. **Amazon S3:** Amazon S3 is used for storing and retrieving large amounts of data, including documents, images, and other content that chatbots may need to access or provide to customers.

- Reliable storage: With its high availability and durability, S3 ensures that customer data and interaction logs are stored securely and can be accessed whenever needed.
- Content management: S3 can store various media files that chatbots can share with customers, such as product images, user manuals, and tutorial videos, making interactions more informative and engaging.
- Seamless integration: Integrating S3 with other AWS services allows for the efficient retrieval and management of data, ensuring that customers receive up-to-date information promptly.

4. **AWS IAM:** AWS IAM manages access to AWS resources, ensuring secure and controlled access to sensitive data and services.

- Security and compliance: IAM enforces strict access controls, ensuring that customer data is protected against unauthorized access. This builds trust with customers who are increasingly concerned about data privacy and security.
- Granular permissions: IAM allows for fine-grained permissions, enabling businesses to provide precise access to different parts of the system based on roles, which ensures that only authorized personnel can make changes that impact customer interactions.
- Audit and monitoring: IAM's logging capabilities help track access and actions, facilitating audits and ensuring compliance with industry standards, further assuring customers of the integrity of their data.

5. **Amazon CloudWatch**: Amazon CloudWatch is a monitoring service that provides data and actionable insights to manage applications and services.

- Performance monitoring: CloudWatch monitors the performance of the ChatOps infrastructure, ensuring that services are running optimally. This helps in maintaining a high-quality customer interaction experience without downtime or slow responses.
- Real-Time alerts: By setting up alarms for specific metrics, CloudWatch can alert administrators to issues in real-time, allowing for quick resolution and minimizing impact on customer experience.
- Insightful analytics: CloudWatch logs provide detailed analytics about system usage and performance, helping businesses understand customer behavior and interaction patterns. This data can be used to optimize chatbot responses and improve overall engagement strategies

Key Difference between Traditional chatbot and ConCraft

Feature	**Traditional Chatbot**	**ConcCaft Chatbot**
Purpose	General-purpose interactions and customer support	Specialized in providing meeting summaries
Data Source	General knowledge base or predefined responses	Meeting datasheets
Information Provided	General information based on user queries	Specific details from meeting summaries
Data Storage	Various databases or pre-configured data sets	Amazon S3 storing meeting datasheets
Query Handling	Basic queries with limited contextual understanding	Comprehensive access to meeting summaries
NLP Capabilities	Varies widely, often limited to basic NLP	Advanced NLP using Amazon Lex
Backend Processing	Basic logic processing	Advanced processing with AWS Lambda
Access Control	Typically limited to simple authentication	Secure access with AWS IAM
Monitoring	Basic monitoring, if any	Advanced monitoring using Amazon CloudWatch
User Interaction	Text-based interaction	Text-based interaction focused on meeting summaries
Customization	Limited customization	Highly customizable to handle specific meeting-related queries
Response Specificity	General responses	Detailed, context-specific responses from meeting datasheets
Integration	General integration capabilities	Seamless integration with AWS services

CHAPTER 2

LITERATURE SURVEY

This literature survey explores various chatbot technologies and platforms, examining their features, capabilities, and applications. The focus is on four prominent chatbot solutions: LiveChat, Jira Help Desk, IBM Watsonx Assistant, and Heptik. Each of these platforms offers unique functionalities that cater to different user needs and business requirements.

LiveChat ChatBot

Overview: LiveChat is a robust customer service platform designed to enhance customer engagement and support through dynamic, real-time interactions. It consists of two main components: the customer-facing Chat Widget and the Agent App.

- Dynamic responses: LiveChat creates chatbots that generate dynamic responses, encouraging customers to interact and engage with the platform.
- Chat widget: The Chat Widget is embedded on websites, providing an accessible interface for customers to initiate conversations. It is intuitive and designed to attract user interaction.
- Agent app: This backend application enables agents to manage conversations, monitor chat sessions, and provide human assistance when necessary.
- Integration: LiveChat integrates seamlessly with various CRM and helpdesk systems, enhancing its utility across different business environments [1].

Applications: LiveChat is widely used in customer support, e-commerce, and sales. It helps businesses respond promptly to customer inquiries, reducing wait times and improving customer satisfaction.

Impact: The immediate connection and personalized interaction fostered by LiveChat lead to higher customer engagement and retention. It also alleviates the workload on human agents by handling routine queries and providing quick resolutions.

Jira Help Desk

Overview: The Chatbot for JIRA Service Management leverages Google's Dialogflow to facilitate natural language conversations between users and the helpdesk system. It aims to expedite issue resolution within the JIRA ecosystem.

- Natural Language Processing (NLP): Utilizing Dialogflow, the chatbot understands and processes user queries expressed in natural language, making interactions seamless and intuitive.
- Integration with JIRA: The chatbot integrates directly with JIRA Service Management, allowing users to log tickets, check statuses, and resolve issues without leaving the chat interface.
- Automated responses: It provides automated, context-aware responses to common inquiries, helping users find solutions quickly. [2]

Applications: The primary applications of the JIRA Help Desk chatbot are found in IT and project management environments. In these settings, the chatbot serves as a valuable assistant, helping users manage tasks, track project progress, and address technical issues. IT departments can use the chatbot to log and monitor support tickets, ensuring that issues are resolved promptly. Project managers can leverage the chatbot to keep track of project milestones, manage workloads, and communicate effectively with team members. By providing real-time updates and facilitating efficient task management, the chatbot supports the smooth execution of projects and the swift resolution of technical problems.

Impact: The impact of the JIRA Help Desk chatbot on organizations is substantial. By streamlining issue resolution processes and reducing the time required to address user problems, the chatbot enhances operational efficiency. Users benefit from immediate assistance and quick access to information, leading to higher satisfaction levels. The reduced dependency on human agents for routine inquiries allows support teams to allocate their resources more effectively, focusing on strategic initiatives and complex problem-solving tasks. This improvement in resource allocation and user support contributes to a more productive and responsive service management environment.

IBM Watsonx Assistant

Overview: IBM Watsonx Assistant is an advanced AI-powered chatbot platform that goes beyond basic chat functionalities to deliver comprehensive customer service solutions.

- Advanced NLP and machine learning: Watsonx Assistant employs sophisticated NLP and machine learning algorithms to understand complex queries and provide accurate responses.
- Backend integrations: The platform integrates with various backend systems and utilizes robotic process automation (RPA) to perform tasks and retrieve information, offering a seamless user experience.
- Contextual understanding: It can maintain context over multiple interactions, ensuring coherent and relevant conversations.
- Multi-channel support: Watsonx Assistant is accessible through web, mobile, and social media platforms, providing flexible engagement options for users. [3]

Applications: Watsonx Assistant is utilized across various industries, including finance, healthcare, and retail. In the finance sector, it helps customers manage their accounts, process transactions, and answer complex financial queries. In healthcare, the assistant can provide patients with information on medical conditions, schedule appointments, and assist with telehealth services. In retail, it supports customers by guiding them through purchasing processes, handling returns, and offering product recommendations. By providing 24/7 support, guiding customers through complex processes, and automating routine tasks, Watsonx Assistant significantly enhances customer service capabilities.

Impact: The impact of IBM Watsonx Assistant on businesses is profound. By delivering immediate and accurate responses, the assistant significantly boosts customer satisfaction. Its ability to perform actions dynamically and integrate with backend systems improves operational efficiency, reducing the workload on human agents and allowing them to focus on more complex tasks. This adaptability and integration capability make Watsonx Assistant a powerful tool for transforming customer service operations. Businesses can leverage this technology to streamline their processes, reduce response times, and provide a higher level of service,

Heptik

Overview: Heptik is a chat-based personal assistant platform that was launched as an app for Android and iOS in India. It aims to help users accomplish various tasks through chat interactions.

- Personal assistance: The Heptik app acts as a personal assistant, helping users with tasks such as booking tickets, making reservations, and providing information.
- Human expertise: By September 2014, Heptik had integrated 125 chat experts who assisted users with their queries, ensuring a blend of automated and human support.
- Mobile platform: Designed for mobile users, Heptik offers an intuitive interface for on-the-go assistance. [4]

Applications: Heptik is primarily targeted at individual users seeking assistance with daily tasks. It is used for a variety of purposes, from travel arrangements to shopping and general information.

Impact: By combining AI with human expertise, Heptik provides a reliable and versatile personal assistant experience.

Comparative Analysis:

Each of these chatbot platforms brings unique strengths to the table, catering to different aspects of user interaction and business requirements:

- LiveChat excels in customer engagement and real-time support, making it ideal for businesses focused on enhancing customer interaction and satisfaction.
- Jira Help Desk leverages powerful NLP capabilities to streamline issue resolution within the JIRA ecosystem, significantly improving operational efficiency in project and service management contexts.
- IBM Watsonx Assistant offers a highly advanced, adaptable solution that integrates seamlessly with backend systems, providing comprehensive support across multiple industries.
- Heptik combines AI with human expertise to offer personalized assistance.

2.1 Problem Identification

In the modern workplace, collaboration and effective communication are pivotal for the success of any project. However, several challenges arise when team members cannot attend meetings. ConCraft addresses these problems to streamline information access and improve team productivity. The following points detail the specific problems ConCraft aims to solve:

1. **Missed meetings**

- Issue: In today's fast-paced work environment, it is not always possible for team members to attend every meeting.
- Impact: This results in team members missing out on critical discussions, updates, and decisions, which can hinder their ability to contribute effectively to the project. The absence from meetings can lead to significant knowledge gaps, making it challenging for the absentee members to catch up and stay aligned with the ongoing project developments. This disconnect can create a ripple effect, where the missed information could potentially lead to misinformed decisions and actions that are out of sync with the rest of the team.

2. **Crucial information gap**

- Issue: Missing a meeting can result in missing out on important discussions, decisions, and updates that are crucial for project progress and individual contributions.
- Impact: This gap can lead to uninformed decisions, repeated discussions, and overall inefficiency within the team. When team members are not up-to-date with the latest information, they might make decisions based on outdated or incomplete data. This not only reduces the efficiency of the team but also increases the likelihood of errors and the need for rework, thereby slowing down the overall progress of the project.

3. **Inefficient catch-up mechanism**

- Issue: The current method of catching up on missed meetings involves manually reviewing datasheets or notes, which is time-consuming and inefficient.
- Impact: Team members spend valuable time sifting through extensive documentation instead of focusing on their core tasks. This manual process of catching up is not only labor-intensive but also prone to human errors. Important details might be overlooked or misunderstood, leading to further miscommunication and misalignment. The time spent on this manual review could be better utilized in productive activities that directly contribute to the project's success.

4. Delayed access to information

- Issue: Accessing meeting details, such as agenda, updates, and action items, is not immediate and requires sifting through extensive documentation.
- Impact: This delay can prevent timely action on important tasks, leading to missed deadlines and decreased productivity. In fast-paced projects, timely access to information is crucial for making informed decisions and taking prompt actions. Any delay in accessing the necessary information can cause a bottleneck in the workflow, slowing down the entire project's momentum and potentially leading to missed opportunities or deadlines.

5. Lack of real-time updates

- Issue: Team members who miss meetings do not have real-time access to the latest project developments, impacting their ability to contribute effectively.
- Impact: This can lead to a lack of synchronization within the team, with some members working on outdated information while others are up-to-date. This disparity in information can create confusion and disjointed efforts within the team, reducing overall efficiency and effectiveness. It becomes challenging to maintain a cohesive direction when team members are not on the same page regarding the project's status and next steps.

6. Coordination and collaboration challenges

- Issue: Missing out on meeting information hampers coordination and collaboration among team members, leading to potential misunderstandings and misaligned goals.
- Impact: This lack of coordination can result in duplicated efforts, conflicting actions, and a project that is less cohesive and aligned. Effective collaboration relies on all team members having access to the same information and understanding the project's goals and objectives. When this is not the case, it can lead to fragmented efforts, with team members inadvertently working at cross-purposes, thus diminishing the overall quality and progress of the project.

7. Reduced accountability and transparency

- Issue: Without equal access to meeting information, it becomes difficult to maintain accountability and transparency within the team.
- Impact: Team members may not be fully aware of their responsibilities and the progress of tasks, leading to reduced accountability and potential conflicts. Transparency is key to building trust and ensuring that everyone is held accountable for their contributions.

8. **Fear of missing out (FOMO)**

- Issue: Team members who cannot attend meetings experience FOMO, which affects their engagement and sense of inclusion in the project.
- Impact: This can lead to decreased morale, lower engagement levels, and reduced contributions from affected team members. The fear of missing out on important discussions and decisions can make team members feel disconnected and undervalued, impacting their motivation and willingness to contribute actively to the project. This disengagement can further exacerbate the information gap and reduce overall team performance.

9. **Administrative overhead**

- Issue: Team leads and administrators spend significant time documenting and sharing meeting details, which could be automated to improve efficiency.
- Impact: This administrative burden takes away time that could be better spent on strategic activities and decision-making. The process of manually compiling and distributing meeting notes is not only time-consuming but also detracts from more critical, value-adding activities that team leads and administrators could be focusing on. Automating this process would free up their time, allowing them to concentrate on driving the project forward and making strategic decisions.

10. **Barrier to contribution**

- Issue: Without easy access to meeting details, team members may feel less equipped to contribute effectively, impacting overall team performance and project outcomes.
- Impact: This barrier can lead to disengagement and reduced productivity, as team members may feel disconnected from the project's progress and goals. When team members do not have the information they need, they are less likely to take initiative and contribute meaningfully. This can result in a lack of innovation, slower problem-solving, and a general decrease in the team's effectiveness and productivity.

CHAPTER 3

OBJECTIVE AND MOTIVATION

3.1 Problem Statement

When a team member misses a meeting, they often struggle to catch up on the insights and decisions made during that time. Without access to the meeting notes or data sheets shared during the meeting, staying informed can be challenging. It becomes even more difficult if the information is scattered across various sources or if there's no easy way to retrieve the necessary documents.

3.2 Objectives:

1. To ensure easy access to meeting information team members can access critical meeting data anytime, promoting consistency and cohesion despite scheduling conflicts.
2. To streamlining information retrieval by allowing team members to access needed insights quickly and effortlessly through natural language commands.
3. To fostering collaboration and inclusivity by ensuring that every team member can access and contribute to meeting insights, regardless of their attendance.
4. To promote transparency and accountability by democratizing access to meeting discussions and decisions.
5. To empower informed decision-making which empowers team members to make informed decisions by providing seamless access to accurate and up-to-date meeting data.
6. To bridge information gaps which ensures that all team members, regardless of their location or meeting attendance, have equal access to the same information.
7. To provide accurate and consistent information wich ensures the accuracy and consistency of information shared within the team by extracting data directly from source documents.
8. To enhance team effectiveness and success which enhances team effectiveness by providing easy access to meeting information and streamlining information retrieval.

3.3 Scope & Motivation

Scope: The scope of ConCraft encompasses several key areas that are vital to its successful implementation and utility within an organization. By integrating with existing systems, facilitating seamless user interaction, ensuring usability and accessibility, enhancing collaboration and knowledge sharing, and addressing specific needs in FinOps companies, ConCraft aims to revolutionize how teams manage and access meeting information.

1. **Integration with existing systems**

- Comprehensive System Integration: ConCraft will be integrated with the organization’s current systems used for storing and managing meeting datasheets. This integration is crucial to ensure that the chatbot can effectively access, retrieve, and process information. Depending on the organization's infrastructure, this may involve connecting to various cloud storage solutions, databases, or collaboration platforms where datasheets are stored.
- Ensuring Data Compatibility: A significant part of this integration process will involve ensuring data compatibility. ConCraft will be designed to work seamlessly with different file formats and storage systems, such as Google Drive, Microsoft OneDrive, SharePoint, and other enterprise data management systems. This will ensure that regardless of where the datasheets are stored, ConCraft can access and process them efficiently.
- Secure Data Access: Security is a paramount concern in any data integration process. ConCraft will implement robust security protocols to ensure that data access is both secure and compliant with organizational policies and industry regulations. This includes utilizing encryption, access controls, and secure authentication mechanisms to protect sensitive meeting information.

2. **User interaction and information retrieval**

- Natural language processing (NLP): ConCraft will enable team members to interact with it using natural language commands. The chatbot will leverage advanced NLP capabilities to understand and respond to user queries accurately. Users can request specific information from the datasheets, such as meeting agendas, project updates, action items, and key decisions, through simple and intuitive commands.

- Query customization: The chatbot will support a wide range of queries, allowing users to customize their information retrieval based on their specific needs. Whether a team member needs a summary of the last meeting, details about a particular decision, or a list of action items assigned to them, ConCraft will provide prompt and accurate responses.
- Real-time information access: ConCraft's integration with backend systems will enable real-time access to updated information. This means that as soon as meeting datasheets are updated or new information is added, it becomes immediately accessible to users through the chatbot. This real-time access is crucial for maintaining up-to-date knowledge and making timely decisions.

3. Usability and accessibility

- Intuitive user interface: ConCraft will be designed with a strong emphasis on usability, ensuring that all team members can easily interact with the chatbot. The user interface will be intuitive, featuring simple navigation and clear instructions. Users will not require any technical expertise to utilize ConCraft effectively.
- Multiple access modes: To accommodate different user preferences and work environments, ConCraft will support multiple modes of access. This includes web applications, mobile apps, and integrations with popular chat platforms such as Slack, Microsoft Teams, and WhatsApp.
- Accessibility features: ConCraft will incorporate accessibility features to support users with varying needs. This includes voice commands for hands-free operation, screen reader compatibility for visually impaired users, and multilingual support to cater to non-English speakers. These features ensure that ConCraft is inclusive and usable by all team members.

4. Facilitating collaboration and knowledge sharing

- Enhanced collaboration: By providing quick and easy access to meeting information, ConCraft will significantly enhance collaboration within the team. Team members can stay updated on project developments, understand their responsibilities, and contribute effectively, even if they miss a meeting. This fosters a more cohesive and engaged team.
- Knowledge sharing: ConCraft will serve as a central hub for meeting-related information, facilitating better knowledge sharing within the organization. Team members can easily access historical data, review past decisions, and understand the context of ongoing

projects. This comprehensive access to information helps in maintaining continuity and coherence in team efforts.

- Transparency and accountability: The transparency provided by ConCraft ensures that all team members have access to the same information. This level of transparency promotes accountability, as everyone is aware of decisions made and actions assigned. It also helps in building trust within the team, as there are no information silos or discrepancies.

5. Application in FinOps companies

- Streamlining financial operations: While ConCraft is beneficial for any team-oriented environment, its utility is particularly pronounced in FinOps companies. Financial operations require constant updates, precise information, and timely decision-making. ConCraft will help these companies streamline their operations by ensuring that all relevant financial data and meeting outcomes are readily accessible to the entire team.
- Real-time financial insights: In FinOps companies, access to real-time financial insights is crucial. ConCraft will enable team members to quickly retrieve up-to-date financial information, meeting summaries, and action items related to financial operations. This timely access to information supports better financial planning, analysis, and decision-making.
- Regulatory compliance and reporting: FinOps companies often need to adhere to strict regulatory requirements and reporting standards. ConCraft can assist in ensuring compliance by providing easy access to meeting records, financial decisions, and action plans. This streamlined access to documentation simplifies the process of preparing reports and audits.
- Facilitating financial collaboration: Financial operations often involve collaboration between various departments and stakeholders. ConCraft will facilitate this collaboration by providing a unified platform for accessing meeting information. Team members from different departments can stay informed and aligned, ensuring cohesive and coordinated financial operations.

Motivation:

The motivation behind ConCraft is rooted in a deep understanding of the challenges faced by teams in today's fast-paced work environments. We recognize that in the dynamic landscape of modern workplaces, missing meetings can occur for a variety of reasons, from conflicting schedules to unexpected emergencies. However, we also understand the critical importance of staying updated on crucial information and decisions discussed during these meetings. ConCraft aims to address this challenge head-on by providing a comprehensive solution that empowers every team member to effortlessly access and retrieve insights from meeting datasheets, even when they couldn't be present physically. Through the innovative integration of Amazon Lex and AWS Lambda, we've developed an intelligent chatbot that serves as a reliable virtual assistant, available to assist team members with their information needs anytime, anywhere. Our motivation stems from a deep-seated desire to foster collaboration, transparency, and inclusivity within teams. We firmly believe that by ensuring everyone has equal access to the same information, we can enhance communication, streamline decision-making processes, and ultimately drive success for the entire team. With ConCraft, no team member is left behind. Regardless of their attendance at meetings, every individual can feel connected, informed, and empowered to contribute their best to the team's objectives. By democratizing access to meeting insights and promoting open communication, ConCraft creates a culture where every voice is heard and valued, leading to increased engagement, motivation, and productivity. Ultimately, our vision for ConCraft extends beyond just providing a technical solution. We aspire to create a transformative tool that not only facilitates information retrieval but also cultivates a sense of belonging and ownership among team members. Through ConCraft, we aim to break down barriers, bridge gaps, and empower teams to achieve their full potential, united in their pursuit of common goals. Together, we can build a future where every team member has the opportunity to thrive, contribute, and succeed.

- **Empowering team members with real-time assistance:** ConCraft leverages advanced natural language processing (NLP) capabilities to provide real-time responses to team members' queries about meeting content. This immediate access to information ensures that all team members, whether they attended the meeting or not, can stay up-to-date with project developments. By enabling team members to ask questions in everyday language, ConCraft democratizes access to crucial information, fostering a more informed and proactive workforce. This empowerment translates to higher engagement and productivity, as team members can contribute more effectively to ongoing projects.

- **Promoting inclusivity and transparency:** By ensuring that every team member has equal access to meeting information, ConCraft fosters an environment of transparency and inclusivity. This equal access helps mitigate feelings of exclusion among team members who may not be able to attend every meeting. The inclusive nature of ConCraft promotes a culture where every voice is heard and valued, enhancing team cohesion and trust. Transparency in information sharing also reduces the risk of miscommunication and misunderstandings, paving the way for more effective collaboration and decision-making.

- **Streamlining decision-making processes:** ConCraft enhances the efficiency of decision-making by providing quick access to meeting minutes, action items, and other critical information. This rapid retrieval of accurate information allows team members to make informed decisions promptly, reducing delays and improving overall project execution. In fast-paced environments, the ability to make quick, informed decisions is crucial for maintaining agility and meeting deadlines. ConCraft's reliable information flow ensures that decisions are based on the most recent data, minimizing the risk of errors and rework.

- **Reducing cognitive load:** Manually sifting through extensive meeting documentation can be time-consuming and mentally taxing. ConCraft alleviates this burden by providing instant access to specific meeting information through a simple query interface. By reducing the cognitive load on team members, ConCraft allows them to focus more on their core responsibilities and less on administrative tasks. This shift not only enhances individual productivity but also contributes to a more efficient and effective team dynamic.

- **Enhancing communication and collaboration:** Effective communication is the cornerstone of successful teamwork. ConCraft ensures that all team members are on the same page by providing a reliable means of retrieving and disseminating meeting information. This consistency in communication helps prevent misunderstandings and errors, fostering a more collaborative and harmonious work environment. Enhanced communication also means that team members can share insights and feedback more effectively, driving innovation and continuous improvement within the team.

- **Building trust and accountability:** Trust is built on transparency and accountability. ConCraft promotes these values by ensuring that all team members have access to the same information. When information is shared openly, team members feel more confident in the decision-making processes and more accountable for their contributions. This level of trust and accountability enhances team morale and encourages a more engaged and committed workforce. It also mitigates the risk of conflicts arising from misinformation or lack of information.

- **Facilitating remote and distributed work:** In an era where remote work and distributed teams are becoming the norm, having a reliable means of staying connected is essential. ConCraft addresses this need by providing seamless access to meeting information regardless of team members' physical locations. This connectivity ensures that remote and distributed team members can participate fully in projects and decision-making processes, fostering a sense of inclusion and unity within the team. ConCraft's capabilities bridge the gap created by geographical distances, ensuring that the entire team remains cohesive and productive.

- **Supporting continuous improvement:** The interactions and feedback collected through ConCraft can be invaluable for identifying areas of improvement in team processes and communication strategies. By analyzing this feedback, teams can continuously refine and enhance their workflows, ensuring they are always evolving and adapting to new challenges. This commitment to continuous improvement helps maintain a competitive edge and drives the team towards excellence. ConCraft thus not only addresses immediate needs but also contributes to the long-term success and resilience of the team.

- **Mitigating information overload:** In today's fast-paced work environment, information overload can be a significant challenge. ConCraft helps mitigate this by providing concise and relevant information upon request, preventing team members from being overwhelmed by unnecessary details. This targeted information delivery allows team members to stay focused on what matters most, improving their efficiency and effectiveness. By streamlining information access, ConCraft ensures that team members can quickly find what they need without getting bogged down by extraneous data.

- **Fostering a culture of openness and trust:** When information is easily accessible to all team members, it fosters a culture of openness and trust. ConCraft's ability to provide instant access to meeting details ensures that everyone is informed and aligned, reducing the chances of misunderstandings and miscommunications. This transparency builds trust within the team, as members feel confident that they are all working with the same information. A culture of openness and trust enhances collaboration and creates a positive work environment where team members feel valued and respected.

- **Enhancing team morale:** When team members feel informed and included, their morale improves. ConCraft ensures that all team members have access to the same information, fostering a sense of belonging and purpose. This inclusivity boosts team morale, as members feel more connected to the project and more valued for their contributions. Higher morale translates to increased motivation and productivity, as team members are more likely to go above and beyond to achieve collective goals. ConCraft's impact on team morale is thus a key driver of overall team success.

- **Improving project execution:** Accurate and timely information is critical for effective project execution. ConCraft ensures that team members have quick access to the latest meeting minutes, decisions, and action items, enabling them to stay updated and make Informed decisions. This reliable information flow reduces the risk of errors and rework, leading to more efficient and effective project execution. By keeping the team aligned and informed, ConCraft helps ensure that deadlines are met and project goals are achieved.

- **Empowering non-technical team members:** ConCraft's user-friendly interface allows team members to query the chatbot using everyday language, making it accessible to all, regardless of their technical expertise. This democratization of information access empowers non-technical team members to stay informed and contribute effectively to projects. By making information accessible to everyone, ConCraft ensures that all team members can leverage their unique skills and insights, enhancing overall team performance and innovation.

- **Ensuring consistency in information sharing:** Consistent information sharing is crucial for maintaining a coherent direction for the team. ConCraft ensures that all team members receive the same information, reducing discrepancies and potential conflicts. This consistency helps maintain alignment within the team, ensuring that everyone is working towards the same goals with the same understanding. Consistent information sharing also reinforces accountability, as team members are aware of their responsibilities and the expectations placed upon them.

- **Driving innovation and creativity:** By ensuring that all team members have access to the same information, ConCraft fosters an environment where innovation and creativity can thrive. When team members are well-informed, they can build on each other's ideas more effectively, leading to innovative solutions and creative problem-solving. ConCraft's role in facilitating transparent communication and information sharing is thus a catalyst for innovation, driving the team towards continuous improvement and success.

- **Facilitating onboarding and training:** New team members often face a steep learning curve when joining a project. ConCraft can ease this transition by providing instant access to past meeting information, project updates, and action items. This accessibility helps new team members quickly get up to speed, reducing the time required for onboarding and training. By streamlining the onboarding process, ConCraft ensures that new hires can become productive members of the team more quickly, enhancing overall team efficiency.
- **Enhancing employee engagement:** Engaged employees are more productive, creative, and committed to their work. ConCraft enhances employee engagement by ensuring that team members have easy access to the information they need to do their jobs effectively. This accessibility empowers employees, making them feel valued and supported. Engaged employees are more likely to contribute innovative ideas and solutions, driving the team towards greater success. ConCraft's role in enhancing engagement is thus a critical factor in building a high-performing team.

- **Facilitating knowledge management:** Effective knowledge management is essential for organizational success. ConCraft contributes to this by ensuring that all meeting information is easily accessible and retrievable. This organized and centralized access to knowledge helps prevent the loss of valuable information and ensures that team members can leverage past insights and decisions. By facilitating effective knowledge management, ConCraft enhances the team's ability to learn from past experiences and improve future performance.

- **Supporting compliance and record-keeping:** Maintaining accurate records of meetings and decisions is crucial for compliance and accountability. ConCraft ensures that all meeting information is documented and easily accessible, supporting the team's compliance and record-keeping efforts. This reliable documentation helps the team meet regulatory requirements and provides a clear audit trail for future reference. By supporting compliance, ConCraft helps protect the organization from potential legal and regulatory issues.

- **Reducing meeting fatigue:** Excessive meetings can lead to fatigue and reduced productivity among team members. ConCraft helps alleviate this by ensuring that team members can access crucial meeting information without needing to attend every meeting. This reduces the need for frequent meetings and allows team members to focus more on their core responsibilities. By reducing meeting fatigue, ConCraft helps maintain high levels of productivity and engagement within the team, contributing to overall success.

CHAPTER 4

PROPOSED METHODOLOGY

4.1 System Architectural Diagram

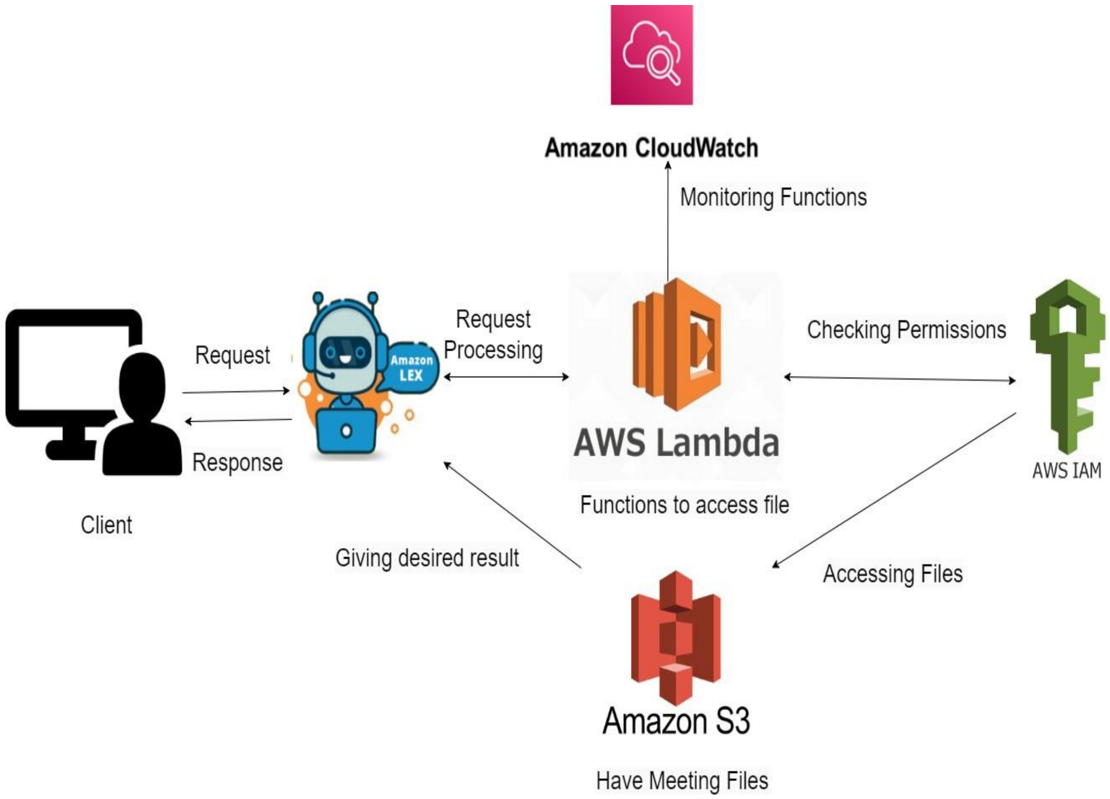


Fig 4.1: System architectural diagram

The Fig 4.1 leverages several AWS services to create a robust, scalable, and efficient solution for managing and retrieving meeting files based on user requests. By combining Amazon Lex, AWS Lambda, Amazon S3, IAM, and CloudWatch, the system ensures seamless interaction between components, secure access to resources, and continuous monitoring for optimal performance. This architecture not only enhances productivity and communication within teams but also provides a scalable foundation for future expansions and integrations.

1. **Client:** The client serves as the entry point for user interactions. This could be a user interacting through a web or mobile application, issuing requests to the system. These requests might include commands such as retrieving meeting details, updating meeting

records, or asking specific questions about meeting content. The client interface must be intuitive and user-friendly, ensuring that users can easily input their requests in natural language. The client application sends these requests to Amazon Lex for processing, initiating the workflow within the AWS ecosystem.

2. **Amazon Lex:** Amazon Lex is a service for building conversational interfaces using voice and text. It leverages advanced natural language understanding (NLU) and automatic speech recognition (ASR) capabilities. When a client submits a request, Amazon Lex receives and processes it, understanding the user's intent. Lex is configured with intents and slots, which represent the actions the user wants to perform and the necessary parameters for those actions, respectively. After interpreting the request, Lex determines if additional information is required from the user or if the request can be directly passed to a backend service, such as an AWS Lambda function, for further processing.

3. **AWS Lambda:** AWS Lambda is a serverless compute service that allows code execution in response to predefined events without provisioning or managing servers. In this architecture, Lambda functions are triggered by Amazon Lex to handle user requests. Lambda functions encapsulate the business logic necessary for processing these requests. For instance, if a user asks for specific meeting details, the Lambda function will be responsible for querying the appropriate data source. Lambda functions interact with various AWS services, such as Amazon S3, for data retrieval or storage. The stateless nature of Lambda ensures scalability and resilience, as each request can trigger a new instance of the function, running in parallel to others.

4. **Amazon S3 (Simple Storage Service):** Amazon S3 is a scalable object storage service used for storing and retrieving any amount of data. In this architecture, S3 is employed to store meeting files, including documents, recordings, and other related data. Lambda functions access S3 to retrieve or store these files as part of their processing logic. S3 ensures high availability and durability, making it a reliable storage option for critical meeting data. When a Lambda function needs to access a file, it sends a request to S3 using the appropriate API calls. This interaction involves reading from and writing to S3 buckets, which are the fundamental containers in S3 for storing data.

5. **AWS IAM (Identity and Access Management):** AWS IAM manages access to AWS resources, ensuring that the right users and services have the appropriate permissions. In this system, IAM roles and policies are defined to regulate access to Amazon S3 and other resources by Lambda functions. When a Lambda function is invoked, IAM checks the function's permissions to ensure it has the rights to perform the required actions. For example, IAM policies will define whether a Lambda function can read from or write to specific S3 buckets. This security layer is crucial for maintaining the integrity and confidentiality of the data within the system.

6. **Amazon CloudWatch:** Amazon CloudWatch is a monitoring and management service built for developers, system operators, and IT managers. It provides data and actionable insights to monitor applications, understand and respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. In this architecture, CloudWatch is used to monitor the performance of Lambda functions. It tracks metrics such as invocation count, duration, and error rates. CloudWatch also logs function execution details, which are invaluable for debugging and performance tuning. Alerts can be set up in CloudWatch to notify administrators of any anomalies or performance issues, ensuring that the system operates smoothly.

Flow of Interaction:

1. **Request initiation by client:**

- The user initiates a request through a web or mobile application. This request is composed in natural language and sent to Amazon Lex. The client application must be capable of handling user inputs and sending structured requests to Lex.

2. **Request processing by Amazon Lex:**

- Amazon Lex receives the user's request and uses its NLU capabilities to interpret the intent. It determines the necessary actions and data required to fulfill the request. If additional information is needed, Lex prompts the user for more input.

3. **Invocation of AWS Lambda:**

- Once Lex has all necessary information, it invokes an AWS Lambda function to process the request. Lex passes relevant data to the Lambda function as parameters.

4. **Permission check via IAM:**

- Before the Lambda function can access any resources, IAM checks its permissions. IAM ensures that the Lambda function has the rights to interact with the specified resources, such as reading from or writing to an S3 bucket.

5. **Interaction with Amazon S3:**

- With the necessary permissions confirmed, the Lambda function performs its logic. If the request involves accessing or manipulating meeting files, the Lambda function interacts with Amazon S3. This could involve retrieving a file from an S3 bucket, processing its content, or uploading a new file to S3.

6. **Response handling by Lambda:**

- After completing the required operations, the Lambda function processes the results. It formats the response appropriately and sends it back to Amazon Lex.

7. **Response delivery by Amazon Lex:**

- Amazon Lex receives the processed response from the Lambda function. It then communicates this response back to the client application, delivering the information the user requested.

8. **Monitoring by Amazon CloudWatch:**

- Throughout this process, Amazon CloudWatch monitors the performance of the Lambda functions. It logs execution details, tracks performance metrics, and provides insights for optimizing the system. Alerts can be configured to notify administrators of any issues, ensuring proactive management of the system.

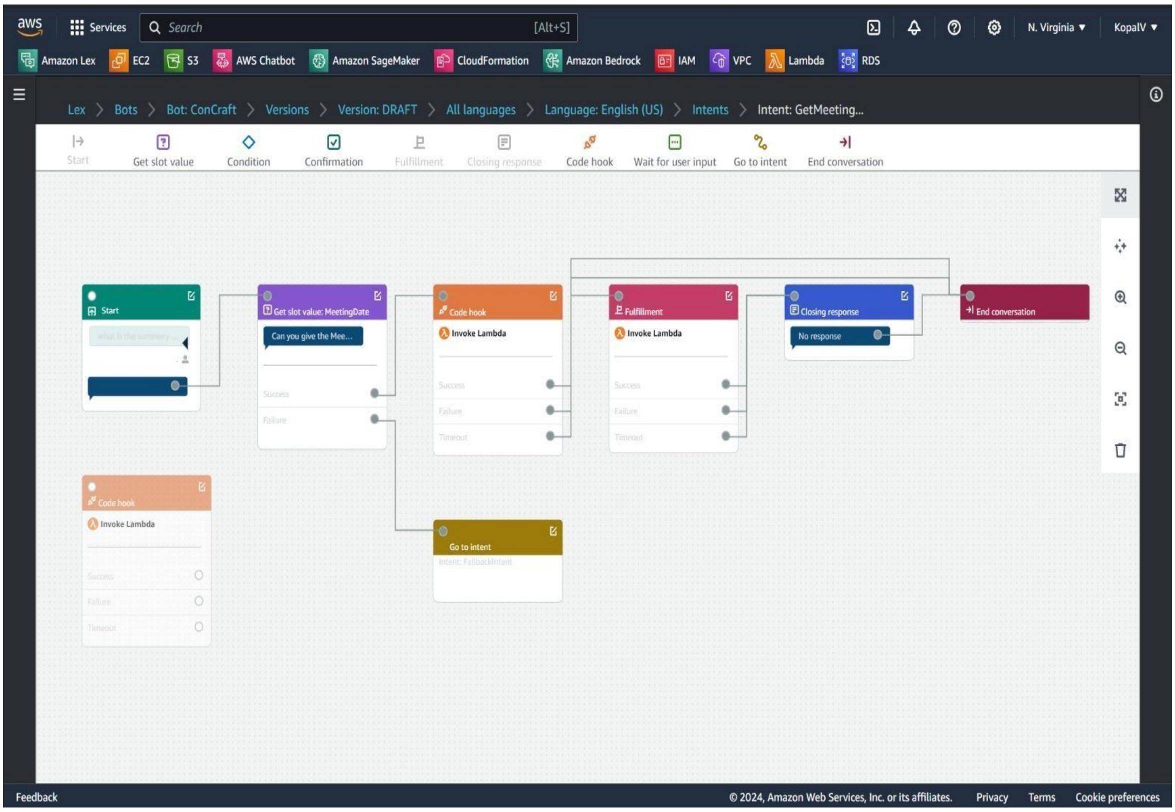


Fig 4.2: Topology of project

4.2 Functional & Non-Functional Requirements

Functional Requirement:

1. **Meeting data input:** Meeting data input is the foundational step in ensuring that ConCraft functions effectively. During meetings, all significant discussions, decisions, and action items are meticulously documented by the team lead or designated scribe. This data is then input into a structured datasheet, capturing every essential detail. The accuracy and comprehensiveness of this input are crucial, as it forms the basis for all subsequent information retrieval and processing tasks performed by ConCraft. Ensuring that the datasheet is updated promptly and accurately after each meeting guarantees that team members have access to the latest information, thereby maintaining continuity and clarity in team operations.

2. **Information retrieval:** Information retrieval is at the core of ConCraft's functionality. The chatbot enables team members to query and extract specific pieces of information from the meeting datasheets quickly and efficiently. Users can employ natural language commands to ask ConCraft for details about meeting agendas, project updates, decisions

made, and action items assigned. This process eliminates the need for manual searching and sifting through documents, significantly reducing the time and effort required to find relevant information. The ability to retrieve information on demand ensures that team members stay informed and can act on the latest updates promptly.

3. **Data processing and storage:** Data processing and storage involve handling the meeting datasheets to ensure they are stored securely and processed accurately for easy retrieval. When a datasheet is uploaded, ConCraft uses AWS Lambda functions to parse and structure the data, making it readily accessible for user queries. The system stores the processed data in a secure and organized manner, often leveraging cloud storage solutions like Amazon S3. This setup not only ensures data integrity and security but also allows for scalable storage that can handle large volumes of meeting data over time. Efficient data processing and storage are critical for maintaining the reliability and performance of ConCraft.

4. **User commands and responses:** User commands and responses are integral to ConCraft's interactive capabilities. Users interact with the chatbot through natural language commands, asking specific questions or requesting information. ConCraft, powered by Amazon Lex, processes these commands, interprets the user's intent, and provides accurate and relevant responses. This interaction needs to be seamless and intuitive, ensuring that users receive the information they need without frustration. The chatbot's ability to understand various phrasing and context is crucial for delivering precise answers, thereby enhancing the user experience and utility of ConCraft.

5. **User interface:** The user interface (UI) of ConCraft is designed with usability and accessibility in mind. It provides an intuitive and straightforward platform for team members to interact with the chatbot. Whether accessed through a web application, mobile app, or integrated chat platforms, the UI must be consistent and user-friendly. The interface includes features like a chat window for entering commands, visual aids for displaying retrieved data, and navigation tools for accessing different functionalities of the chatbot. A well-designed UI ensures that all team members, regardless of technical proficiency, can easily interact with ConCraft and access the information they need.

6. **Collaboration and sharing:** Collaboration and sharing are enhanced through ConCraft's ability to disseminate meeting information efficiently among team members. By providing a centralized platform where all meeting data is stored and can be accessed, ConCraft ensures that everyone is on the same page. Team members can share retrieved information, discuss updates, and coordinate actions based on the latest meeting insights. This capability fosters a collaborative environment where knowledge is shared openly, transparency is maintained, and collective decision-making is encouraged. The platform's design supports continuous engagement and communication, ensuring that all team members can contribute effectively, regardless of their physical presence at meetings.

Non Functional Requirement

1. **Security:** Security is a paramount non-functional requirement for ConCraft. Given the sensitive nature of meeting discussions, decisions, and action items, it is crucial that the platform ensures data confidentiality, integrity, and availability. ConCraft must implement robust encryption protocols for data at rest and in transit to protect against unauthorized access and data breaches. Access control mechanisms should be in place to ensure that only authorized personnel can retrieve or manipulate information. Regular security audits and compliance with relevant standards and regulations, such as GDPR or HIPAA (if applicable), are essential to maintain the trust of users and protect the organization from potential legal and financial repercussions.

2. **Portability:** Portability refers to ConCraft's ability to function seamlessly across different environments and devices. This requirement ensures that the chatbot can be accessed and used on various platforms, including web browsers, mobile devices (iOS and Android), and desktop applications. By being platform-agnostic, ConCraft maximizes its usability and accessibility, allowing team members to interact with the chatbot regardless of their preferred device or operating system. This flexibility is critical in modern work environments where team members may use a variety of devices and need to access information on-the-go. Additionally, portability ensures that ConCraft can be easily integrated into existing organizational ecosystems without significant modifications.

3. **Performance:** Performance is a critical non-functional requirement that impacts the user experience and overall efficiency of ConCraft. The platform must deliver quick response times, handling queries and retrieving information from datasheets promptly to maintain user satisfaction. Performance benchmarks should include low latency, high throughput, and the ability to handle concurrent user sessions without degradation in service quality. ConCraft should be designed to scale efficiently to accommodate varying loads, ensuring consistent performance during peak usage times. Regular performance testing and optimization are necessary to identify and mitigate potential bottlenecks, ensuring that the chatbot remains responsive and reliable under all operating conditions.

4. **Error handling and logging:** Effective error handling and logging are essential for maintaining the reliability and user trust in ConCraft. The platform should be designed to gracefully handle various types of errors, whether they are related to user input, system failures, or network issues. Clear and informative error messages should be provided to users, guiding them on how to resolve issues or contact support if necessary. Comprehensive logging mechanisms are necessary to track and record all system events, errors, and user interactions. These logs are invaluable for diagnosing and troubleshooting issues, performing audits, and enhancing system security. Regular monitoring and analysis of logs help in identifying patterns that could indicate underlying problems, allowing for proactive maintenance and improvements.

5. **Help and support:** Providing robust help and support features is a crucial non-functional requirement for ensuring a positive user experience with ConCraft. The platform should include comprehensive documentation and user guides that cover all aspects of its functionality, from initial setup and configuration to advanced usage scenarios. An intuitive help system should be integrated within the chatbot, allowing users to access context-sensitive help directly from the interface. Additionally, a responsive support system, including live chat support, email assistance, and a knowledge base, should be available to address user queries and issues promptly. Training materials and tutorials can further empower users to make the most of ConCraft's features, enhancing their productivity and satisfaction with the platform.

4.3 Use case/ Activity Diagram

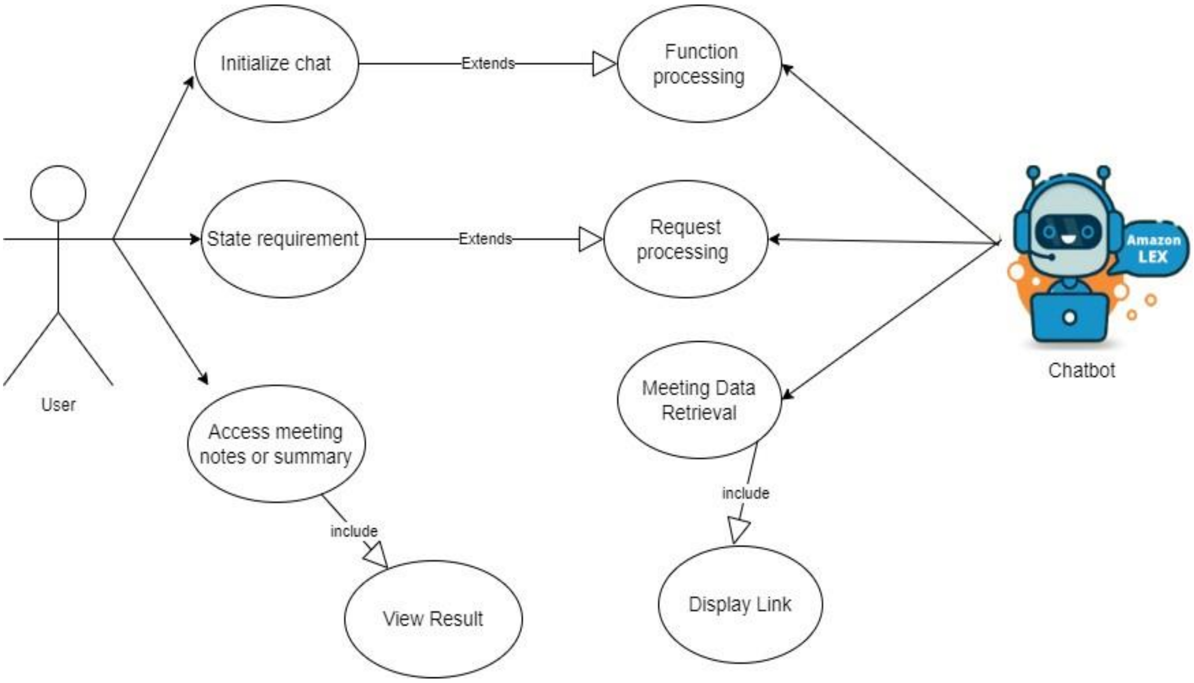


Fig 4.3: ConCraft Use Case Diagram

The Fig 4.3 depicts the interaction between a user and a chatbot, powered by Amazon Lex, for accessing meeting notes or summaries. It illustrates the sequence of actions and the flow of information between the user and the system components, showcasing how ConCraft facilitates streamlined information retrieval. Here's a detailed explanation of the elements and the entire process:

Actors:

- User: The individual who interacts with the system to retrieve meeting notes or summaries. The user could be any team member who needs to stay informed about meeting discussions, decisions, and action items.
- Chatbot (Amazon Lex): The system component that interacts with the user, processes their requests, and retrieves the required data. Amazon Lex powers the chatbot, enabling it to understand and respond to natural language queries effectively.

Use Cases:

- **Initialize chat**: The user starts a conversation with the chatbot. This action extends to Function processing, indicating that certain backend functions are triggered to initialize the chat session properly. This may involve setting up a session, initializing context, and preparing the chatbot to handle requests.

- **State requirement**: The user specifies their request or requirement, such as asking for meeting notes or a summary. This action extends to Request processing, meaning it triggers the chatbot to understand and process the user's request. The chatbot interprets the natural language query and determines the best way to fulfill the requirement.
- **Access meeting notes or summary**: The user requests access to specific meeting data. This action includes viewing the result, meaning the user eventually wants to see the output of their request. The user may ask for specific details from the meeting, such as action items, decisions made, or general summaries.

Processes within chatbot (Amazon Lex):

- **Function processing**: Backend processes that are triggered during the initialization of the chat session. This might include setting up the session context, fetching user-specific data, and ensuring that the chatbot is ready to handle subsequent requests efficiently.
- **Request processing**: The chatbot processes the user's stated requirements by interpreting the natural language input, understanding the context, and deciding on the best way to fulfill the request. This involves analyzing the user's query, mapping it to relevant data retrieval functions, and ensuring accuracy in the information provided.
- **Meeting data retrieval**: The chatbot retrieves the required meeting data based on the user's request. This involves querying the relevant databases or data sources to fetch the necessary meeting notes or summaries. Display Link is a part of this process, where a link to the meeting data is provided to the user. This link might lead to a detailed document, a web page, or a summarized view of the meeting notes.
- **View result**: The user views the result of their request, which could be the meeting notes or summary. This step ensures that the user has access to the information they need in a clear and understandable format.

Flow Summary:

- **Initialization:** The user initiates a chat with ConCraft, setting the stage for subsequent interactions.
- **Requirement stating**: The user states their requirement (e.g., requests for meeting notes). This is where the user articulates what they need from the system, leveraging natural language.

- **Processing**: The chatbot processes the function and request. During this phase, the chatbot's backend processes kick in, interpreting the request and preparing to retrieve the necessary data.
- **Data retrieval**: The chatbot retrieves the necessary meeting data. This involves accessing databases, fetching documents, and compiling the requested information.
- **Result display**: The chatbot displays a link (if applicable) to the retrieved data. This link provides the user with a direct path to view the meeting notes or summary.
- **Access and view**: The user accesses the meeting notes or summary by clicking the provided link. This step allows the user to review the retrieved information and ensures they have the details needed to stay informed.
- **Viewing results:** Finally, the user views the results. This could involve reading through the meeting notes, examining key points, and understanding the decisions and action items discussed during the meeting.

Detailed Explanation:

The use case diagram and its components highlight the efficiency and user-centric design of ConCraft. By enabling seamless interaction between users and the chatbot, ConCraft addresses the critical need for timely and accurate information retrieval. The chatbot's ability to process natural language queries ensures that users do not need specialized training to use the system, making it accessible to everyone within the team.

The integration with backend processes ensures that the chatbot can handle various requests dynamically, providing a robust and scalable solution for information retrieval. The emphasis on viewing results through a direct link ensures that users can access detailed information without unnecessary navigation, streamlining the overall user experience.

Overall, ConCraft's use case diagram underscores its potential to enhance team collaboration, transparency, and productivity by providing an intuitive and efficient means of accessing meeting information.

4.4 Flow Chart

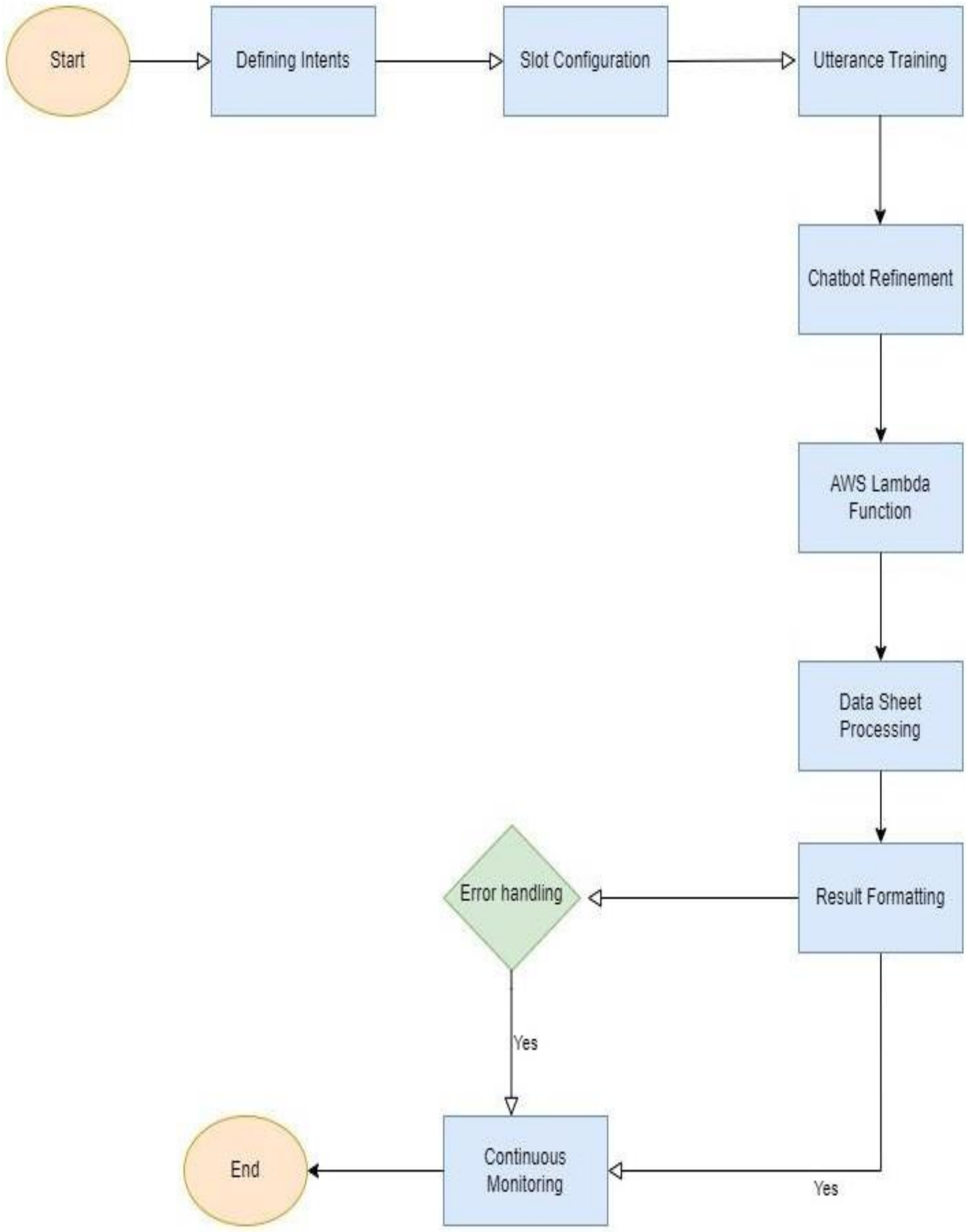


Fig 4.4: The flow diagram

Here's a step-by-step explanation of each phase:

1. **Start:**

- The process begins with the initiation of the ConCraft project. This step marks the entry point into the sequence of actions required to develop and implement the chatbot.

2. **Defining intents:**

- Purpose: Identify the primary functions and objectives of the chatbot.
- Activities: Determine what the chatbot needs to accomplish, such as retrieving meeting agendas, summarizing action items, or providing project updates. Each function is defined as an intent.

3. **Slot configuration:**

- Purpose: Specify the types of data the chatbot will need to fulfill each intent.
- Activities: Configure slots, which are variables that capture user input related to each intent. For example, for an intent to retrieve meeting notes, slots might include the date of the meeting, project name, or specific keywords.

4. **Utterance training:**

- Purpose: Train the chatbot to understand different ways users might express the same request.
- Activities: Provide a variety of sample utterances for each intent. These are different phrases or sentences users might say to trigger the intent. For example, for the intent "Get meeting summary," sample utterances could be "Give me the meeting summary," "What happened in the last meeting?" or "Summarize the last meeting for me."

5. **Chatbot refinement:**

- Purpose: Improve the accuracy and performance of the chatbot.
- Activities: Continuously test the chatbot with different utterances and refine the intents and slots based on feedback. Adjust the training data and retrain the model to handle new scenarios or improve existing responses.

6. **AWS Lambda function:**

- Purpose: Implement serverless computing to handle backend processes.
- Activities: Develop and deploy AWS Lambda functions to execute specific tasks when triggered by the chatbot. These functions handle data retrieval, processing, and interaction with other AWS services, ensuring the chatbot can access and manipulate the datasheets.

7. **Data sheet processing:**

- Purpose: Extract and process information from meeting datasheets.
- Activities: Parse the datasheets to identify and organize the relevant information. This may involve reading structured data, extracting key details, and converting them into a format that the chatbot can use to respond to queries.

8. **Result formatting:**

- Purpose: Prepare the extracted data for user-friendly presentation.
- Activities: Format the processed data into clear and concise responses. Ensure that the information is presented in a way that is easy for users to understand, whether through text, tables, or other formats.

9. **Error handling:**

- Purpose: Manage and resolve any issues that arise during data processing or response generation.
- Activities: Implement mechanisms to detect and handle errors, such as missing data, incorrect input, or processing failures. Provide meaningful error messages to users and log errors for further analysis.

10. **Continuous monitoring:**

- Purpose: Ensure the ongoing performance and reliability of ConCraft.
- Activities: Monitor the chatbot's interactions and performance in real-time. Collect metrics and user feedback to identify areas for improvement. This involves checking for any anomalies or issues that need to be addressed.

11. **Decision point:**

- Purpose: Determine if there are any errors that need to be addressed.
- Activities: After processing and formatting results, check if there were any errors. If errors are detected, the system loops back to the error handling phase for resolution. If no errors are found, the process proceeds to continuous monitoring.

12. **End:**

- The process concludes, either temporarily until the next user interaction or as part of a cyclic process where continuous monitoring ensures the chatbot's ongoing functionality and improvement.

CHAPTER 5

IMPLEMENTATION AND RESULT ANALYSIS

5.1 Hardware Requirements and Software Requirements

Developer requirement

- **Minimum Hardware Requirement**

RAM	4GB RAM
Processor	i3 11 gen
Graphics Card	2 GB

- **Software Requirement**

AWS CLI	Version 2

User requirement

Any compatible device to support internet connectivity and browser support to access the webpage for running the project online through internet.

- **Minimum Hardware requirement**

System	With internet connectivity

- **Software requirement**

Browser	Any

5.2 Implementation Details

Our methodology for implementing the solution involves a comprehensive approach, combining the capabilities of Amazon Lex for natural language understanding and AWS Lambda for backend processing. Below are the detailed steps:

1. **Amazon Lex configuration:**

- Intent definition: The first step in configuring Amazon Lex involves defining intents. Intents represent the various actions or goals users want to achieve through the chatbot. For ConCraft, we define multiple intents based on the types of user requests anticipated. One of the primary intents might be "RetrieveDataSheetIntent," which captures user requests for accessing meeting datasheets. Other possible intents could include "GetActionItemsIntent" for retrieving specific action items from meetings or "GetProjectUpdatesIntent" for summarizing recent project updates. Each intent is carefully crafted to address specific user needs and streamline their interaction with the chatbot.
- Slot configuration: Slots are used within each intent to capture specific pieces of information required to fulfill user requests. In the context of ConCraft, slots could include details such as the filename of the datasheet, the date of the meeting, or the type of information requested (e.g., agenda, action items, decisions). For example, in the "RetrieveDataSheetIntent," slots like "datasheetName" and "meetingDate" help the chatbot understand exactly which document the user is referring to. Proper slot configuration is critical to ensure that the chatbot can accurately gather all necessary information to process user queries effectively.
- Utterance training: Training the chatbot with a variety of utterances is essential to improve its ability to understand and interpret user queries accurately. This involves providing Amazon Lex with numerous examples of how users might phrase their requests. For instance, to train the "RetrieveDataSheetIntent," we might include utterances like "Show me the datasheet from the last meeting," "I need the datasheet for the project update," or "Get the minutes from yesterday's meeting." By exposing the chatbot to diverse phrasings, we enhance its ability to comprehend different ways users might express their needs, thus improving its responsiveness and accuracy.
- Iterative refinement: The process of configuring Amazon Lex is iterative. After the initial setup, continuous refinement is necessary based on user feedback and testing results. As users interact with the chatbot, we gather data on its performance, identifying areas where

it may misinterpret queries or fail to provide accurate responses. Adjustments to intents, slots, and utterances are made accordingly to improve the chatbot's accuracy and effectiveness over time. This iterative approach ensures that ConCraft evolves and adapts to better meet user needs.

2. **AWS Lambda development:**

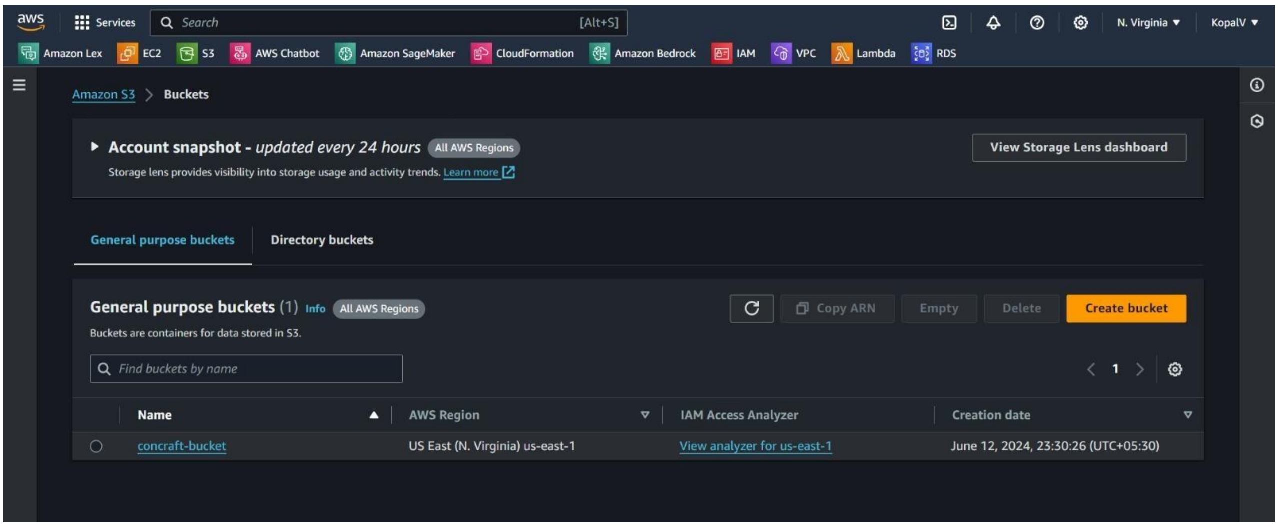
- Fulfillment logic: AWS Lambda functions serve as the backend for processing user requests. When ConCraft receives a query, the Lambda function is triggered to handle the request. The fulfillment logic involves retrieving the requested datasheet from a designated Amazon S3 bucket based on the provided filename. For example, when a user asks for a specific datasheet, the Lambda function uses the filename captured in the slot to locate and retrieve the file from S3. This backend processing is crucial for fetching the correct document and preparing it for further processing.
- Data sheet processing: Once the datasheet is retrieved, the Lambda function processes it to extract relevant information. This step may involve parsing the datasheet to identify key sections such as agendas, action items, decisions, and updates. Advanced data processing techniques, including natural language processing (NLP) and machine learning algorithms, can be employed to extract insights, summarize information, or calculate relevant statistics. For instance, if a user requests action items, the Lambda function can parse the datasheet to find and compile all listed action items, ensuring that the response is precise and comprehensive.
- Result formatting: The processed results are then formatted in a clear and concise manner suitable for presentation to the user. This ensures that users receive relevant and actionable information in response to their queries. For example, if the user requested action items from a meeting, the response should be a well-organized list of tasks with associated details such as deadlines and responsible parties. Proper result formatting enhances the user experience by making the retrieved information easy to read and understand.
- Error handling: Robust error handling mechanisms are implemented within the Lambda function to gracefully manage exceptions and edge cases. This includes handling scenarios such as invalid user inputs, failures in accessing the S3 bucket, or errors during data processing. For instance, if the requested datasheet is not found, the chatbot should provide a helpful error message guiding the user on how to proceed. Effective error handling ensures that the system remains reliable and user-friendly.

3. **Monitoring and Optimization:**

- Continuous monitoring: Continuous monitoring is essential to ensure the optimal performance of both the chatbot and the Lambda function. Using AWS CloudWatch metrics and logs, we track the performance of ConCraft, identifying any bottlenecks or issues that may arise. Monitoring helps in detecting problems early and provides insights into the system's behavior under different loads. Regular analysis of these metrics allows us to make informed decisions about necessary optimizations and adjustments.
- Scalability considerations: Scalability is a critical aspect of ConCraft's design. The system must be able to handle varying loads and scale seamlessly to accommodate increased usage. Implementing auto-scaling mechanisms ensures that resources are dynamically allocated based on demand, maintaining performance during peak times. Optimizing resource allocation involves fine-tuning the Lambda function and other components to ensure they can efficiently handle large numbers of concurrent requests without degradation in service quality.
- Feedback incorporation: User feedback and analytics data play a vital role in the iterative refinement process. By analyzing user interactions and feedback, we gain valuable insights into how the chatbot is being used and where improvements can be made. This feedback loop allows us to continuously enhance ConCraft's accuracy, responsiveness, and overall user experience. For example, if users frequently ask questions that the chatbot struggles to answer, we can refine the training data and adjust the configuration to better handle those queries.

a.) Screenshots of project:

1. In the following figure 5.1 we have created this bucket to upload meeting file in s3.



▸ Account snapshot - *updated every 24 hours* All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Learn more

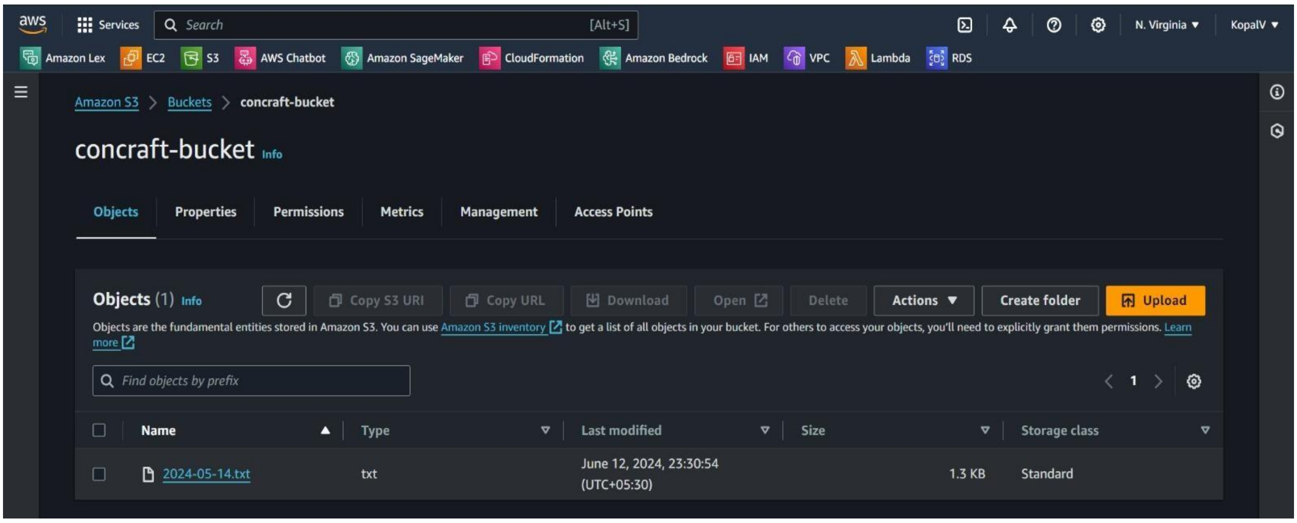
General purpose buckets (1) Info All AWS Regions

Buckets are containers for data stored in S3.

	Name ▲	AWS Region ▽	IAM Access Analyzer	Creation date ▽
○	concraft-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	June 12, 2024, 23:30:26 (UTC+05:30)

Fig 5.1: ConCraft bucket

2. In the following figure 5.2 we have uploaded the file object in the bucket.



concraft-bucket Info

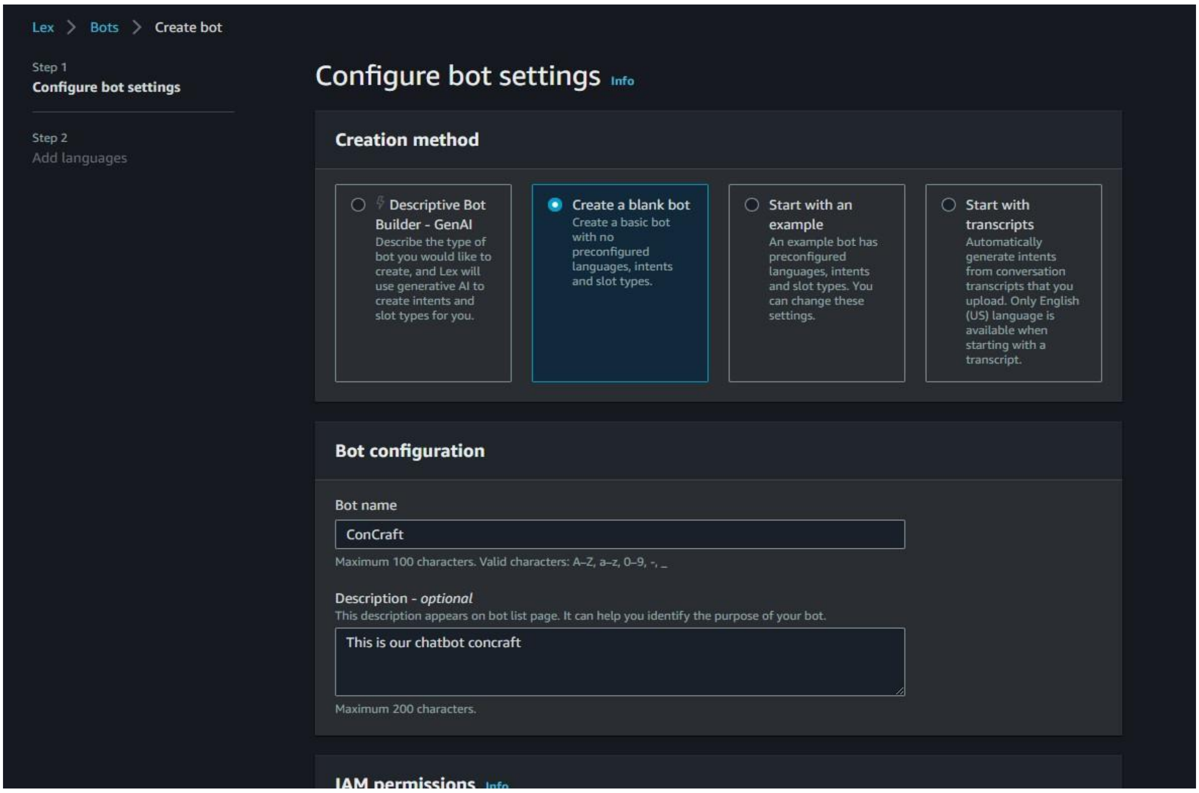
Objects (1) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	2024-05-14.txt	txt	June 12, 2024, 23:30:54 (UTC+05:30)	1.3 KB	Standard

Fig 5.2: ConCraft file object

3. In the following figure 5.3 we have created the ConCraft bot.



Configure bot settings

Creation method

○ Descriptive Bot Builder - GenAI
Describe the type of bot you would like to create, and Lex will use generative AI to create intents and slot types for you.

● Create a blank bot
Create a basic bot with no preconfigured languages, intents and slot types.

○ Start with an example
An example bot has preconfigured languages, intents and slot types. You can change these settings.

○ Start with transcripts
Automatically generate intents from conversation transcripts that you upload. Only English (US) language is available when starting with a transcript.

Bot configuration

Bot name: ConCraft

Maximum 100 characters. Valid characters: A–Z, a–z, 0–9, -, _

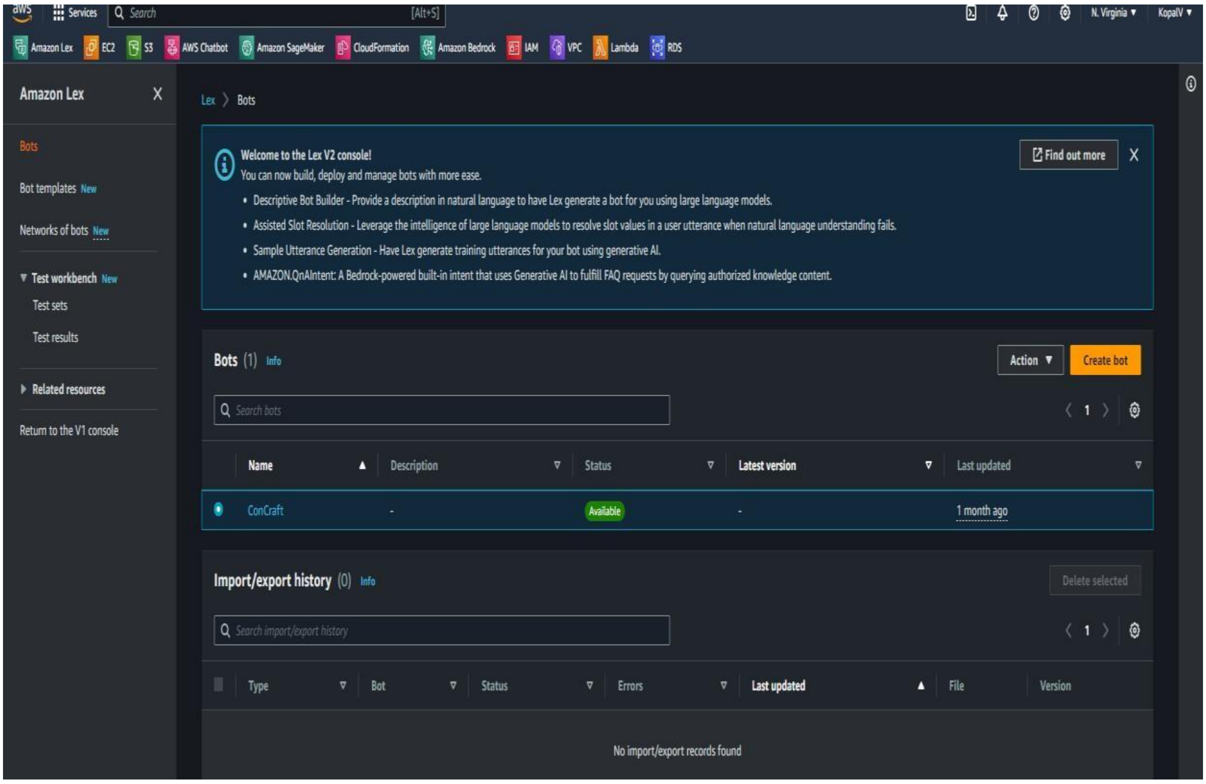
Description - *optional*
This description appears on bot list page. It can help you identify the purpose of your bot.

This is our chatbot concraft

Maximum 200 characters.

Fig 5.3: Creating ConCraft bot

4. In the following figure 5.4 we have imported the bot using compressed file.



Welcome to the Lex V2 console!
You can now build, deploy and manage bots with more ease.

- Descriptive Bot Builder - Provide a description in natural language to have Lex generate a bot for you using large language models.
- Assisted Slot Resolution - Leverage the intelligence of large language models to resolve slot values in a user utterance when natural language understanding fails.
- Sample Utterance Generation - Have Lex generate training utterances for your bot using generative AI.
- AMAZON.QnAIntent: A Bedrock-powered built-in intent that uses Generative AI to fulfill FAQ requests by querying authorized knowledge content.

Bots (1)

	Name ▲	Description	Status	Latest version	Last updated
●	ConCraft	-	Available	-	1 month ago

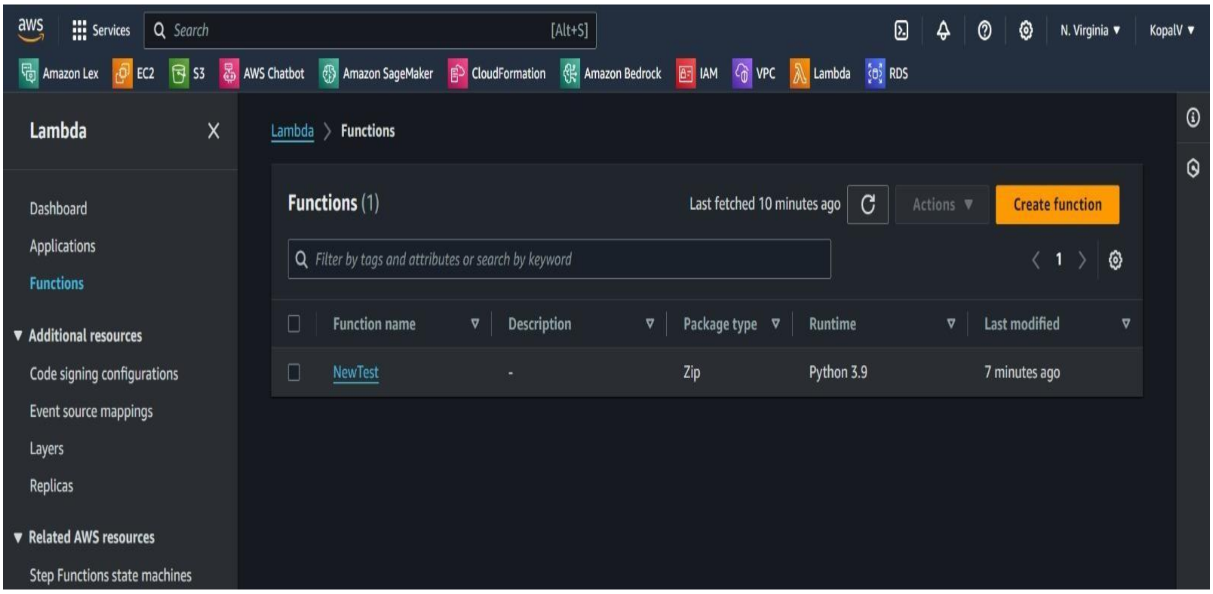
Import/export history (0)

	Type	Bot	Status	Errors	Last updated ▲	File	Version

No import/export records found

Fig 5.4: Successfully created Concraft bot

5. In the following figure 5.5 we have successfully created the lambda function.

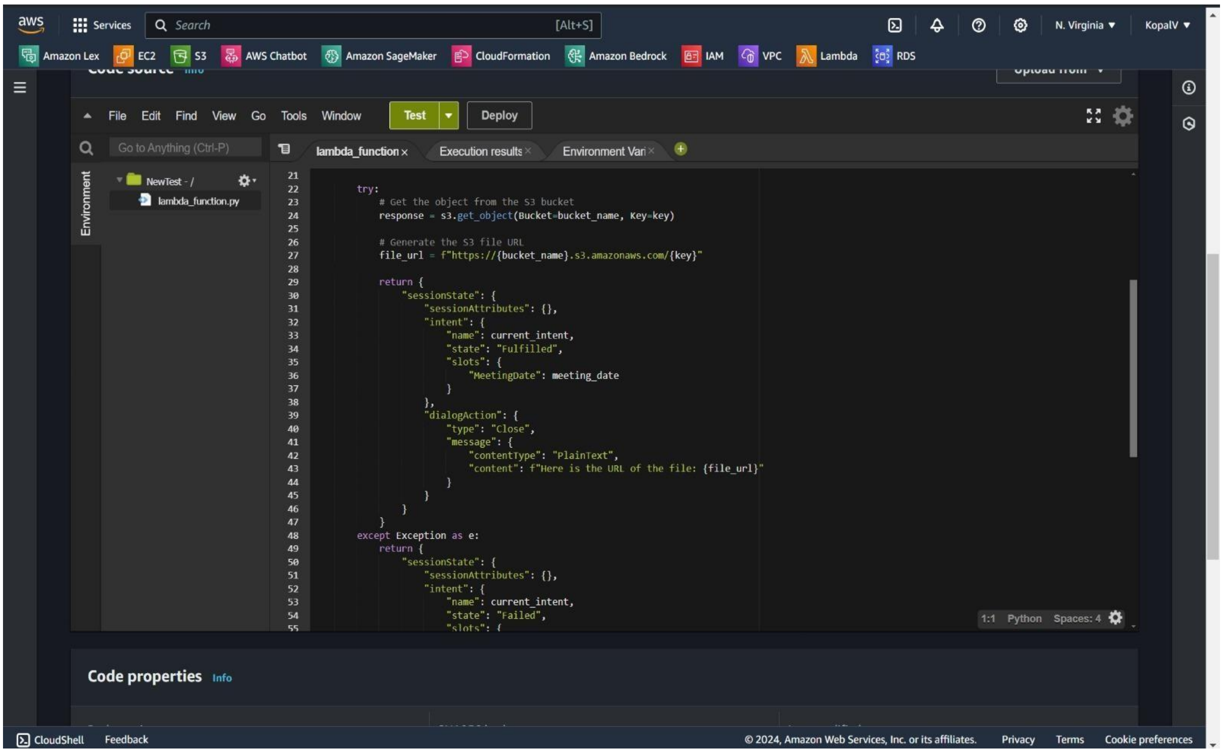


Functions (1)

□	Function name	Description	Package type	Runtime	Last modified
□	NewTest	-	Zip	Python 3.9	7 minutes ago

Fig 5.5: Successfully created Lambda function

6. In the following figure 5.6 we have coded the lambda function for the chatbot



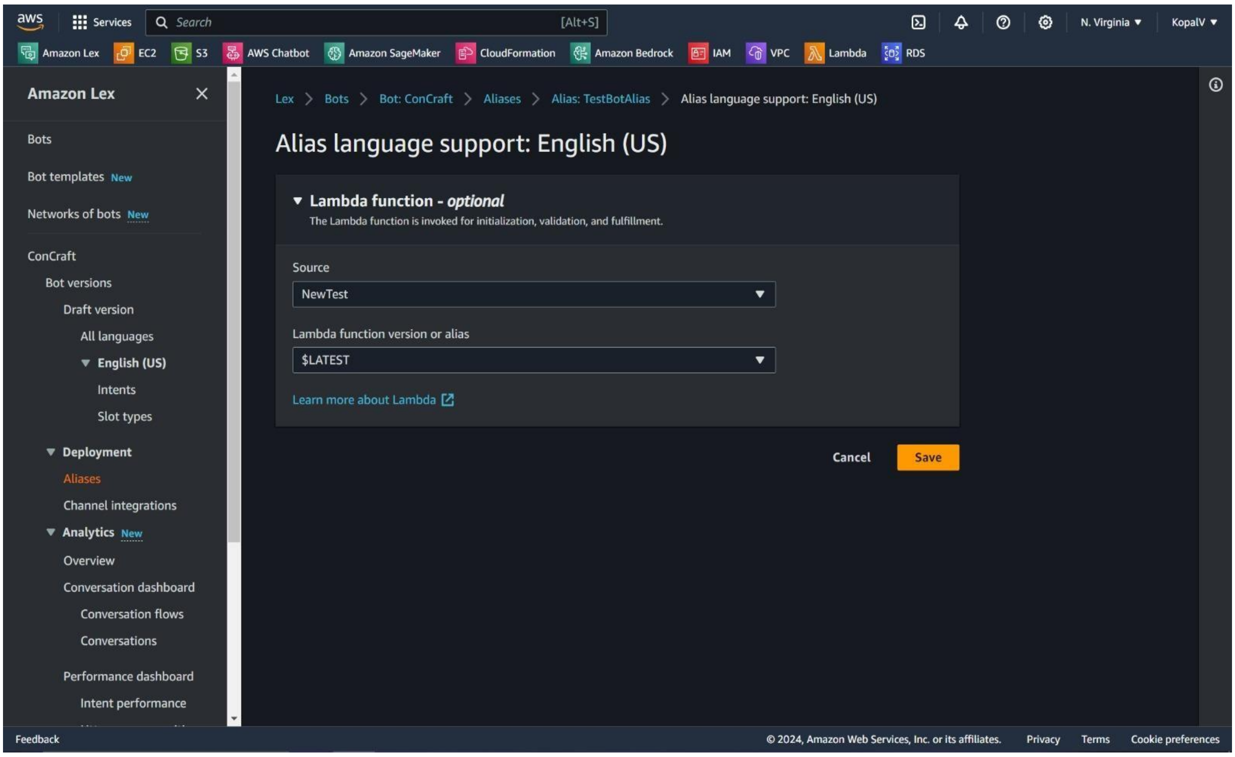
```
    try:
        # Get the object from the S3 bucket
        response = s3.get_object(Bucket=bucket_name, Key=key)

        # Generate the S3 file URL
        file_url = f"https://{bucket_name}.s3.amazonaws.com/{key}"

        return {
            "sessionState": {
                "sessionAttributes": {},
                "intent": {
                    "name": current_intent,
                    "state": "Fulfilled",
                    "slots": {
                        "MeetingDate": meeting_date
                    }
                },
                "dialogAction": {
                    "type": "Close",
                    "message": {
                        "contentType": "PlainText",
                        "content": f"Here is the URL of the file: {file_url}"
                    }
                }
            }
        }
    except Exception as e:
        return {
            "sessionState": {
                "sessionAttributes": {},
                "intent": {
                    "name": current_intent,
                    "state": "Failed",
                    "slots": {
```

Fig 5.6: Lambda function

7. In the following figure 5.7 we have added the function in the bot alias.



Alias language support: English (US)

Lambda function - *optional*

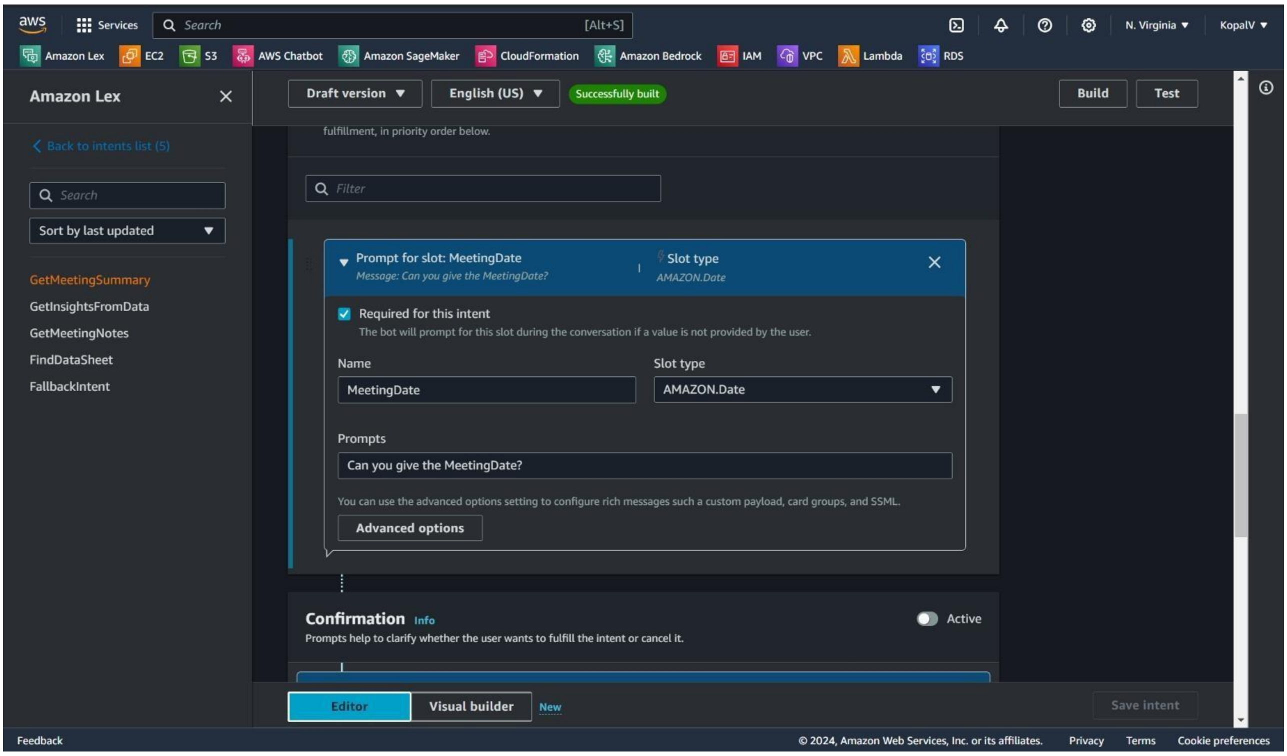
The Lambda function is invoked for initialization, validation, and fulfillment.

Source: NewTest

Lambda function version or alias: $LATEST

Fig 5.7: Configuration of function with Lex

8. In the following figure 5.8 we have added the Meeting Date as the slot



fulfillment, in priority order below.

Prompt for slot: MeetingDate

Message: Can you give the MeetingDate?

Slot type

AMAZON.Date

☑ Required for this intent

The bot will prompt for this slot during the conversation if a value is not provided by the user.

Name: MeetingDate

Slot type: AMAZON.Date

Prompts: Can you give the MeetingDate?

You can use the advanced options setting to configure rich messages such a custom payload, card groups, and SSML.

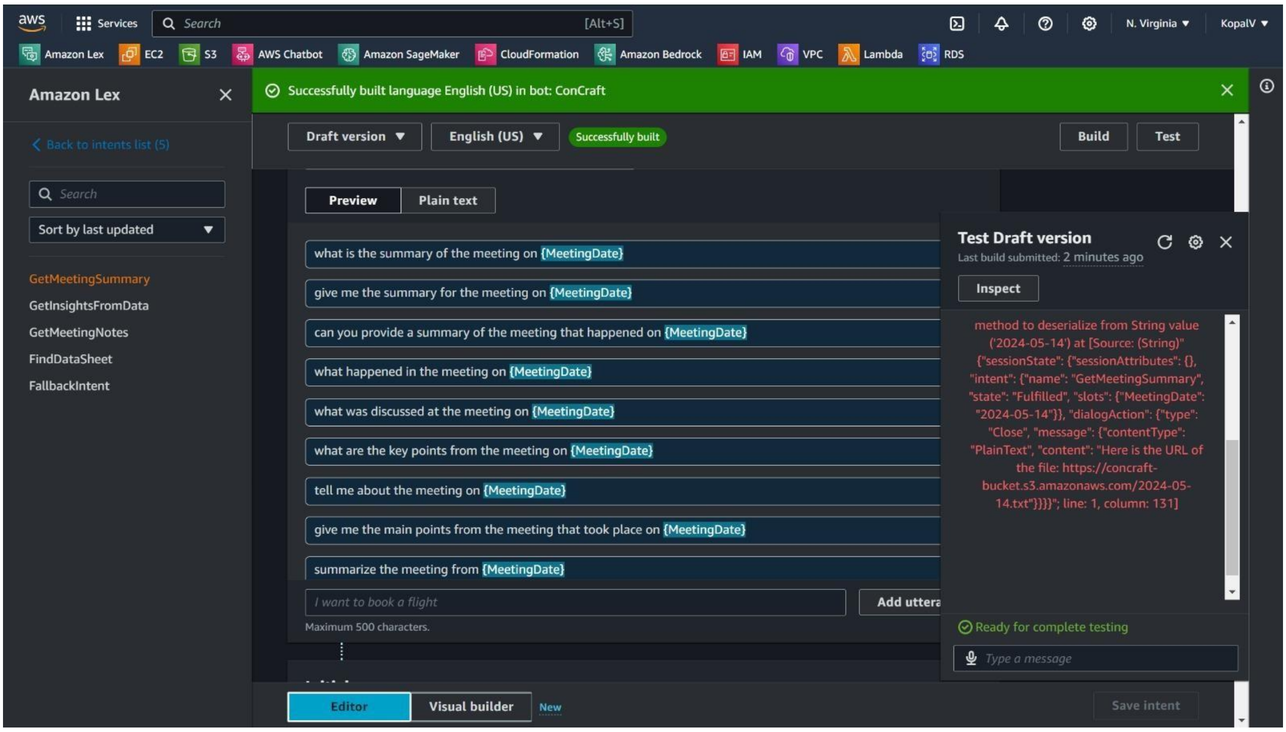
Confirmation Info

Active

Prompts help to clarify whether the user wants to fulfill the intent or cancel it.

Fig 5.8: Lex configuration

9. In the following figure 5.9 we have successfully tested the URL response



Successfully built language English (US) in bot: ConCraft

what is the summary of the meeting on {MeetingDate}

give me the summary for the meeting on {MeetingDate}

can you provide a summary of the meeting that happened on {MeetingDate}

what happened in the meeting on {MeetingDate}

what was discussed at the meeting on {MeetingDate}

what are the key points from the meeting on {MeetingDate}

tell me about the meeting on {MeetingDate}

give me the main points from the meeting that took place on {MeetingDate}

summarize the meeting from {MeetingDate}

I want to book a flight

Maximum 500 characters.

Test Draft version

Last build submitted: 2 minutes ago

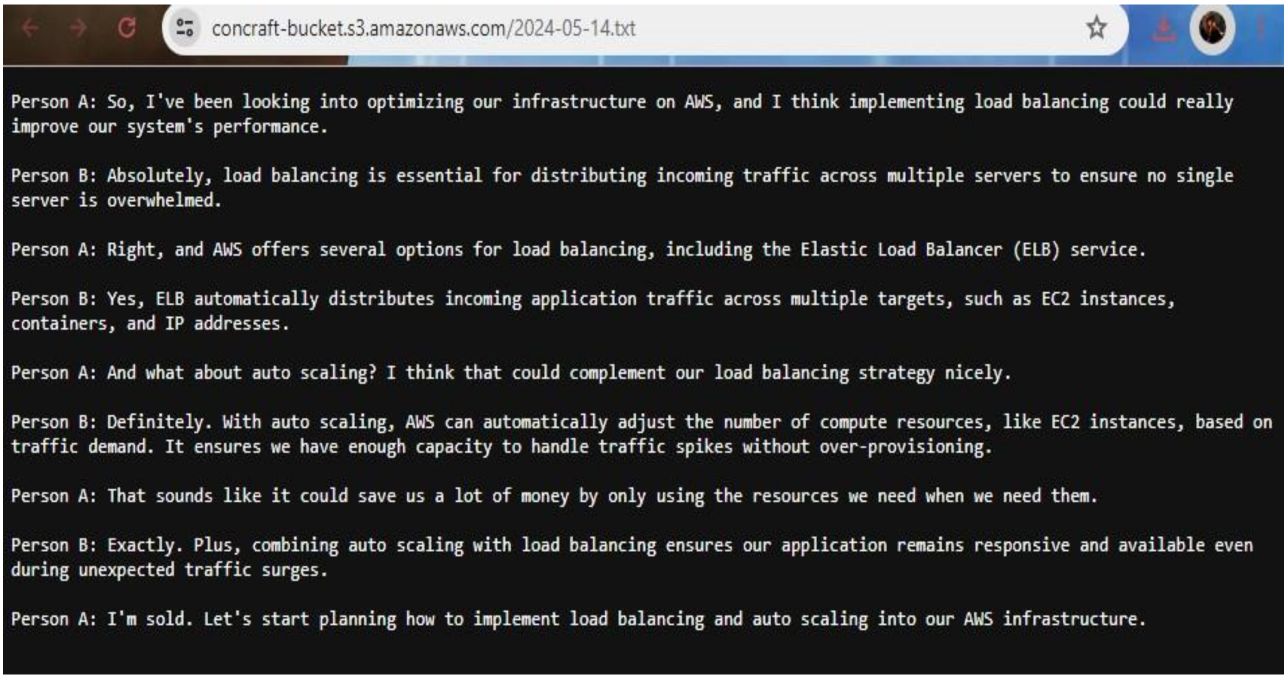
```
method to deserialize from String value ('2024-05-14') at [Source: (String)" {"sessionState": {"sessionAttributes": {}, "intent": {"name": "GetMeetingSummary", "state": "Fulfilled", "slots": {"MeetingDate": "2024-05-14"}}, "dialogAction": {"type": "Close", "message": {"contentType": "PlainText", "content": "Here is the URL of the file: https://concraft-bucket.s3.amazonaws.com/2024-05-14.txt"}}}}"; line: 1, column: 131]
```

Ready for complete testing

Type a message

Fig 5.9: Generating output with file link

10. In the following figure 5.10 we have successfully navigated the URL



```
Person A: So, I've been looking into optimizing our infrastructure on AWS, and I think implementing load balancing could really
improve our system's performance.

Person B: Absolutely, load balancing is essential for distributing incoming traffic across multiple servers to ensure no single
server is overwhelmed.

Person A: Right, and AWS offers several options for load balancing, including the Elastic Load Balancer (ELB) service.

Person B: Yes, ELB automatically distributes incoming application traffic across multiple targets, such as EC2 instances,
containers, and IP addresses.

Person A: And what about auto scaling? I think that could complement our load balancing strategy nicely.

Person B: Definitely. With auto scaling, AWS can automatically adjust the number of compute resources, like EC2 instances, based on
traffic demand. It ensures we have enough capacity to handle traffic spikes without over-provisioning.

Person A: That sounds like it could save us a lot of money by only using the resources we need when we need them.

Person B: Exactly. Plus, combining auto scaling with load balancing ensures our application remains responsive and available even
during unexpected traffic surges.

Person A: I'm sold. Let's start planning how to implement load balancing and auto scaling into our AWS infrastructure.
```

Fig 5.10: Accessing file stored in S3 through Lex

Cost Estimation Chart for ConCraft Project Using AWS Services

The ConCraft project leverages Amazon Lex for building an intelligent chatbot to assist users with retrieving meeting datasheets, action items, and project updates. Given the estimated monthly cost of $1.48 and a total annual cost of $17.76, it is essential to explore cost optimization strategies to ensure the project remains cost-effective while delivering high performance.

1. Current Cost Estimate Breakdown Based on the AWS Pricing Calculator:

- Monthly Cost: $1.48
- Annual Cost: $17.76
- Services Used: Amazon Lex
- Region: US East (N. Virginia)

2. Configurations:

- Number of speech requests: 100
- Number of text requests: 100
- Number of speech intervals: 2
- Number of text requests (Streaming Conversation): 2
- Number of training minutes (Automated Chatbot): 2
-

Cost Optimization Strategies

1. **Optimizing Amazon Lex usage:** Amazon Lex costs are influenced by the number of requests, speech intervals, and training minutes. Here are some strategies to optimize these costs:

   Reduce the number of requests:

   Aggregate Data: Instead of multiple requests for small pieces of data, aggregate requests to reduce the total number of API calls.

   Caching Mechanism: Implement a caching mechanism to store frequently accessed data, reducing repeated requests.

- Optimize speech intervals:

  Batch Processing: Group speech interactions to minimize the number of speech intervals. Efficient Interaction Design: Design the chatbot interaction to be as concise as possible, minimizing the duration of each speech interaction.

- Optimize training minutes:

  Incremental Training: Instead of retraining the model frequently, use incremental training methods to update the model only with new data.

Scheduled Training: Schedule training during off-peak hours to potentially benefit from lower costs or available promotional credits.

2. **Utilize AWS free tier**

- AWS Free Tier for Amazon Lex: Make full use of the AWS Free Tier, which offers up to 10,000 text requests and 5,000 speech requests per month for the first year.

3. **Monitor and analyze usage**

- AWS CloudWatch: Use AWS CloudWatch to monitor API usage and set up alarms for unusual spikes in usage that could lead to higher costs.
- Cost Explorer: Regularly review costs and usage patterns using AWS Cost Explorer to identify opportunities for optimization.

4. **Leverage reserved instances and savings plans**

- Savings Plans: Consider AWS Compute Savings Plans for any Lambda functions used in conjunction with Amazon Lex, which can provide significant cost savings over on-demand pricing.

5. **Efficient Lambda function execution:** Ensure that AWS Lambda functions are optimized for performance and cost by:

- Minimizing the function's execution time.
- Reducing the amount of memory allocated if not necessary.
- Reusing Lambda execution contexts where possible.
- Use provisioned concurrency for Lambda functions that require consistent start times, balancing the cost with the need for performance.

6. **Integration with other AWS services:** Use S3 Storage Cost Management: For storing and retrieving meeting datasheets:

- Use S3 lifecycle Policies to transition data to less expensive storage classes (e.g., S3 Standard-IA or S3 Glacier) when appropriate.
- Enable S3 intelligent-tiering for automatic cost savings based on access patterns.

7. **Implement cost allocation tags**

- Cost Allocation Tags: Tag AWS resources with meaningful identifiers (project, environment, team) to gain better visibility into cost drivers and identify areas for cost savings.

8. **Review and right-size services regularly**

- Regular audits: Conduct regular audits of service usage and configurations to ensure that they are aligned with current needs and not over-provisioned. Automation and Efficient Workflow Design

9. **Automation**: Automate repetitive tasks and workflows to reduce manual intervention and potential errors that could lead to increased costs.

- Serverless Architecture: Continue leveraging serverless architecture to avoid costs associated with idle resources.

10. Cost optimization example scenarios

    Scenario 1: Reducing the number of requests

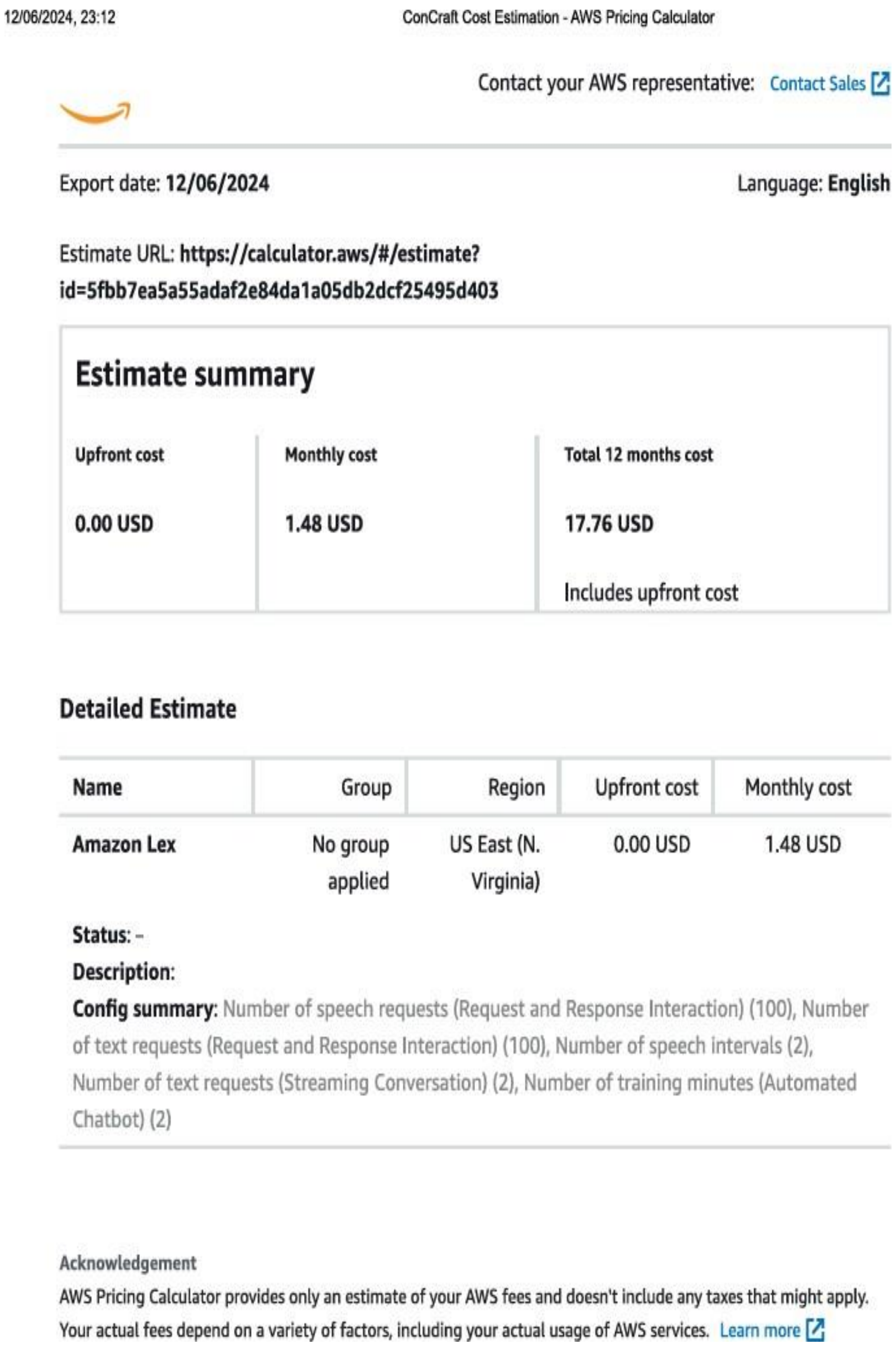
- Before optimization: 100 speech requests, 100 text requests.
- After optimization: 50 speech requests (through aggregation), 75 text requests (through caching).

  Scenario 2: Optimizing Lambda Function Execution

- Before optimization: 256 MB memory, 5 seconds execution time.
- After optimization: 128 MB memory, 3 seconds execution time.and features effectively are key to optimizing costs for the ConCraft project.

Cost Estimation Chart

Optimization area	**Current monthely cost**	**Optimized monthly cost**	**Annual savings**
Amazon Lex Requests	$1.48	$1.00	$5.76
AWS Free Tier Utilization	-	-	$0.00
Lambda Function Optimization	-	-	$0.00
S3 Storage Management	-	-	$0.00
Reserved Instances/Savings Plans	-	-	$0.00
Total Estimated Savings	$1.48	$1.00	$5.76



12/06/2024, 23:12 ConCraft Cost Estimation - AWS Pricing Calculator

Contact your AWS representative: Contact Sales

Export date: **12/06/2024**

Language: **English**

Estimate URL: **https://calculator.aws/#/estimate?id=5fbb7ea5a55adaf2e84da1a05db2dcf25495d403**

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost
0.00 USD	1.48 USD	17.76 USD Includes upfront cost

Detailed Estimate

Name	Group	Region	Upfront cost	Monthly cost
Amazon Lex	No group applied	US East (N. Virginia)	0.00 USD	1.48 USD

Status: -

Description:

Config summary: Number of speech requests (Request and Response Interaction) (100), Number of text requests (Request and Response Interaction) (100), Number of speech intervals (2), Number of text requests (Streaming Conversation) (2), Number of training minutes (Automated Chatbot) (2)

Acknowledgement

AWS Pricing Calculator provides only an estimate of your AWS fees and doesn't include any taxes that might apply. Your actual fees depend on a variety of factors, including your actual usage of AWS services. Learn more

Fig 5.11: Cost Estismation

5.3 Project Scheduling using GANTT chart

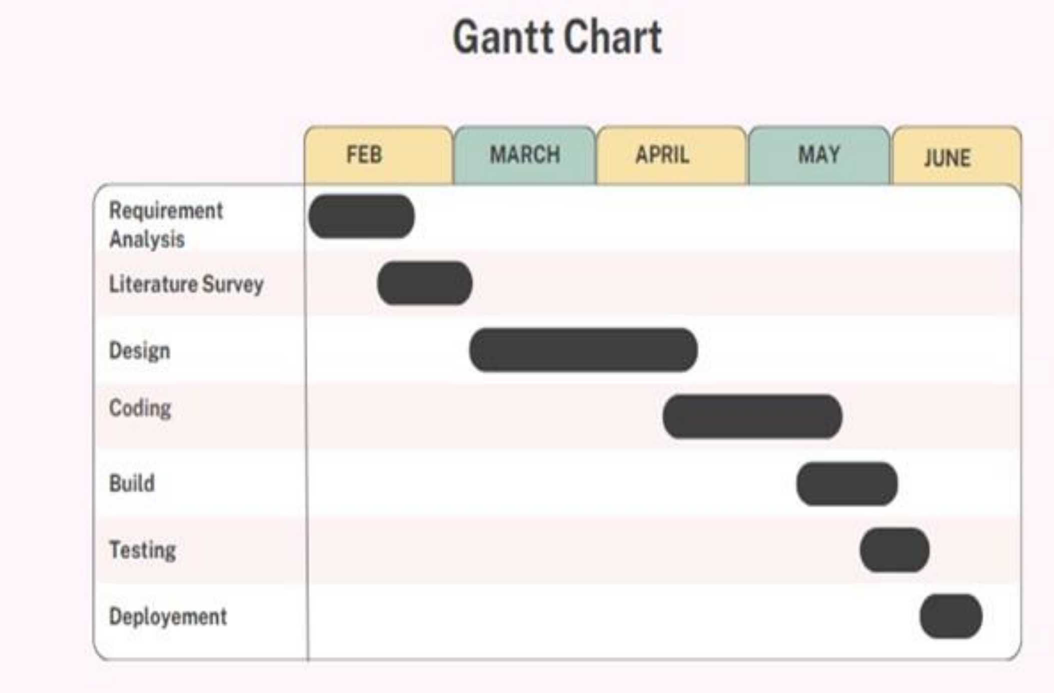


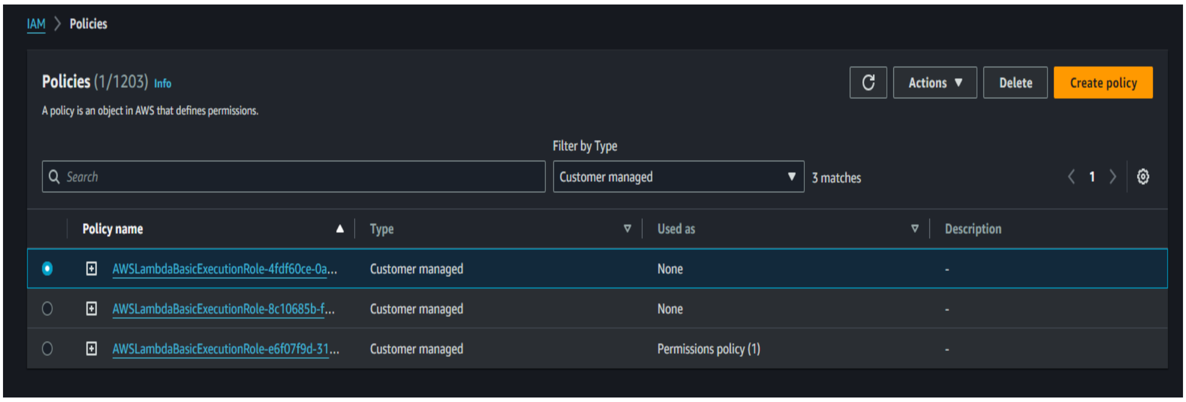
Fig 5.12: Gantt chart shows the project timeline

5.4 Software Testing

a) Test Case: Integration testing (by using lex service)

Integration testing is crucial for ensuring that the various components of the ConCraft project work seamlessly together. For ConCraft, this involves testing the interactions between Amazon Lex, AWS S3, AWS IAM, and AWS Lambda. Each of these services plays a vital role in the overall functionality of the chatbot, and integration testing helps identify any issues that might arise when these components interact

Result



Policies (1/1203) Info

A policy is an object in AWS that defines permissions.

Filter by Type: Customer managed ▼ 3 matches

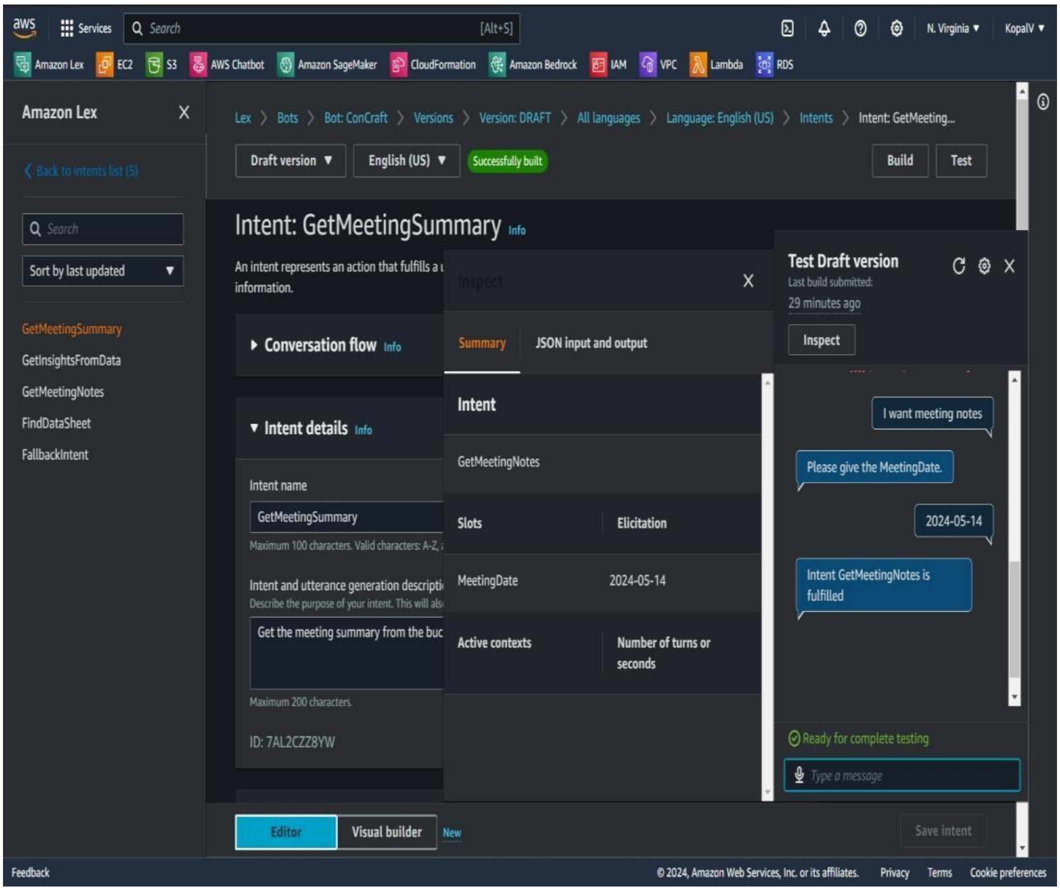
	Policy name	Type	Used as	Description
●	AWSLambdaBasicExecutionRole-4fdf60ce-0a...	Customer managed	None	-
○	AWSLambdaBasicExecutionRole-8c10685b-f...	Customer managed	None	-
○	AWSLambdaBasicExecutionRole-e6f07f9d-31...	Customer managed	Permissions policy (1)	-

Fig 5.13: Testing of chatbot

b) Test Case: Intent testing (by using lex service)

Intent testing ensures that the chatbot accurately understands and processes user requests corresponding to specific intents. For ConCraft, two primary intents are crucial: "GetMeetingSummaryIntent" and "GetMeetingNotesIntent." This testing will involve verifying that the chatbot correctly identifies user intents, gathers necessary information through slots, and provides appropriate responses.

Result



Lex > Bots > Bot: ConCraft > Versions > Version: DRAFT > All languages > Language: English (US) > Intents > Intent: GetMeeting...

Draft version ▼ English (US) ▼ Successfully built

Intent: GetMeetingSummary

An intent represents an action that fulfills a u... information.

Intent name: GetMeetingSummary

Maximum 100 characters. Valid characters: A-Z, a...

Intent and utterance generation descripti...

Describe the purpose of your intent. This will als...

Get the meeting summary from the buc...

Maximum 200 characters.

ID: 7AL2CZZ8YW

Intent	
GetMeetingNotes	
Slots	Elicitation
MeetingDate	2024-05-14
Active contexts	Number of turns or seconds

Test Draft version

Last build submitted: 29 minutes ago

I want meeting notes

Please give the MeetingDate.

2024-05-14

Intent GetMeetingNotes is fulfilled

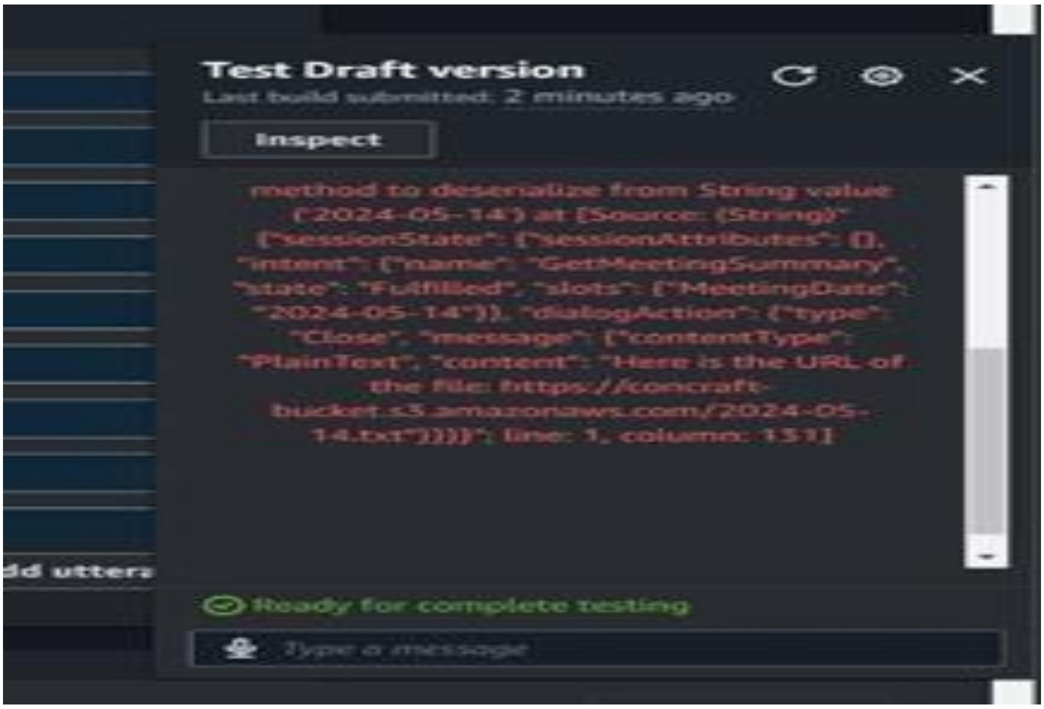
Ready for complete testing

Fig 5.14: Intent fullfilled

c) **Test Case: Lambda function invocation (by using lex service)**

Testing the invocation of AWS Lambda functions by the Amazon Lex service is crucial to ensure that the backend processing for various intents is executed correctly. The Lambda function acts as the backend engine that processes the user's request and provides the necessary response.

Result


Test Draft version
Last build submitted: 2 minutes ago
Inspect
dd utter
Ready for complete testing
Type a message

```
method to deserialize from String value
('2024-05-14') at [Source: (String)"
{"sessionState": {"sessionAttributes": {},
"intent": {"name": "GetMeetingSummary",
"state": "Fulfilled", "slots": {"MeetingDate":
"2024-05-14"}}, "dialogAction": {"type":
"Close", "message": {"contentType":
"PlainText", "content": "Here is the URL of
the file: https://concraft-
bucket.s3.amazonaws.com/2024-05-
14.txt"}}}}"; line: 1, column: 151]
```

Fig 5.15: NewTest function testing

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 Conclusion

Throughout the course of our project, ConCraft has proven to be an invaluable tool in capturing and delivering essential meeting details, thereby enabling team members who were unable to attend to stay informed and engaged. This capability underscores the importance of leveraging advanced technologies like Amazon Lex and AWS Lambda to enhance workplace communication and productivity. The success of ConCraft highlights the transformative potential of intelligent chatbots in creating more inclusive and efficient work environments. By bridging the gap between team members who are present in meetings and those who are not, ConCraft fosters a culture of transparency, accountability, and seamless collaboration.

ConCraft has not only demonstrated its ability to retrieve and process critical information from meeting datasheets efficiently, but it has also showcased the significance of accessibility and usability in modern workplace tools. The intuitive design and robust functionality of ConCraft ensure that all team members, regardless of their location or schedule, can access the information they need to contribute effectively. This democratization of information is crucial in today's fast-paced work environments, where the ability to make informed decisions swiftly can significantly impact overall team performance and project outcomes.

Moreover, ConCraft's success illustrates the broader trend of integrating artificial intelligence into everyday business operations. Intelligent chatbots like ConCraft can handle repetitive tasks, provide quick access to information, and facilitate better decision-making processes, thereby freeing up valuable time for team members to focus on more strategic activities. This shift not only enhances individual productivity but also drives collective team success by ensuring that everyone is on the same page.

In summary, ConCraft is more than a project; it is a transformative solution that empowers team members to contribute their best, fosters seamless collaboration, and ensures that no crucial information is ever missed. With ConCraft, we bid farewell to the fear of missing out and welcome a new era of connected and informed teamwork. The comprehensive integration with existing systems, user-friendly interface, and continuous monitoring for improvements make ConCraft a robust and reliable tool for any organization. As we conclude this project, we are confident that ConCraft will continue to serve as a valuable asset for teams, helping them stay connected and informed, regardless of their physical presence in meetings. Its future potential, including features like multilingual support, voice-activated commands, and integration with additional platforms and project management tools, promises to further enhance its utility and relevance in an ever-evolving workplace landscape.

6.2 Future scope

The future development of ConCraft aims to enhance its functionality, usability, and integration capabilities to better serve global teams and diverse linguistic backgrounds. Below are detailed enhancements planned for ConCraft:

- **Multilingual support:** To cater to global teams and diverse linguistic backgrounds, we will introduce multilingual support in ConCraft. This feature will enable ConCraft to understand and respond in multiple languages, making it a versatile tool for international organizations. By implementing natural language processing (NLP) models capable of handling various languages, ConCraft will provide seamless interactions regardless of the user’s native tongue. This enhancement will include automatic language detection, allowing the chatbot to switch languages based on the user’s input. For example, a team member in France could interact with ConCraft in French, while a colleague in Japan could use Japanese. This capability not only enhances accessibility but also ensures that language barriers do not impede the flow of crucial information within global teams. Additionally, multilingual support will facilitate smoother collaboration in multinational projects, promoting inclusivity and ensuring that all team members are equally informed and engaged.

- **Voice-activated commands:** Implementing voice-activated commands will significantly enhance the usability of ConCraft. Voice interactions will allow users to engage with the chatbot using verbal instructions, making it particularly useful in hands-free environments or for users with accessibility needs. This feature will leverage advanced speech recognition technologies to accurately capture and interpret spoken queries. For instance, a user could simply say, "Hey ConCraft, show me the latest project updates," and receive the information without needing to type. This functionality is ideal for on-the-go scenarios, such as during meetings, driving, or when users are otherwise occupied with tasks that require their hands. By providing a more natural and convenient way to interact with ConCraft, voice-activated commands will improve user experience and efficiency. This enhancement will be particularly beneficial in environments where typing is impractical or for users who find voice commands more intuitive and faster.

- **Integration with additional platforms:** Expanding ConCraft's integration capabilities beyond Amazon Lex and AWS Lambda is a strategic priority. By integrating with popular collaboration tools like Slack, Microsoft Teams, and Zoom, we aim to make ConCraft accessible from users' preferred communication platforms. This will allow team members to access meeting information directly within the environments they already use for their daily workflows. For example, a user could ask ConCraft for the latest action items while in a Zoom meeting or check project updates in a Slack channel. These integrations will streamline the process of retrieving information and reduce the need to switch between different applications, thereby enhancing productivity and user convenience. Moreover, such integrations will enable ConCraft to deliver notifications and updates proactively within these platforms, ensuring that team members are always informed about the latest developments. By being embedded in commonly used tools, ConCraft will become a seamless part of the team's daily routine, enhancing its utility and relevance.

- **Integration with project management tools:** Integrating ConCraft with project management tools like Jira, Trello, and Asana will provide a comprehensive solution for tracking action items and project updates. This integration will streamline workflow by ensuring that all relevant information is consolidated in one place. For example, when action items are documented in ConCraft during a meeting, they can automatically be synced with the corresponding tasks in Jira or Trello. This seamless integration will eliminate the need for manual data entry, reduce the risk of oversight, and ensure that all team members have access to up-to-date project information. Additionally, ConCraft could fetch and summarize status reports from these project management tools, providing team members with a holistic view of project progress and outstanding tasks during meetings or on request. By bridging the gap between meeting discussions and project execution, these integrations will enhance collaboration, accountability, and project management efficiency.

- **Advanced analytics and insights:** To further empower teams, ConCraft will evolve to provide advanced analytics and insights. Leveraging machine learning and data analytics, ConCraft will analyze meeting data to identify patterns, trends, and key performance indicators. For example, it could track the frequency of certain topics, monitor the progress of action items over time, and highlight potential bottlenecks or areas of concern. These insights will be presented through intuitive dashboards and visualizations, enabling team leaders and members to make informed decisions based on data-driven insights. This feature will not only help in retrospective analysis but also in proactive planning and strategy formulation, ensuring that teams can continuously improve their processes and outcomes. For instance, teams could identify recurring challenges and address them systematically, leading to enhanced efficiency and effectiveness over time.

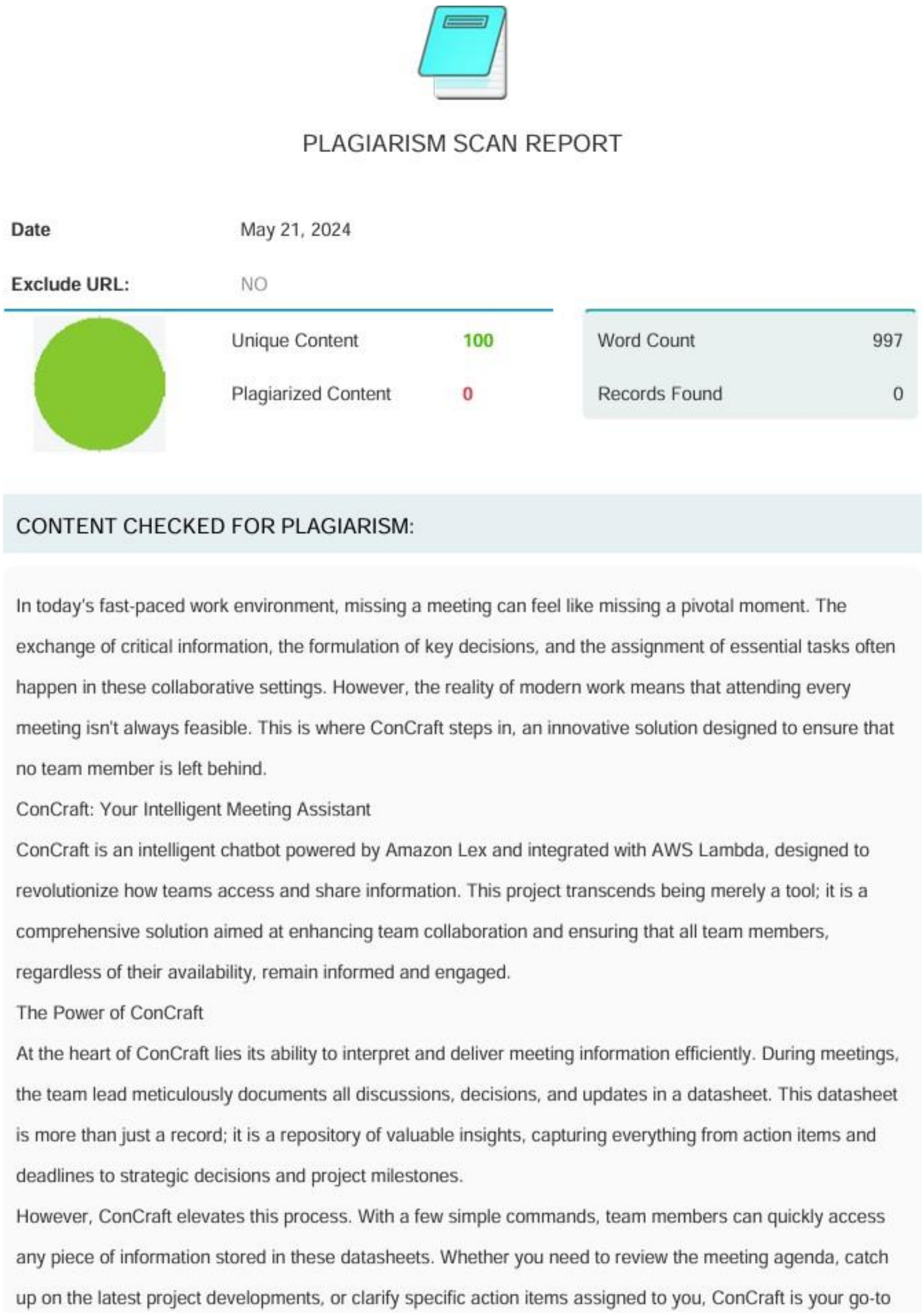
- **Customizable workflows and automation:** To accommodate the unique needs of different organizations, ConCraft will offer customizable workflows and automation capabilities. Users will be able to define custom triggers and actions within the chatbot's framework. For instance, an organization could set up a workflow where specific types of meeting notes automatically generate follow-up reminders or notifications to relevant stakeholders. Automation will also extend to routine tasks such as summarizing meeting minutes, distributing updates, and tracking deadlines. By automating these repetitive tasks, ConCraft will save time and reduce administrative overhead, allowing team members to focus on more strategic activities. Customizable workflows will also enable teams to tailor ConCraft's functionality to their specific operational processes, thereby maximizing its utility and relevance. This adaptability ensures that ConCraft can evolve alongside the organization's changing needs, maintaining its effectiveness and value over time.

REFERENCES

WEB REFERENCES:

1. **https://www.livechat.com/marketplace/apps/chatbot/#description**
2. **https://www.atlassian.com/software/jira**
3. **https://www.ibm.com/products/watsonx-assistant**
4. **https://www.haptik.ai/**
5. **https://docs.aws.amazon.com/**
6. **https://calculator.aws/#/**

Appendix –I (Plagiarism Report)



PLAGIARISM SCAN REPORT

Date	May 21, 2024
Exclude URL:	NO

Unique Content	100
Plagiarized Content	0

Word Count	997
Records Found	0

CONTENT CHECKED FOR PLAGIARISM:

In today's fast-paced work environment, missing a meeting can feel like missing a pivotal moment. The exchange of critical information, the formulation of key decisions, and the assignment of essential tasks often happen in these collaborative settings. However, the reality of modern work means that attending every meeting isn't always feasible. This is where ConCraft steps in, an innovative solution designed to ensure that no team member is left behind.

ConCraft: Your Intelligent Meeting Assistant

ConCraft is an intelligent chatbot powered by Amazon Lex and integrated with AWS Lambda, designed to revolutionize how teams access and share information. This project transcends being merely a tool; it is a comprehensive solution aimed at enhancing team collaboration and ensuring that all team members, regardless of their availability, remain informed and engaged.

The Power of ConCraft

At the heart of ConCraft lies its ability to interpret and deliver meeting information efficiently. During meetings, the team lead meticulously documents all discussions, decisions, and updates in a datasheet. This datasheet is more than just a record; it is a repository of valuable insights, capturing everything from action items and deadlines to strategic decisions and project milestones.

However, ConCraft elevates this process. With a few simple commands, team members can quickly access any piece of information stored in these datasheets. Whether you need to review the meeting agenda, catch up on the latest project developments, or clarify specific action items assigned to you, ConCraft is your go-to