System Design Phase

For the

Fresenius Medical Care

Conversational AI

May 8th, 2024

prepared by

Team #503_2

1. Executive Summary

This report outlines the System Design Phase of our Conversational AI project at Fresenius Medical Care. This project focuses on enhancing the HR interaction process. Our objective is to streamline HR inquiries, making them quicker and more user-friendly through deploying an AI-powered chatbot system.

Since our last phase, we have refined our system analysis based on the feedback that we received. This document details the architecture of the proposed system, illustrated by Data Flow Diagrams that show how data flows and Entity Relationship Diagrams map the relationships within our database. The implementation of a CRUD Matrix ensures that our data remains consistent and accurate throughout the system's operations.

Central to our design is the AI chatbot, designed to handle routine HR inquiries autonomously, thus freeing up human resources for more complex tasks. The chatbot will be powered by the Meta LLaMA 3-8B model within AWS Bedrock, reflecting our commitment to leveraging cutting-edge technology to enhance functionality and user experience. This section also includes interactive examples to showcase how users will interact with the new system, both for inputting and receiving information.

The Physical System Design section evaluates potential solutions for integrating AI into our processes by weighing their feasibility from operational, technical, and economic perspectives. It highlights the strategic choice of AI models and cloud infrastructure that will support the chatbot's integration into the existing HR workflows.

We have included a phased Implementation Plan to ensure a smooth rollout of the new system, which aims at encouraging easy adoption and minimizing any disruptions. This project phase has been both challenging and informative, bringing us closer to achieving significant improvements in HR operations through the strategic integration of AI into our processes.

2. Evaluation Letter

FRESENIUS
MEDICAL CARE

Client Evaluation Letter

To whom it may concern,

I am Global Head of Data Platform & Engineering and Data Management at Fresenius Medical Care, a publicly traded global company that provides kidney dialysis services and manufactures dialysis-related products. We work within a larger global Digital Technology and Innovation team and provide data platform and analytics solutions for all different aspects of our business. We have just completed the Spring 2024 semester.

Fresenius Medical Care is a world leader in providing products and services for individuals suffering from chronic kidney disease. They are the largest provider of dialysis treatments globally, operating over 4,050 clinics and serving around 344,000 patients worldwide. Their product portfolio also includes dialysis machines and dialyzers.

With the growth of technology in Data-driven decision-making and AI, the data analytics team started to leverage these technological advancements to reach their mission objective better and faster.
Key objectives include:

- Transformed the management of HR inquiries at Fresenius Medical Care with cutting-edge Conversational AI technologies.
- Striving for a personalized, interactive customer experience by harnessing advanced Large Language Model frameworks.
- Enhancing employee engagement and providing comprehensive information on HR policies and processes.

A conversational AI chatbot will provide instant responses to employee inquiries, offering round-the-clock support and eliminating the need for customers to wait in queues or navigate through complex menus. This Gen AI framework will understand natural language and context, providing personalized assistance and guiding users through tasks seamlessly. Our goal is to increase higher employee satisfaction levels and enhances the overall experience.

Chatbots automate routine tasks and processes, such as answering FAQs, handling customer inquiries, and routing requests to the appropriate department. This will free up human agents to focus on more complex issues, improving productivity and response times. Additionally, this chatbot will handle multiple conversations simultaneously, allowing businesses to scale their customer support operations efficiently without increasing costs proportionally.

Conversational AI chatbots will gather valuable data from customer interactions, including common queries, user preferences, and areas for improvement. This data can be analyzed to identify trends, optimize responses, and refine the chatbot's capabilities over time. By leveraging Large Language Model (LLM) algorithms, the chatbot can continuously learn and adapt to provide more accurate and effective assistance, driving ongoing improvements in customer service quality.

We recently concluded our final call for this semester, during which we reviewed the project's status, projected deliverables, and upcoming steps. The presentation they shared with me met our expectations impeccably. The dedication and effort invested by the team were evident, making it clear how far they've come.

My key takeaways at this stage are:

- Through my interactions with the team, I have gained a high level of confidence in the integrity of the project's work.
- Our company's deliverables will include the development of a Conversational AI Chatbot, leveraging Generative AI, poised to revolutionize HR enquiry handling.
- Anticipated outcomes from these deliverables should yield Increased Efficiency and Scalability, Data-driven Insights and Continuous Improvement, ultimately leading to an Improved Customer Experience.

I have to say, I am impressed with the overall process to this point. It is nice to be able to work with smart, outgoing, and affable college students like these ones. I applaud the University for encouraging students to work with us on this project.

Our company benefiting from this is something my team and I are very thankful for. We can always use help to strengthen our business.

Respectfully,
Kanti Singh
Director of Data & Analytics - Data Platforms & Data Management
Data & Analytics Center of Excellence
Global Digital Technology & Innovation
Fresenius Medical Care
Email: kanti.singh@fmc-na.com | Website: www.fmcna.com

FRESENIUS
MEDICAL CARE

3. Statement of Work (Approved)

Client and Industry Background

Fresenius Medical Care stands as a global leader in kidney care, recognized for its comprehensive services and significant impact on patient care in the dialysis sector. With a robust presence across the world, the company operates approximately 4,100 dialysis centers, including around 2,700 in the United States and North America, highlighting its extensive reach and dedication to addressing the needs of patients with kidney ailments. Fresenius Medical Care is distinguished for having the largest share of home hemodialysis (HD) treatments, underscoring its commitment to offering flexible and patient-centric solutions. As the foremost value-based renal care company in the United States, it sets the standard for excellence and innovation in the industry.

Fresenius Medical Care's operations encompass two primary domains: care delivery and care enablement. The care delivery aspect focuses on providing direct patient care in clinics worldwide, showcasing the company's dedication to healthcare delivery. In contrast, care enablement revolves around the production and distribution of dialysis products, serving both Fresenius Medical Care's own network and other dialysis providers, including major companies and smaller entities. This dual approach not only amplifies its influence in the healthcare sector but also positions Fresenius Medical Care as a pivotal player in advancing dialysis care and technology. The organization's extensive data collection, managed by the Global Medical Office, further exemplifies its role in pioneering research and development within the kidney care industry.

The Problems, Opportunities, or Directives

Performance: The current manual process of fielding HR inquiries is inefficient, diverting valuable time that could be allocated to patient care. The introduction of conversational AI aims to streamline this process, enhancing both efficiency and effectiveness by automating routine information delivery and allowing staff to focus more on patient care.

Information: The current process for employees to obtain HR information is inefficient, leading to delays. The proposed AI chatbot will enable instant, interactive access to information, streamlining the process and saving time.

Economics: While not expensive, traditional ways of responding to employee inquiries take considerable amounts of time. By streamlining these processes, conversational AI will increase productivity and decrease the need for additional labor.

Control/Security: Ensuring up-to-date and consistent information across employee interactions poses a significant challenge. Conversational AI can offer a controlled environment where updates are centrally managed and disseminated, ensuring all employees receive the same high-quality information.

Efficiency: Repetitive employees' inquiries create additional workloads for staff. Conversational AI can address these inquiries directly, freeing up staff time and streamlining the delivery and personalization of employees’ requirements.

Service: There is a clear demand for personalized and readily accessible information. The implementation of a conversational AI chatbot can personalize responses for specific employee inquiries, improving the effectiveness and quality of HR services.

Project scope

Project objective

Conversational AI: Take a revolutionary trip with our state-of-the-art conversational AI that has been cleverly combined with policy tech. We plan to transform the employee experience in the upcoming year, reaching previously unheard-of levels of engagement and pleasure. With the help of our intelligent chatbot, HR procedures will be streamlined, opening up new possibilities for efficiency and allowing employees to concentrate on what they do best. This strategic project aims to seamlessly integrate human expertise and technology to make every HR engagement enjoyable. Our success will be gauged not only by the increased output it yields but also by the noticeable improvement in the mood at work. Observing the shift as each encounter becomes evidence of our dedication to quality, carefully measured using strong engagement metrics, thoughtful employee input, and the expert effectiveness of our query response. Come along with us as we reinvent employee engagement for the future.

High-Level Requirements

Scope Inclusions:

- **Conversational AI Development:** Design and implement conversational AI for both web and mobile platforms, ensuring seamless integration with existing employee policy materials. This includes leveraging natural language processing (NLP) to facilitate interactive and engaging employee experiences.
- **System Analysis and Design:** Evaluate existing HR queries and procedures to create the AI's decision-making structure, which includes an extensive FAQ list and response mechanisms tailored to certain categories.
- **Data Integration and Security:** Assure the conversational AI's safe integration with Workday and Policy Tech's current systems, keeping in mind privacy and security laws and emphasizing real-time access to employee data and HR policies.
- **AI Development and Training:** Create an AI tool that allows for natural user interaction, and then apply machine learning to teach it using simulated interactions and historical HR queries.

- **Testing and Quality Assurance:** Confirm the efficacy of the AI in responding to HR queries by conducting extensive testing, including user acceptance testing (UAT) with employees.
- **Deployment and Monitoring:** The conversational AI will be gradually introduced to the workforce while ongoing data collecting on user engagement will be monitored.
- **User Experience Enhancements:** Focus on improving the overall user interface and experience to make the conversational AI system intuitive and accessible to employees with varying levels of tech-savviness.

Exclusions:

- **Medical Content Creation:** The project will not involve the creation of new medical content or the modification of existing clinical protocols. Instead, it will focus on helping employees with HR queries.
- **Diagnostic Features:** The development of medical diagnostic features within the conversational AI system is excluded to maintain the project's focus on education and information dissemination.
- **Integration with External Non-Fresenius Healthcare Systems:** The project will not include integration with external healthcare systems or databases outside of the Fresenius network to ensure data privacy and compliance with healthcare regulations.

Use Case Integration

Enhancing Employee Engagement and Efficiency through Conversational AI in the HR department.

Goal: To deploy a conversational AI system within a 12-month period to improve employee engagement and satisfaction, and to reduce the time staff spends on repetitive, time-inefficient HR tasks.

Actors:

- HR Team
- Employees (users of the system)
- IT Department (system implementers)

- Conversational AI System (chatbot)

Preconditions:

- Workday and Policy Tech systems are in place and contain up-to-date HR policies and employee data.
- The HR team has identified frequently asked questions and common employee positions and departments asking the questions.
- IT infrastructure supports the integration of AI technologies.

Basic Flow:

- Employees ask a question.
- Feed question to the chatbot.
- Identifying if the question is an FAQ or not.
    - If yes, output answer via Chatbot.
    - If not, ask the user to raise a ticket for HR.
- Give answers via Chatbot using PolicyTech and Workday.

Postconditions:

- Employees engage with conversational AI for HR inquiries.
- Reduced the number of routine inquiries handled by the HR staff.

Success Metrics:

- **User Engagement Metrics:** Track the number of inquiries handled by the AI, the resolution rate, and the average handling time.
- **Staff Feedback:** Conduct surveys to gauge employee satisfaction and gather qualitative feedback on the system's performance and usability.
- **Efficiency of Inquiry Handling:** Measure the reduction in time the HR team spends on inquiries since the implementation of the AI.

Exception Paths:

- An employee asks a question that the AI cannot answer, triggering an escalation to the HR team.
- Technical issues arise with integration, requiring IT intervention.
- Frequency: Daily interaction expected from employees across the organization.

Business Rules:

- The AI should prioritize data security and compliance with privacy regulations.
- The AI should escalate complex inquiries to the HR team based on predefined criteria.

Outcomes:

- Enhanced Employee Engagement: The AI provides quick, accurate responses, improving the overall employee experience.
- Increased HR Efficiency: The HR team can focus on strategic tasks as the AI handles routine inquiries.
- Continuous Improvement: Ongoing analysis of engagement metrics and feedback leads to regular system enhancements.

Risks and Mitigations:

- User adoption may be slower than expected plan for comprehensive change management and user training.
- Integration complexities with existing systems; involve IT experts early in the project to anticipate technical challenges.
- By clearly defining the scope, including the integration of the use case and how success will be measured, stakeholders will have a shared understanding of what the project entails and what it aims to achieve.

Constraints

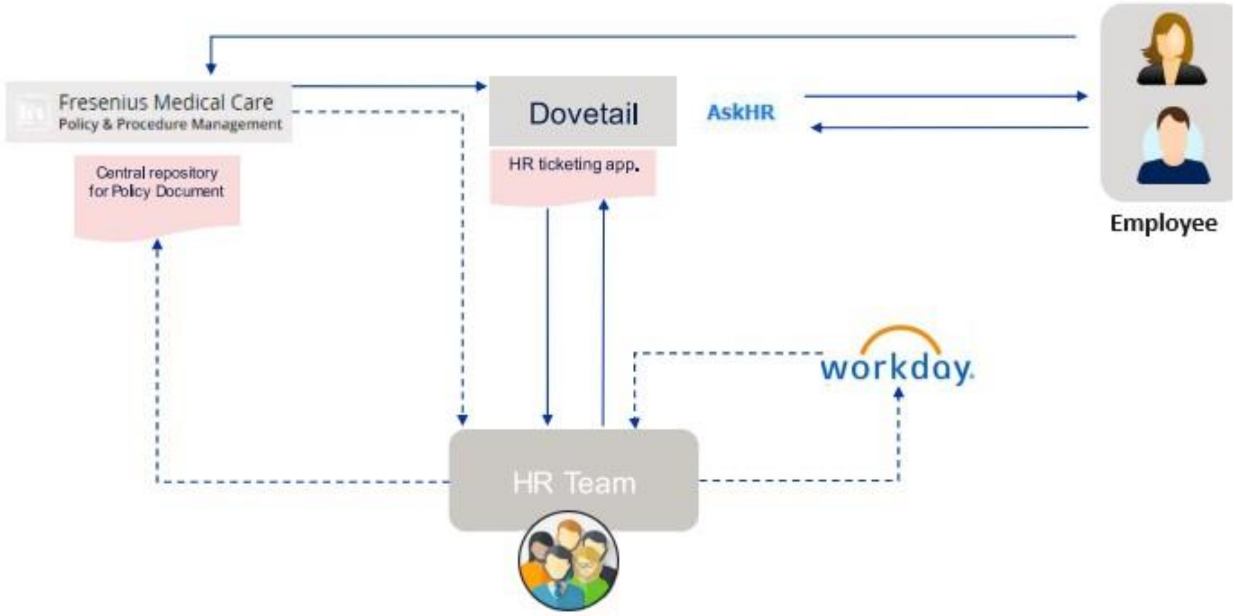
- **Client Constraints:** Strict adherence to policy regulations and budget limitations are critical. The project must ensure data privacy and security to comply with policy standards, presenting a major constraint in the development and implementation of conversational AI.

- **Team Constraints**: The project team may encounter limitations due to restricted access to specialized AI development talent. This, along with potential technical challenges, could impact the project timeline. Additionally, limited resources and the need for specialized skills in AI and healthcare data management further complicate the project execution.

4. System Analysis

(Approved)

a. Current System



Fact findings and information-gathering techniques used in studying the current systems.

The current system has the following characteristics:

1. **Policy Tech:** It serves as the central repository for HR policy documents. Employees can access the policies stored directly through the internal policy tech website. The Policy Documents are stored in SharePoint.
2. **Ticketing System (Dovetail):** Users create tickets in the ticketing system and receive answers therein when they have questions or problems. This is a standard procedure for monitoring and handling requests as well as making sure that all correspondence is recorded and auditable.
3. **Workday (Employee Data):** HR, finance, and planning are frequently included in Workday, a well-liked cloud-based enterprise management platform. Here, it is used to store employee data, such as personal information, work history, benefits, and more.
4. **Manual operations:** Data entry, data transfer between systems, manual ticket handling, and other manual operations seem to be highly relied upon in the architecture. This can frequently result in longer processing times, a larger chance of errors, and inefficiencies.

The functionality of the **ticketing system** within the present design will be provided by the following details:

1. **Queuing Mechanism:** All user requests and issues are arranged in a queue for the ticketing system to function.
2. **Ticket Assignment:** HR staff members can see the tickets which are open and pick the tickets based on their bandwidth. Availability, level of experience, or the specifics of the problem described in the ticket may all play a role in this.
3. **Problem Solving:** After receiving a ticket, an HR agent attempts to address the user's problem. Consulting Workday employee data and accessing the policy documents from policy tech may be necessary for this.

In simple terms, to avoid overworking any team members, load balancing makes sure that tickets are split equally across the HR staff members who are available.

This process description is a good example of a service management system that is influenced by the ITIL (Information Technology Infrastructure Library) that aims to match the needs of the business and its users with IT services. Conversational AI enters the picture when the company analyzes these factors and works constantly to refine the system to increase customer pleasure, decrease resolution times, and increase productivity.

Improving such a system would typically focus on automating manual processes where possible, improving data integration between Dovetail and Workday, and streamlining ticket management to enhance efficiency and reduce response times.

b. Data Flow Diagram

Context Level Diagram:

Description: The Context Level Data Flow Diagram (DFD) presents the overarching process of handling inquiries and requests within a Human Resources (HR) management system. It highlights the flow of information between the employees and the HR department, facilitating efficient communication and documentation management.

External Entities:

User: Represents employees or individuals who interact with the HR system, seeking answers to questions or requesting specific policy documents.

Human Resources: Depicts the HR department responsible for managing employee information, responding to inquiries, and handling policy documents.

Process:

Policy Inquiry Facilitation: Acts as the main process that manages incoming questions from users and channels policy document requests. It also ensures that tickets related to employee information are addressed appropriately, whether they are unanswered or answered.

Data Flows:

- **User to Policy Inquiry Facilitation:**

  Question: Data flow carrying inquiries from the user to the HR management system for processing.
- **Policy Inquiry Facilitation to Human Resources:**

  Employee Information: Data flow representing the employee-related information to the HR department.

  Unanswered Ticket: Data flow indicating inquiries that have not yet been addressed by the HR department.

  Policy Document: Data flow consisting of the requested HR policy documents directed toward the user sent to the HR department.
- **Human Resources to Policy Inquiry Facilitation:**

Answered Ticket: Data flow signifying inquiries that have been responded to by the HR department.

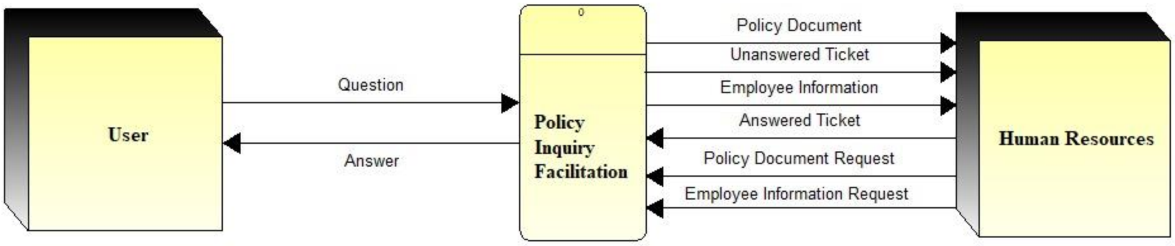
Policy Document Request: Data flow indicating a user's request for specific HR policy documents.

Employee Information Request: Data flow representing the need for employee-related information from the HR department.

- **Policy Inquiry Facilitation to User:**

  Answer: Data flow responding to the user's inquiries.

Visible Analyst Project Name : BUDT_723_CONVERSATIONAL_AI



Level 0:

Description: The Level 0 Data Flow Diagram (DFD) illustrates the foundational processes involved in a human resources information system, capturing the workflow for handling policy inquiries and employee information requests.

External Entities:

User: Represents employees or individuals who interact with the HR system, seeking answers to questions or requesting specific policy documents.

Human Resources: Depicts the HR department responsible for managing employee information, responding to inquiries, and handling policy documents.

Process:

Managing Tickets (Process 1): Operates as the core process for handling user inquiries, transforming them into ticket requests and delivering ticket responses after processing and resolution.

Retrieve Employee Information (Process 2): This process fetches particular employee information from the Employee Information data store in response to requests from Workday or Human Resources.

Retrieve Document Information (Process 3): Dedicated to retrieving specific policy documents from the Policy Document data store and passing them on to the Human Resources department or directly back to the user.

Data Stores:

Policy Tech: Maintains a repository of various HR policy documents ready to be accessed upon request.

Workday: Contains detailed records of employee data, accessible to HR-related processes for various informational needs.

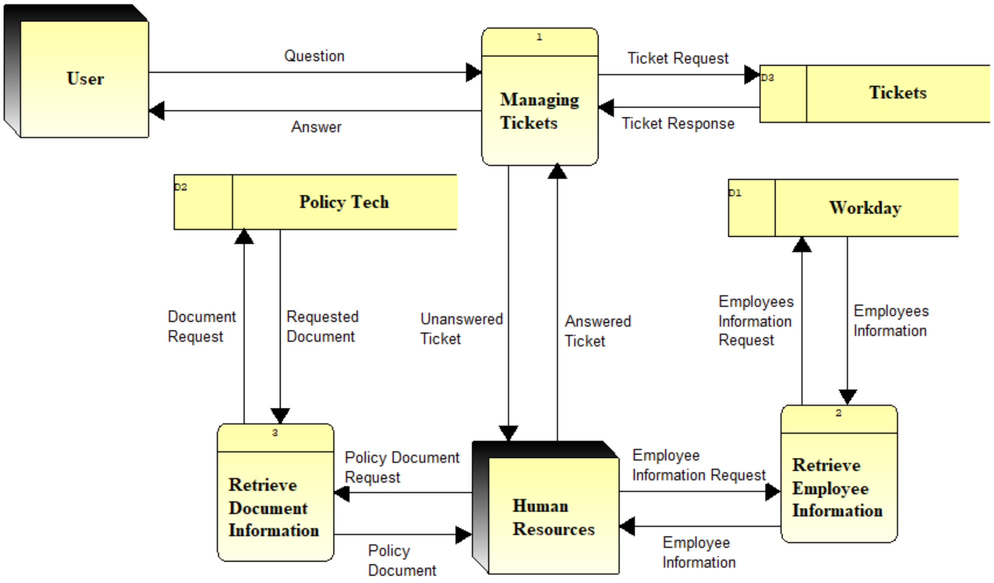
Tickets: Maintains a log of all ticket requests, both open and closed, serving as a reference for inquiries made to the HR department.

Data Flows:

- **User to Managing Tickets:**
  Question: Data flow carrying the user's inquiries to the Managing Tickets process.
- **Managing to User Tickets:**
  Answer: Data flow delivering responses to the user's inquiries.
- **Managing Tickets to Tickets:**
  Ticket Request: Data flow inputting new inquiries into the Tickets data store for tracking and processing.
- **Tickets to Managing Tickets:**

Ticket Response: Data flow outputting the resolution or information regarding user inquiries back to the user.

- **Policy Tech to Retrieve Document Information:**
  Requested Document: Data flow returning the requested policy document.
- **Retrieve Document Information to Policy Tech:**
  Policy Document Request: Data flow requesting retrieval of a specific policy document from the policy tech.
- **Human Resources to Managing Tickets:**
  Unanswered Ticket: Data flow showing pending inquiries that need responses.
- **Managing Tickets to Human Resources:**
  Answered Ticket: Data flow highlighting inquiries that have been resolved and need to be updated in the Tickets data store as closed.
- **Human Resources to Retrieve Employee Information:**
  Employee Information Request: Data flow asking for detailed employee information from the data store.
- **Retrieve Employee Information to Human Resources:**
  Employee Information: Data flow delivering the requested employee information back to Human Resources.
- **Human Resources to Retrieve Document Information:**
  Policy Document Request: Data flow requesting retrieval of a specific policy document from the data store.
- **Retrieve Document Information to Human Resources:**
  Requested Document: Data flow returning the requested policy document to Human Resources.



Level 1:

Description: The Level 1 Data Flow Diagram (DFD) provides a granular depiction of the processes that facilitate the handling of employee inquiries and the management of response tickets within the HR system. It outlines the specific actions taken from the moment a user raises a question to the updating and storing of ticket information in response to those queries.

External Entities:

User: Represents employees or individuals who interact with the HR system, seeking answers to questions or requesting specific policy documents.

Human Resources: Depicts the HR department responsible for managing employee information, responding to inquiries, and handling policy documents.

Processes:

Create Ticket (Process 1.1): This process is where a user's question is converted into a ticket for tracking and resolution within the system.

Save Ticket Details (Process 1.2): After a ticket has been processed and a response is generated, this process ensures the details of the ticket, including the resolution, are stored, and updated within the system.

Data Stores:

Tickets: A data store that keeps records of all ticket requests and responses, tracking the status of each as either 'Unanswered' or 'Answered'.

Workday: Contains detailed records of employee data, accessible to HR-related processes for various informational needs.

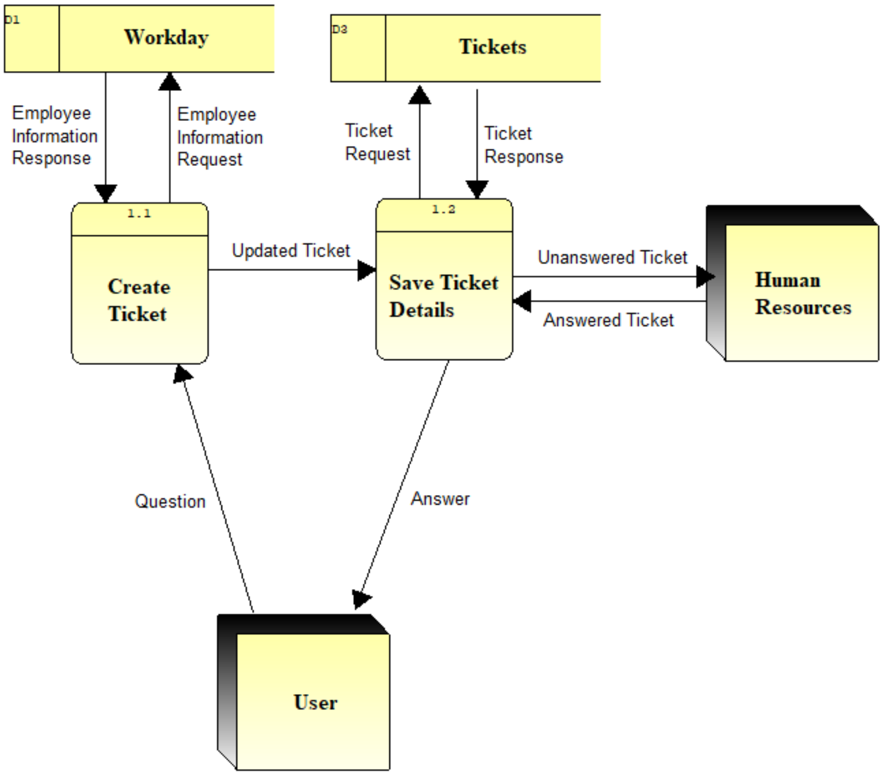
Data Flows:

- **User to Create Ticket:**
  Question: Data flow that captures the inquiry from the user, initiating the creation of a ticket.
- **Create a Ticket to Save Ticket Details:**
  Updated Ticket: Data flow representing the newly created or updated ticket information with employee details ready to be stored in the Tickets data store.
- **Save Ticket Details to Human Resources:**
  Unanswered Ticket: Data flow categorizing tickets that have not yet been addressed by the HR department.
- **Human Resources to Save Ticket Details:**
  Answered Ticket: Data flow labeling tickets that have received a response from HR.
- **Save Ticket Details to User:**
  Answer: Data flow providing the user with the information or resolution to their query.
- **Create Ticket to Workday:**
  Employee Information Request: Data flow initiating a request for information that results in ticket creation.
- **Workday to Create Ticket:**
  Employee Information Response: Data flow sending back the requested information to the Workday system as a result of the ticket query.
- **Save Ticket Details to Tickets:**

Ticket Request: Data flow transmitting the details of new and ongoing tickets to the HR department for processing.

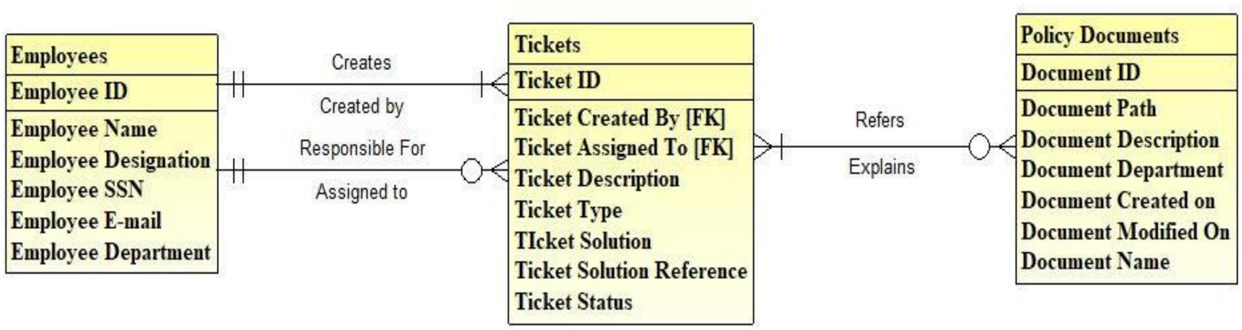
- **Tickets to Save Ticket Details:**

  Ticket Response: Data flow returning the results from HR, including both unanswered and answered ticket updates.



c. Entity Relationship Diagram

This diagram is an Entity-Relationship Diagram (ERD) commonly used in system analysis and database design to illustrate the relationships between data entities in a system. It depicts three primary entities: Employees, Tickets, and Policy Documents.



The attributes listed for each entity in the provided ERD:

Employees: Employee ID, Employee Name, Employee Designation, Employee SSN, Employee Email, Employee Department

Tickets: Ticket ID, *Ticket Created By, Ticket Assigned To,* Ticket Description, Ticket Type Ticket Solution, Ticket Solution Reference, Ticket Status

Policy Documents: Document ID, Document Path, Document Description Document Department, Document Created On, Document Modified On Document Name

These attributes provide the characteristics and properties that define each entity within the system.

Cardinality between the entities:

Employees to Tickets:

"Creates": One-to-Many, where one employee can create one or many tickets, and a ticket can be created by only one employee.

"Responsible For": One-to-Many, where one employee is responsible for 0 or many tickets, and one ticket is being handled by only one employee.

Tickets to Policy Documents:

"Refers": Many-to-Many, where one ticket can refer to many policy documents, and one policy document can explain one or many tickets.

5. Proposed System

Proposed System

The employee now has three options for obtaining inquiries about HR.

1. Dovetail Ticketing:

Purpose: This system serves as a traditional ticketing solution for handling HR-related issues.

Workflow: Employees can submit tickets detailing their queries or issues. These tickets are routed to the appropriate HR personnel, who then handle and resolve them.

Benefits:

Personalized Resolution: Allows for direct communication between the employee and HR, facilitating personalized solutions.

Tracking: Offers a clear tracking mechanism for employees to monitor their ticket status and see how it's being handled.

2. Direct Access to PolicyTech:

Purpose: PolicyTech provides employees with access to all relevant HR documents and policies.

Workflow: Employees can log in and browse through a database of documents, policies, and guidelines to find answers to their questions.

Benefits:

Immediate Access: Employees can instantly access a wide range of HR materials, from company policies to benefit details.

Self-Service: This option empowers employees to resolve their queries independently without waiting for HR support.

3. Chatbot (Conversational AI):

Purpose: The chatbot provides a conversational interface for employees to interact with and get answers to their HR questions.

Workflow: Employees can type in their questions directly to the chatbot.

The chatbot, powered by Amazon Lex and Bedrock NLP models, processes the query, and generates an appropriate response.

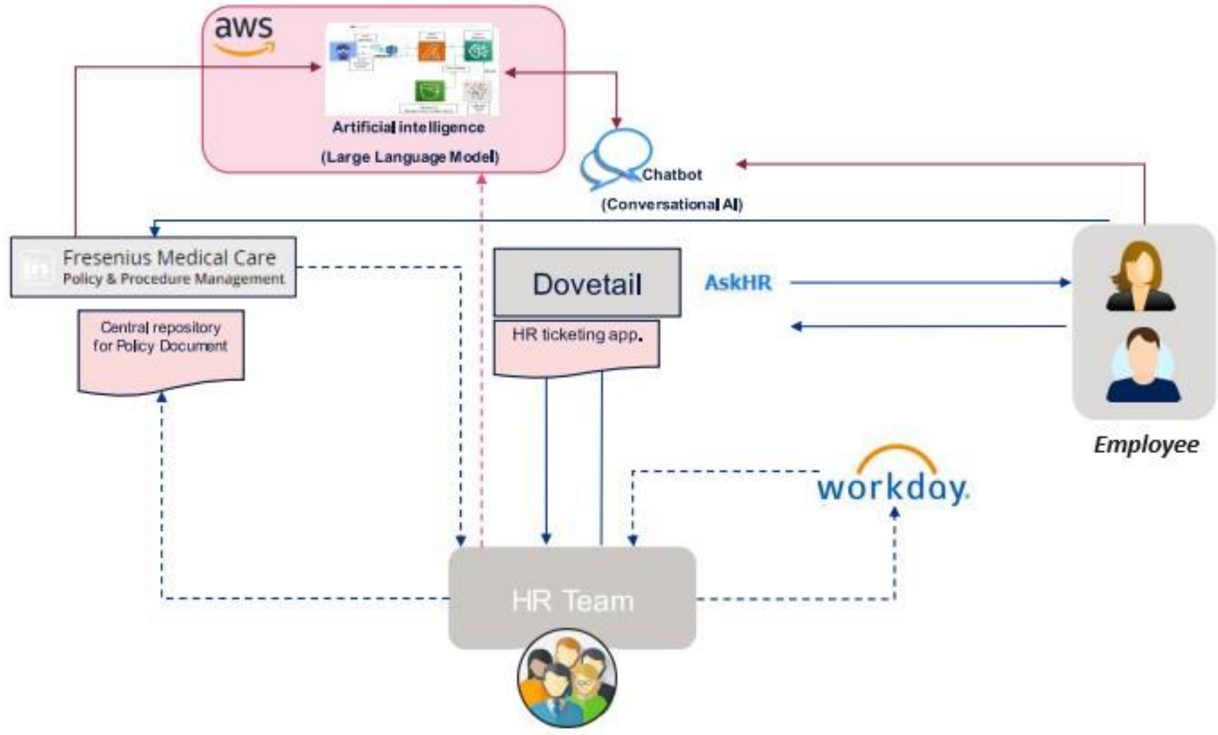
The response can include direct answers, links to PolicyTech documents, or suggestions to submit a ticket via Dovetail.

Benefits:

Immediate Responses: Provide instant answers, reducing the need to wait for HR personnel.

Integration: The chatbot can refer employees to PolicyTech or Dovetail for more detailed support, offering seamless navigation between solutions.

Scalability: Capable of handling multiple queries simultaneously, ensuring a consistent level of support for all employees.



a. Data Flow Diagram (Logical)

Context level diagram:

Description: This Context Level DFD showcases the primary interactions between the users, the Policy Inquiry Facilitation process, and the Human Resources (HR) department. It focuses on how inquiries regarding HR policies are handled and managed.

External Entities:

User: Represents employees or other individuals engaging with the HR system to inquire about policies.

Human Resources: The department responsible for managing responses to policy inquiries, handling policy documents, and updating employee information.

Process:

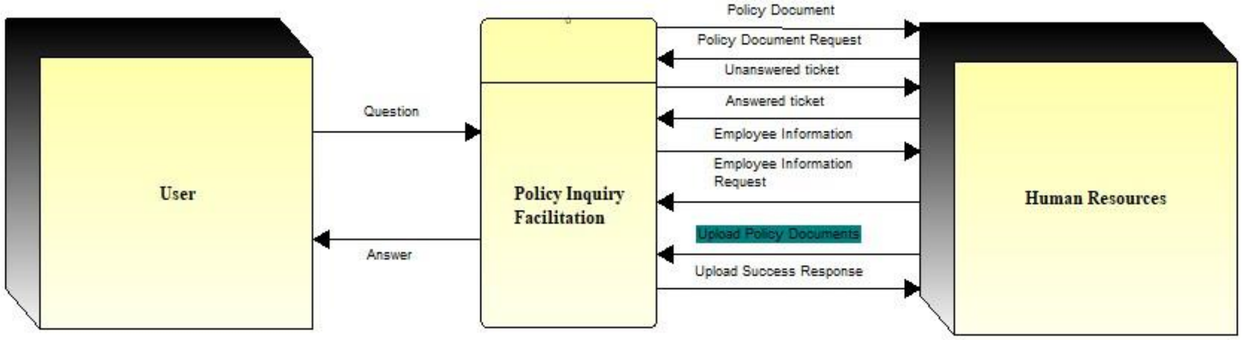
Policy Inquiry Facilitation: Central process that manages questions and requests from users related to HR policies. It serves as an intermediary between the user and the HR department, ensuring that all communications and document transfers are streamlined.

Data Flows:

- **Question:** Data flow from the User to the Policy Inquiry Facilitation, carrying inquiries about HR policies.
- **Answer:** Data flow from the Policy Inquiry Facilitation back to the User, providing the requested information or responses to inquiries.
- **Policy Document Request:** Data flow from the Policy Inquiry Facilitation to Human Resources, requesting specific policy documents.
- **Unanswered Ticket:** Data flow from Human Resources to the Policy Inquiry Facilitation, indicating inquiries that have not yet been addressed.
- **Answered Ticket:** Data flow from Human Resources to the Policy Inquiry Facilitation, signifying inquiries that have been resolved.
- **Employee Information Request:** Data flow from the Policy Inquiry Facilitation to Human Resources, requesting specific employee information.

- **Employee Information:** Data flow from Human Resources to the Policy Inquiry Facilitation, providing the requested employee details.
- **Upload Policy Documents:** Data flow from Human Resources to the Policy Inquiry Facilitation, involving the submission of new or updated policy documents.
- **Upload Success Response:** Data flow from the Policy Inquiry Facilitation back to Human Resources, confirming the successful upload and update of policy documents.

Visible Analyst Project Name : BUDT723_CONVERSATIONAL_AI_PROPOSED



Level 0:

Description: This Level 0 DFD illustrates the detailed processes involved in handling user inquiries within a human resources (HR) system. It depicts that the users have three options for getting answers to their queries about HR policies, employee information, and more. It describes how they interact with each of the three options.

External Entities:

User: Represents employees or other individuals engaging with the HR system to inquire about HR-related issues.

Processes:

Current Ticket Management System (Process 1): Manages and processes tickets raised by users, dealing with more complex or specific inquiries that require detailed handling.

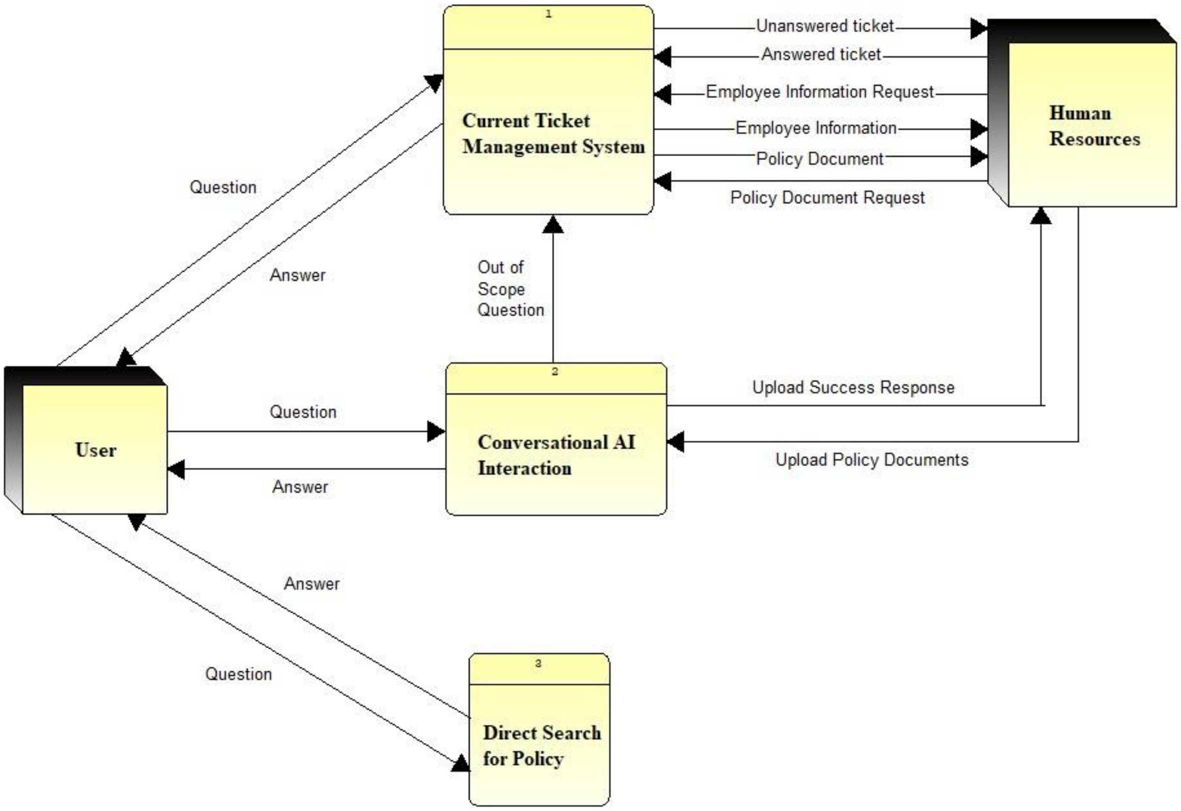
Conversational AI Interaction (Process 2): Provides immediate, AI-driven responses to user inquiries, handling standard questions and directing more complex ones to the appropriate systems.

Direct Search for Policy (Process 3): Allows users to directly search for HR policies, providing quick access to policy documents.

Data Flows:

- **Question:** Data flows from the User to all three systems (Current Ticket Management System, Conversational AI Interaction, Direct Search for Policy), initiating different types of interactions based on the nature of the inquiry.
- **Answer:** Data flows back to the User from each of the three systems, providing the requested information or the outcome of an inquiry.
- **Out of Scope Question:** Data flow from the Conversational AI Interaction to the Current Ticket Management System, indicating that a question cannot be handled by the AI system and needs further processing.
- **Employee Information Request:** Data flow from the Current Ticket Management System to Human Resources, requesting specific employee details.
- **Employee Information:** Data flow from Human Resources to the Current Ticket Management System, providing the requested employee details.
- **Policy Document Request:** Data flow from the Current Ticket Management System to Human Resources, requesting specific policy documents.
- **Policy Document:** Data flow from Human Resources to the Current Ticket Management System, delivering the requested policy documents.
- **Unanswered Ticket:** Data flow from Human Resources to the Current Ticket Management System, indicating inquiries that have not yet been addressed.
- **Answered Ticket**: Data flow from Human Resources to the Current Ticket Management System, indicating inquiries that have been resolved.
- **Upload Policy Documents:** Data flow from Human Resources to the Current Ticket Management System, involving the submission of new or updated policy documents.

- **Upload Success Response**: Data flow from the Current Ticket Management System back to Human Resources, confirming the successful upload and update of policy documents.



Level 1.1 – Current Ticket Management System

Description: This Level 1 DFD delves into the details of managing user inquiries and requests within a human resources system. It specifically outlines the flow of information between users, various data stores, and processes that facilitate the retrieval and management of HR-related information and documents.

External Entities:

User: Represents employees or individuals interacting with the HR system to seek information or submit requests related to HR policies and employee data.

Processes:

Managing Tickets (Process 1.1): Manages incoming inquiries from users, categorizes them, and facilitates the handling of both standard and out-of-scope questions.

Retrieve Document Information (Process 1.2): Dedicated to fetching HR policy documents based on user requests or system needs.

Retrieve Employee Information (Process 1.3): Handles the retrieval of specific employee information upon request from the Workday data store which contains employee information.

Data Stores:

Tickets (Data Store D3): Stores details about user inquiries, tracking their status as either unanswered or answered.

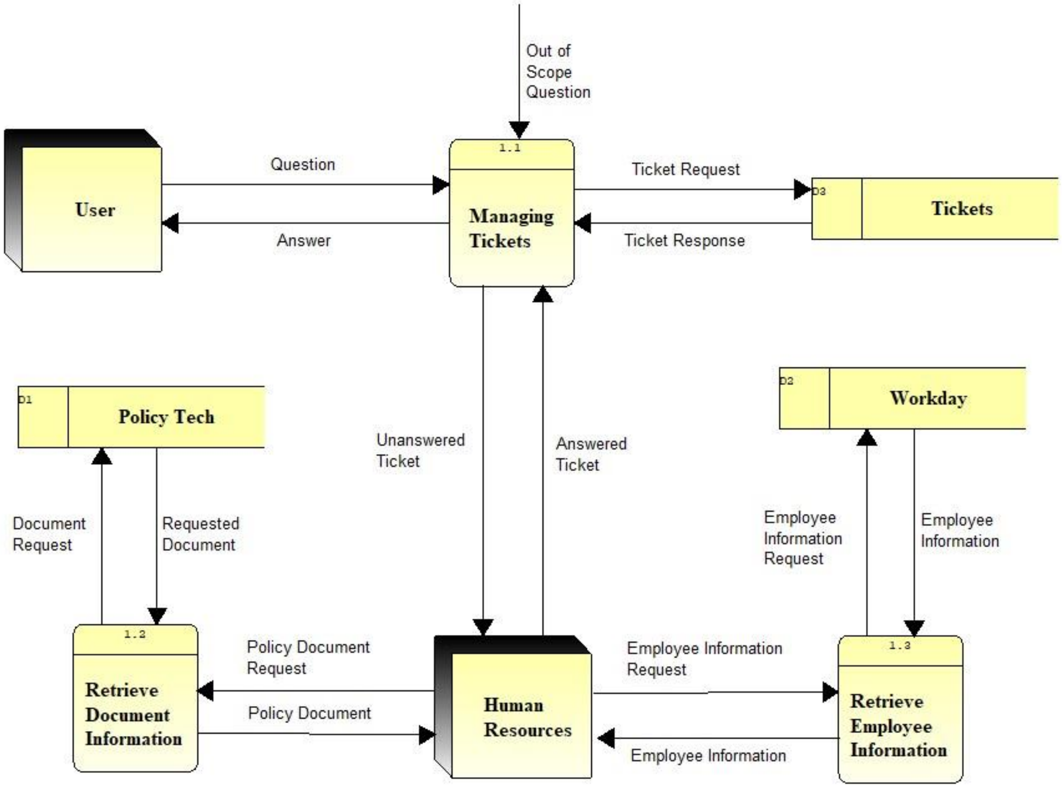
Policy Tech (Data Store D1): A repository of HR policy documents that are accessed upon request.

Workday (Data Store D2): Maintains detailed records of employee information, accessible for fulfilling data requests.

Data Flows:

- **Question**: Data flow from the User to Managing Tickets, initiating the inquiry handling process.
- **Answer**: Data flows back to the User from Managing Tickets and responding to inquiries.
- Out of Scope Question: Data flow from Managing Tickets, indicating questions that require more specialized handling than the system's standard processes.
- **Ticket Request**: Data flow into Tickets from Managing Tickets and registering new inquiries.
- **Ticket Response:** Data flow from Tickets back to Managing Tickets, providing the resolution or information needed to respond to the user.
- **Document Request:** Data flow from the User to Policy Tech, requesting specific HR documents.
- **Requested Document:** Data flow from Policy Tech to Retrieve Document Information, supplying the requested documents.

- **Policy Document Request:** Data flow from Managing Tickets to Retrieve Document Information, specifying the need for certain policy documents.
- **Policy Document**: Data flow from Retrieve Document Information back to Managing Tickets, delivering the requested policy documents.
- **Employee Information Request:** Data flow from Managing Tickets and directly from the User to Retrieve Employee Information, requesting specific employee details.
- **Employee Information:** Data flow from Retrieve Employee Information back to Managing Tickets and directly to the User, providing the requested information.
- **Unanswered Ticket:** Data flow from Tickets to Managing Tickets, indicating inquiries waiting for a response.
- **Answered Ticket:** Data flow from Tickets to Managing Tickets, signaling that an inquiry has been resolved.



Level 1.2 – Conversational AI Interaction

Description: This Level 1 DFD demonstrates the intricate workings of a conversational AI system designed to handle user inquiries about human resources policies, facilitate Knowledge Base updates, and integrate new policy onboarding processes.

External Entities:

User: Represents individuals interacting with the conversational AI system to ask questions related to HR policies.

Processes:

Chatbot Interface (Process 2.1): This is the front-end interface where users submit their questions. The chatbot captures intent and slots (specific data points like dates or names), which are required for the operation, and handles out-of-scope questions by suggesting the user to raise a ticket manually using the current ticketing system.

Logic Handler (Process 2.3): Processes validated responses from the LLM (Large Language Model) and direct tasks such as updating Knowledge bases or managing document uploads.

LLM Model (Process 2.2): Processes HTTP requests from the Chatbot Interface, generating appropriate responses based on the user's inquiry using advanced language understanding from the model training and generation capabilities.

New Policy Onboarding (Process 2.4): A process responsible for incorporating new policy documents into the system, ensuring that the Knowledge base is current and accurate.

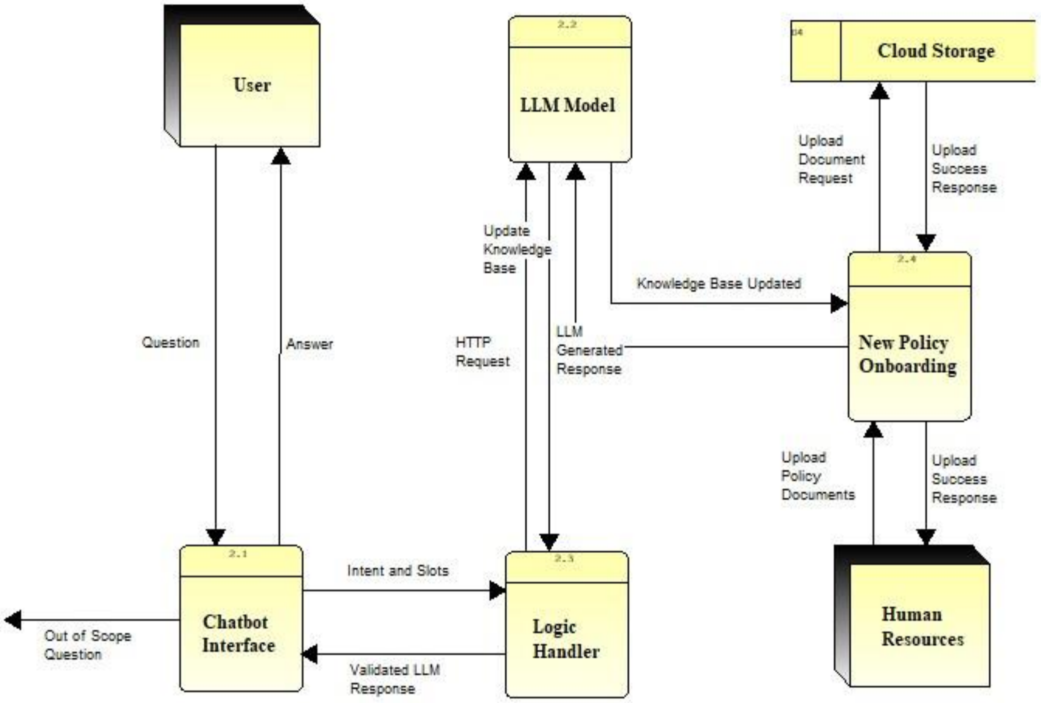
Data Stores:

Cloud Storage (Data Store D4): Stores and manages access to documents, handling requests for document uploads and returns successful responses.

Data Flows:

- Question: Data flow from the User to the Chatbot Interface, initiating the inquiry process.
- Out of Scope Question: Data flow from the Chatbot Interface back to the User, indicating questions that are beyond the chatbot's capacity to handle.
- Answer: Data flows from the Chatbot Interface back to the User, providing the generated answer.

- Intent and Slots: Data flow from the Chatbot Interface to the Logic Handler, conveying detected intent and necessary data extracted from the user's input.
- Validated LLM Response: Data flow from the Logic Handler back to the Chatbot Interface, confirming that the LLM's response is accurate and appropriate before sending it to the user.
- HTTP Request: Data flow from the Chatbot Interface to the LLM Model, requesting response generation.
- LLM Generated Response: Data flow from the LLM Model to the Logic Handler, delivering the generated response based on the user's input.
- Upload Document Request: Data flow from the Logic Handler to Cloud Storage, requesting the upload of new or updated policy documents.
- Upload Success Response: Data flow from Cloud Storage to the Logic Handler, indicating successful document upload.
- Knowledge Base Updated: Data flow from the Logic Handler to the LLM Model, indicating that the Knowledge base has been updated, enhancing the model's response accuracy for future inquiries.
- Upload Policy Documents: Data flow from Human Resources to New Policy Onboarding, indicating new policies that need to be integrated into the system.
- Upload Success Response: Data flow from New Policy Onboarding back to Human Resources, confirming successful integration of new policies.



Level 1.3 – Direct Policy Search

Description: This Level 1 DFD outlines a system designed to facilitate the retrieval of HR policy documents directly by users through keyword searches on the company website. The system streamlines the process of locating specific policy documents, ensuring users can access relevant information quickly and efficiently.

External Entities:

User: Represents individuals who interact with the system, searching for specific HR policy documents on the company website, which displays the Policy Tech documents.

Processes:

Search Using Keyword (Process 3.1): Further refines the search process by utilizing keywords entered by the user to display specific policies within a larger database or repository.

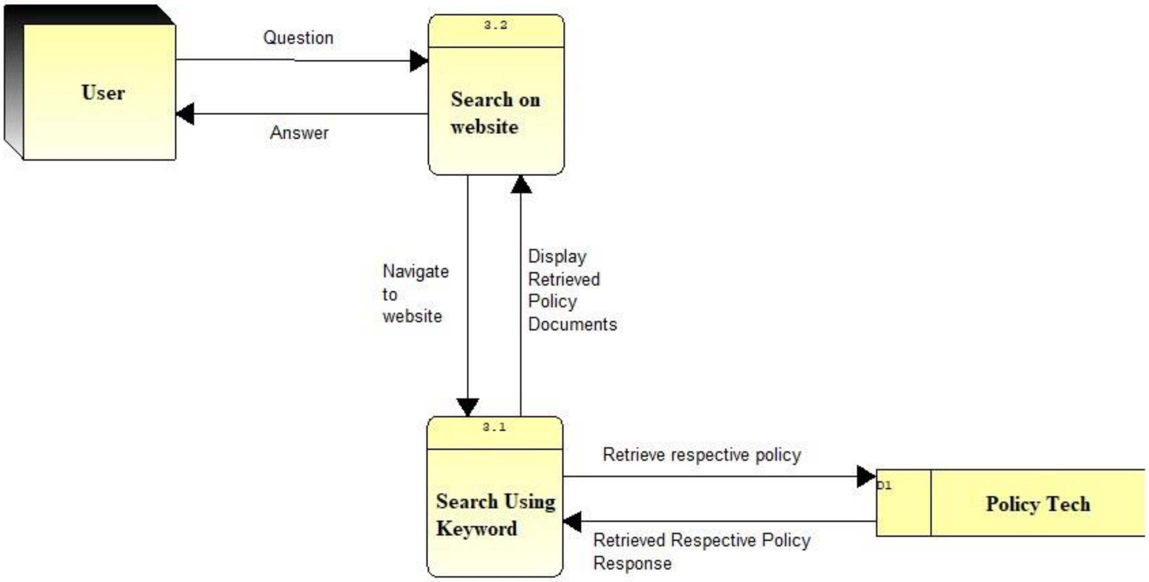
Search on Website (Process 3.2): This process handles the user's initial question by allowing them to navigate and search the website for specific policy documents.

Data Stores:

Policy Tech (Data Store D1): Serves as the repository for all HR policy documents, which can be accessed through keyword searches to retrieve specific policies.

Data Flows:

- **Question**: Data flow from the User to the Search on the Website, initiating the search process.
- **Navigate to Website:** Data flow representing the user's action of navigating within the website to access the search functionality.
- **Display Retrieved Policy Documents:** Data flow from the Search on the Website back to the User, presenting the documents that have been located based on the user's query.
- **Retrieve Respective Policy:** Data flow from Search Using Keyword to Policy Tech, requesting the retrieval of specific policy documents based on the identified keywords.
- **Retrieved Respective Policy Response:** Data flow from Policy Tech back to Search Using Keyword, delivering the specific policy documents that were requested.


User
Question
Answer
3.2
Search on website
Navigate to website
Display Retrieved Policy Documents
3.1
Search Using Keyword
Retrieve respective policy
Retrieved Respective Policy Response
D1
Policy Tech

Level 2.1 - Chatbot Interface

Description: This Level 2 DFD explores the detailed processes within a chatbot interface used to handle and respond to user queries. It illustrates the steps involved in text tokenization, intent recognition, slot filling, and invoking backend functions to generate appropriate responses.

External Entities:

User: Represents individuals interacting with the chatbot, submitting questions that are processed through various natural language processing steps.

Processes:

Text Tokenization (Process 2.1.1): Converts the user's question into tokens that can be easily processed for intent recognition. This is the initial step where raw text is segmented into manageable units (tokens).

Intent Recognition (Process 2.1.2): Analyzes the tokenized text to determine the user's intent. This process identifies the purpose behind the user's query.

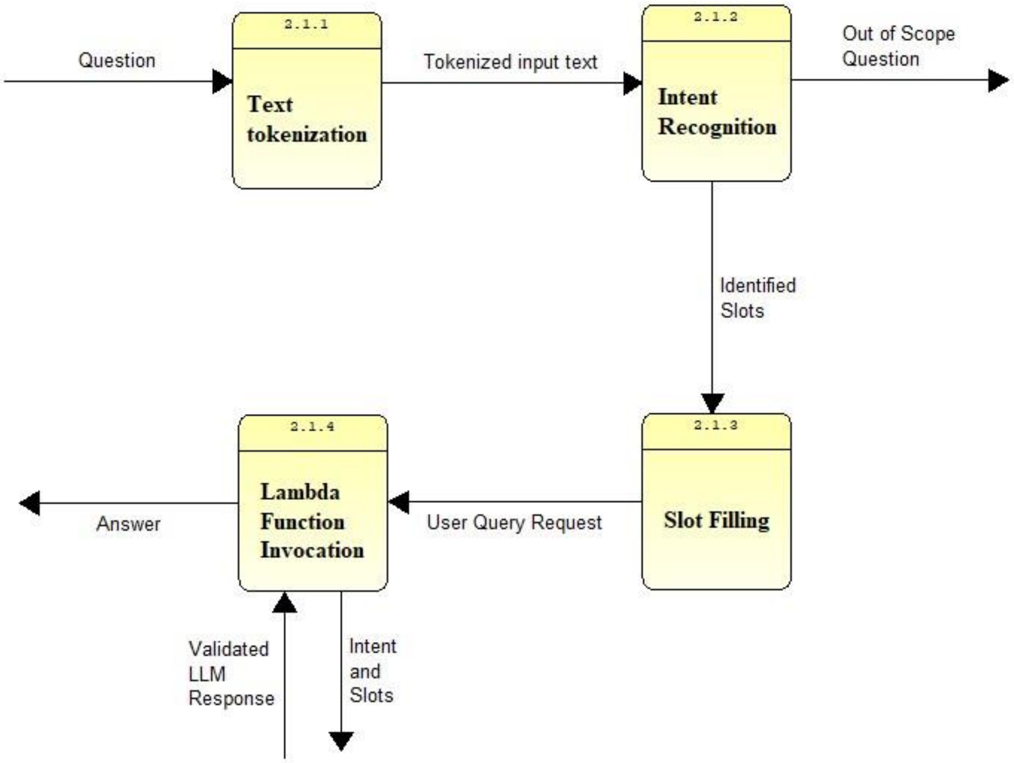
Slot Filling (Process 2.1.3): Based on the identified intent, this process extracts relevant information (slots) from the tokenized text, such as dates, names, or specific details pertinent to the user's query.

Lambda Function Invocation (Process 2.1.4): Invokes a backend function (Lambda function) that processes the identified intent and filled slots to generate a suitable response to the user's query.

Data Flows:

- **Question**: Data flow from the User to Text Tokenization, initiating the processing of the user's input.
- **Tokenized Input Text:** Data flows from Text Tokenization to Intent Recognition, carrying the segmented text.
- **Out of Scope Question:** Data flow from Intent Recognition to the User, indicating questions that cannot be processed within the given intent framework.
- **Identified Slots:** Data flow from Intent Recognition to Slot Filling, providing the necessary information to extract specific data points from the user's query.

- **User Query Request:** Data flow from Slot Filling to Lambda Function Invocation, conveying the complete, processed user query with intent and slots filled for final response generation.
- **Validated LLM Response**: Data flow from Lambda Function Invocation back to the User, delivering the final answer to the user's query, validated, and formatted appropriately.



Level 2.3 – Logic Handler

Description: This Level 2 DFD focuses on the Logic Handler, detailing the processes involved in validating user input, setting up client configurations, generating HTTP request payloads, making HTTP calls, and handling responses within a conversational AI system.

Processes:

Validate the Input (Process 2.3.1): Checks the intent and slots received from the user query for accuracy and completeness. This step ensures that all necessary data is correct and formatted properly before further processing.

Setup Bedrock Client (Process 2.3.2): Configures the client settings necessary for making HTTP requests. This process is crucial for establishing the parameters and authentication details required for subsequent API calls.

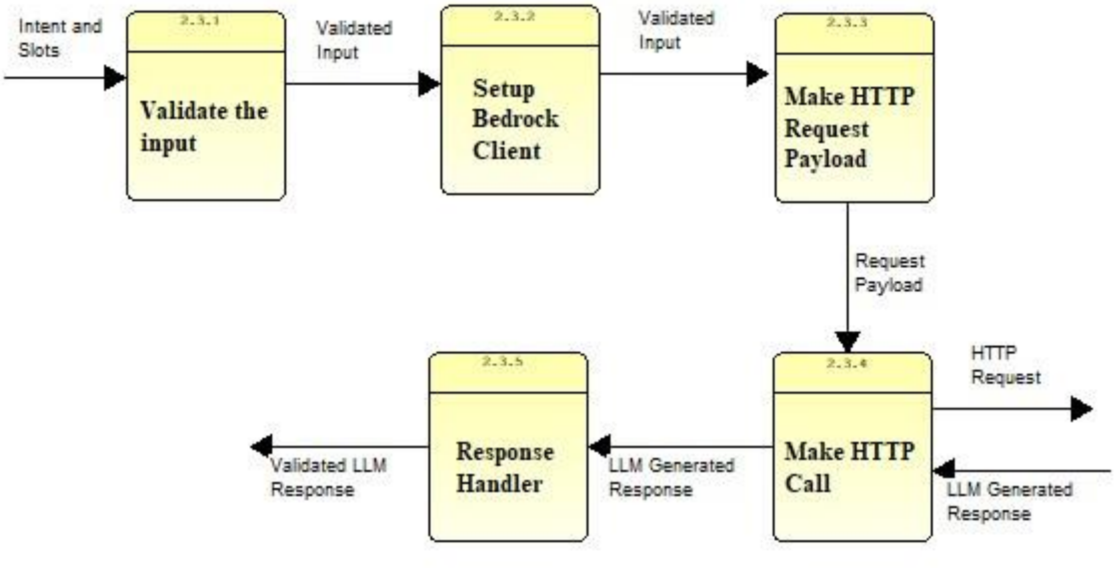
Make HTTP Request Payload (Process 2.3.3): Construct the HTTP request payload based on the validated input. This process involves formatting the request in a way that the endpoint can understand and process it effectively.

Make HTTP Call (Process 2.3.4): Executes the HTTP request using the payload prepared earlier. This step is where the system interacts with external services or APIs to fetch or send data.

Response Handler (Process 2.3.5): Processes the response received from the HTTP call. It evaluates the data returned by the LLM (Large Language Model) or external service, ensuring that the response is properly formatted and suitable for user presentation.

Data Flows:

- **Intent and Slots**: Data flow into Validate the Input, providing the foundational data needed for processing the user's request.
- **Validated Input:** Data flow from Validate the Input to Setup Bedrock Client and Make HTTP Request Payload, indicating that the input has been checked and is ready for further action.
- **Request Payload:** Data flow from Make HTTP Request Payload to Make HTTP Call, delivering the formatted request needed to retrieve data or interact with external APIs.
- **HTTP Request:** Represents the actual request being sent out to external services, originating from the Make HTTP Call process.
- **LLM Generated Response:** Data flow from the external service back to the Response Handler, providing the raw output from the LLM or API.
- **Validated LLM Response:** Data flow from the Response Handler back to the initial system or user interface, signifying the final response that has been processed and is ready for user presentation.



Level 2.2 – LLM

Description: This Level 2 DFD outlines the sequence of steps an LLM takes to process HTTP requests, generate query embeddings, fetch matching embeddings, generate responses based on these embeddings, and update its knowledge base accordingly.

Processes:

Validate HTTP Request (Process 2.2.1): Confirms the validity of incoming HTTP requests, ensuring they contain all necessary parameters and are correctly formatted.

Generate Query Embeddings (Process 2.2.2): Converts validated query parameters into embeddings. These embeddings represent the query in a numerical format that can be processed by the model to find relevant responses.

Fetch Matching Embeddings (Process 2.2.3): Retrieves embeddings from the embeddings data store that match or are closely related to the query embeddings.

Generate Response (Process 2.2.4): Uses the matched information from the fetched embeddings to generate an appropriate response to the user's query.

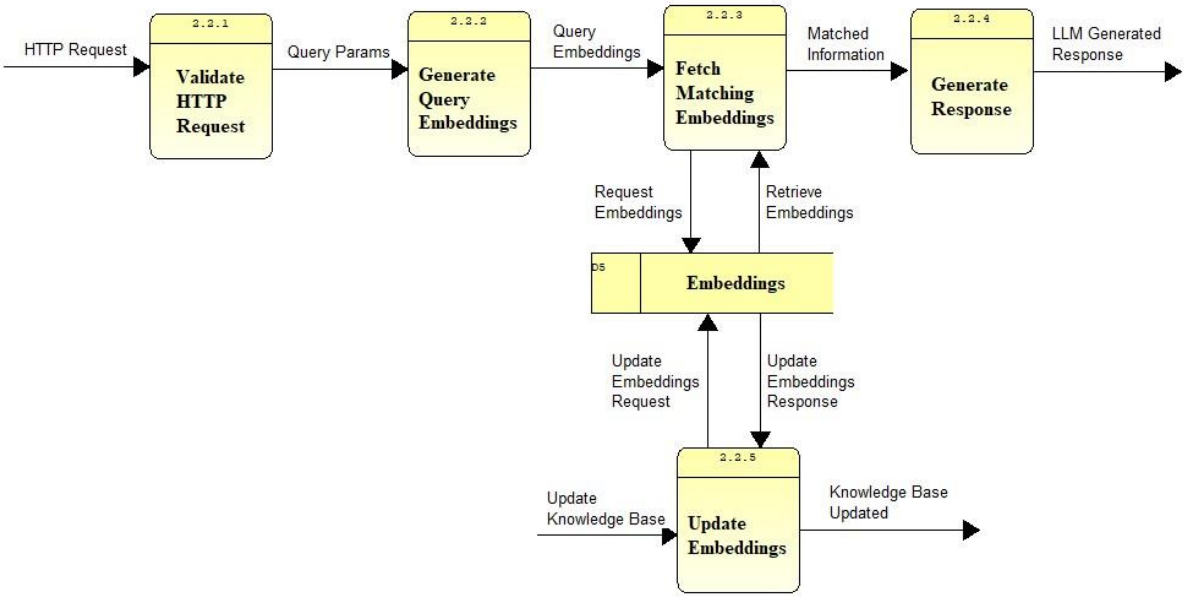
Update Embeddings (Process 2.2.5): Handles requests to update the embeddings in the embeddings data store, ensuring that the LLM's knowledge base remains accurate and up to date.

Data Stores:

Embeddings (Data Store D5): A data store that holds the embeddings used by the LLM to process and match queries. This store is updated as new information is integrated into the system.

Data Flows:

- **HTTP Request**: Data flow into Validate HTTP Request, initiating the validation process.
- **Query Params**: Data flow from Validate HTTP Request to Generate Query Embeddings, carrying the validated parameters for embedding generation.
- **Request Embeddings:** Data flow from Generate Query Embeddings to Fetch Matching Embeddings, signaling the need to retrieve similar embeddings.
- **Retrieve Embeddings:** Data flow from Embeddings to Fetch Matching Embeddings, providing the necessary embeddings for matching.
- **Matched Information**: Data flow from Fetch Matching Embeddings to Generate Response, delivering the matched embeddings used to create the user's response.
- **LLM Generated Response:** Data flow from Generate Response, outputting the final response back to the user or system.
- **Update Embeddings Request:** Data flow from Update Embeddings to Embeddings, indicating the need to update or modify embeddings based on new information.
- **Update Embeddings Response:** Data flow from Embeddings back to Update Embeddings, confirming the successful update of embeddings.
- **Knowledge Base Updated:** Data flow from Update Embeddings, signifying the integration of updated information into the knowledge base.



Level 2.4 – New Policy Onboarding

Description: This Level 2 DFD details the steps involved in integrating new policy documents into a system, from uploading the documents to cloud storage, converting them into embeddings, and updating the knowledge base of a conversational AI system.

Processes:

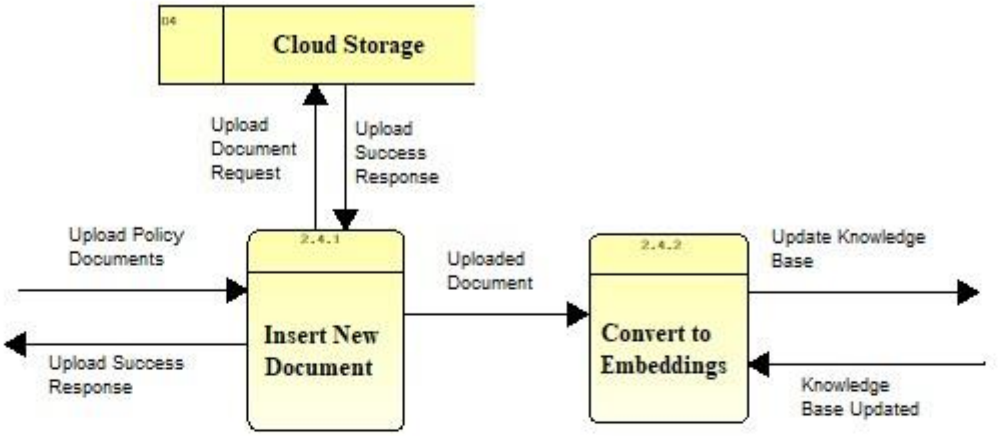
Insert New Document (Process 2.4.1): Handles the uploading of new policy documents to cloud storage. This process ensures that new documents are securely stored and appropriately cataloged.
Convert to Embeddings (Process 2.4.2): Transforms the uploaded documents into embeddings. These embeddings facilitate the AI's ability to understand and retrieve information from documents more efficiently.

Data Stores:

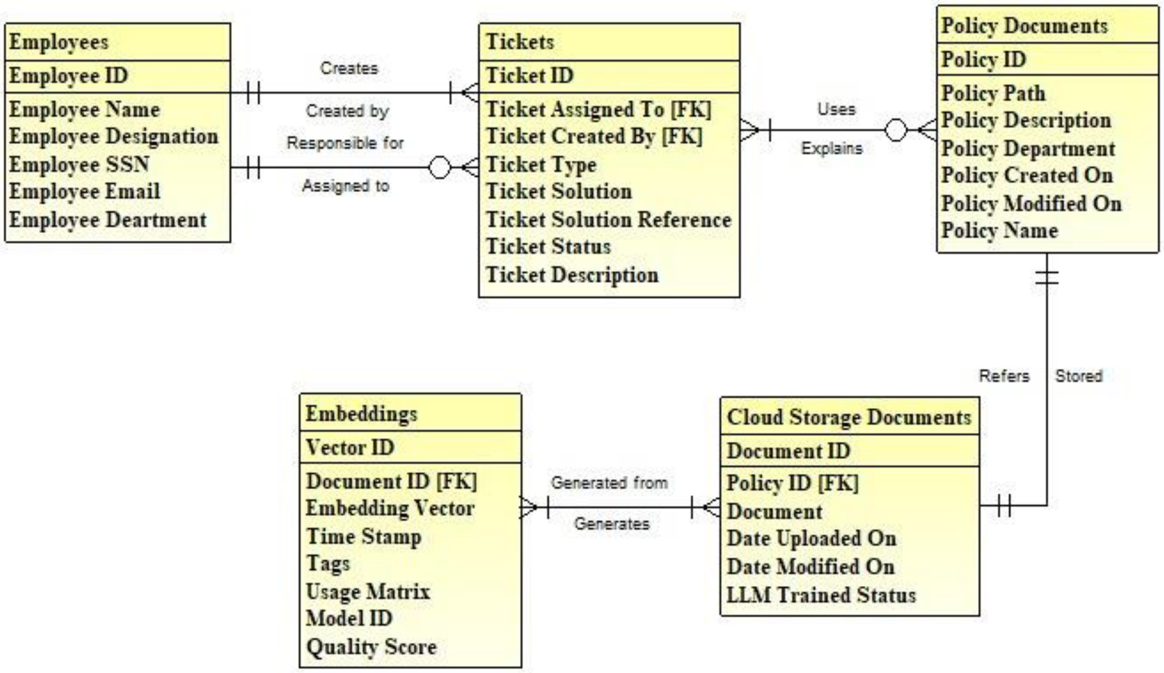
Cloud Storage (Data Store D4): Acts as the repository for storing policy documents. It receives and holds all uploaded documents until they are processed into the system.

Data Flows:

- **Upload Policy Documents:** Data flows from the external system or user to the Insert New Document, signaling that new policy documents need to be integrated.
- **Uploaded Document:** Data flow from Insert New Document to Convert to Embeddings, carrying the new documents that have been successfully uploaded to cloud storage.
- **Upload Document Request**: Data flow from Insert New Document to Cloud Storage, indicating a request to store new policy documents.
- **Upload Success Response:** Data flow from Cloud Storage to Insert New Document, confirming that documents have been successfully uploaded.
- **Convert to Embeddings:** Data flow from Convert to Embeddings to Update Knowledge Base, providing the newly created embeddings ready for integration into the system's knowledge base.
- **Knowledge Base Updated:** Data flow from Convert to Embeddings to the larger system, indicating that the knowledge base has been updated with new information derived from the newly uploaded documents.



b. Entity Relationship Diagram



This diagram is an Entity-Relationship Diagram (ERD) commonly used in system analysis and database design to illustrate the relationships between data entities within a system. It depicts five primary entities: Employees, Tickets, Policy Documents, Cloud Storage Documents, and Embeddings.

Attributes for each entity:

Employees: **Employee ID**, Employee Name, Employee Designation, Employee SSN, Employee Email, Employee Department. These attributes provide detailed information about each employee.

Tickets: **Ticket ID**, ***Ticket Assigned To***, ***Ticket Created By***, Ticket Type, Ticket Solution, Ticket Solution Reference, Ticket Status, Ticket Description. These attributes define the specifics of issues or requests handled within the system.

Policy Documents: Policy ID, Policy Path, Policy Description, Policy Department, Policy Created On, Policy Modified On, Policy Name. This entity stores metadata about policy documents relevant to the system.

Cloud Storage Documents: **Document ID**, ***Policy ID***, Document, Date Uploaded On, Date Modified On, LLM Trained Status. This entity reflects documents stored in cloud storage, detailing their upload and modification dates, along with their training status.

Embeddings: **Vector ID**, ***Document ID***, Embedding Vector, Time Stamp, Tags, Usage Matrix, Model ID, Quality Score. This entity stores information about the embeddings generated from documents, used for machine learning models and search functionalities.

Cardinality between the entities:

Employees to Tickets:

"Creates" and "Responsible For": One-to-Many relationships where one employee can create and be responsible for multiple tickets, while each ticket is created and managed by only one employee.

"Assigned to": Many-to-One relationship where multiple tickets can be assigned to one employee, representing the employee responsible for resolving the tickets.

Tickets to Policy Documents:

"Uses": Many-to-One relationship where multiple tickets can use several policy documents as references for resolution or clarification.

"Explains": One-to-Many relationship where one policy document can explain multiple tickets, providing necessary details or guidelines relevant to the issues described in the tickets.

Embeddings to Cloud Storage Documents:

"Generated from": One-to-One relationship indicating that each embedding is generated from a specific document stored in cloud storage.

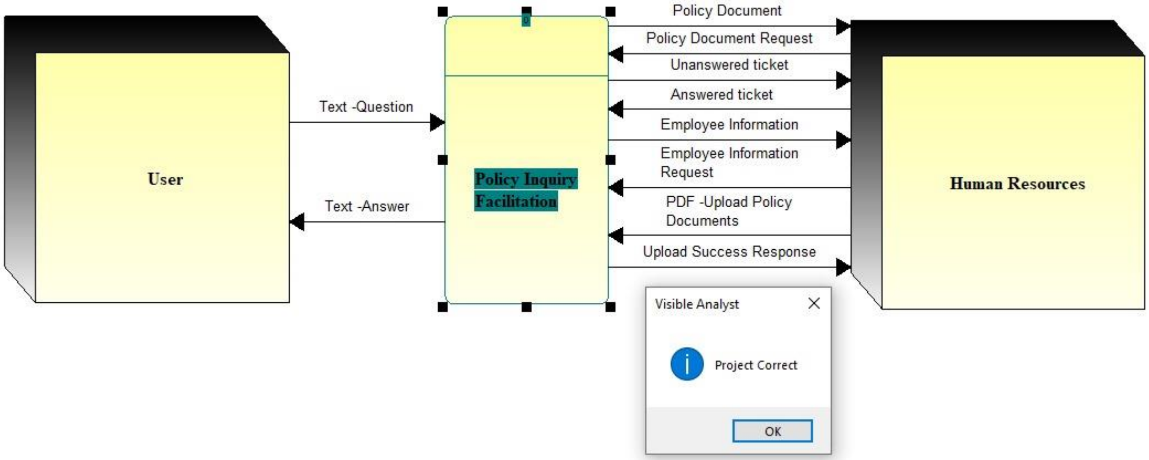
"Generates": One-to-Many relationship from Cloud Storage Documents to Embeddings, showing that multiple embeddings can be generated from documents stored within cloud storage.

Policy Documents to Cloud Storage Documents:

one-to-one relationship. This indicates that each Policy Document corresponds to a unique Cloud Storage Document and vice versa. This relationship suggests that each policy document is uniquely stored as a cloud document, ensuring a one-to-one mapping between these entities.

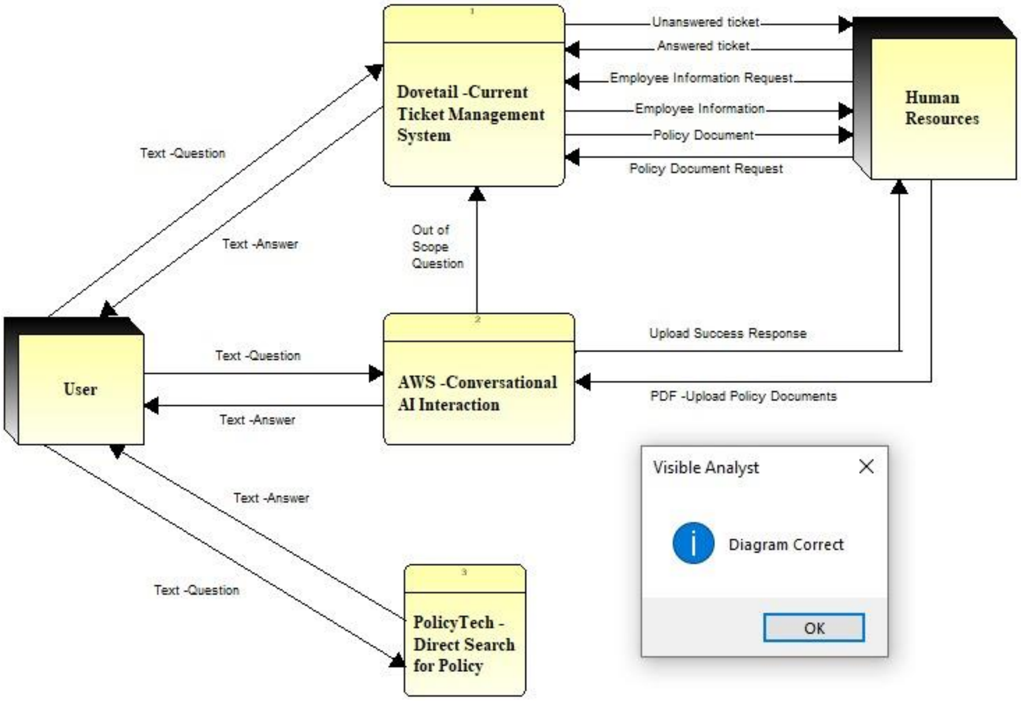
This ERD efficiently maps out the intricate relationships and data flows within the organization, highlighting how employees interact with tickets and documents, and how documents are processed into embeddings for further use in the system’s operations. This setup supports efficient data management and retrieval, essential for operational success in document-intensive environments.

Error reports of all the DFD's (Logical)



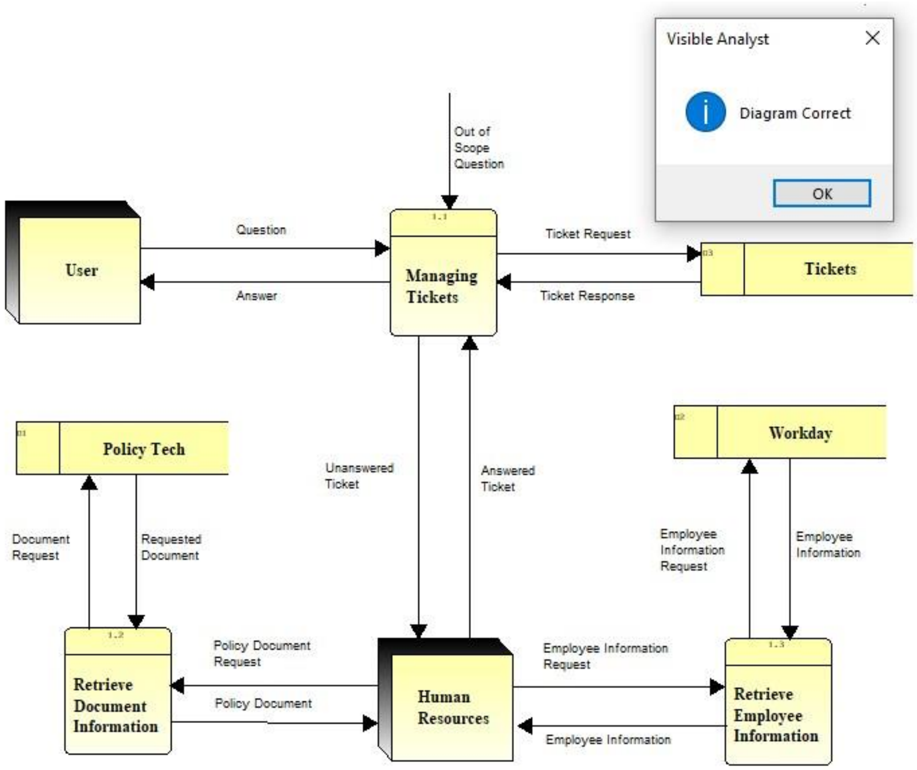
Visible Analyst

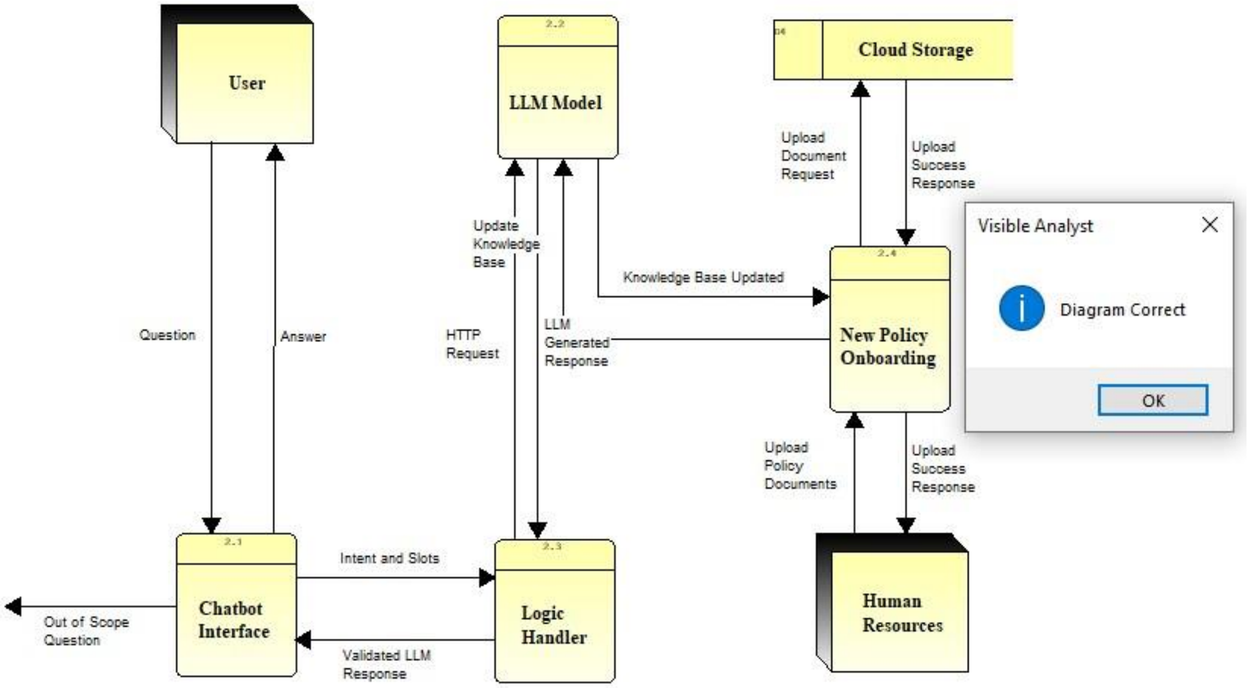
Project Correct

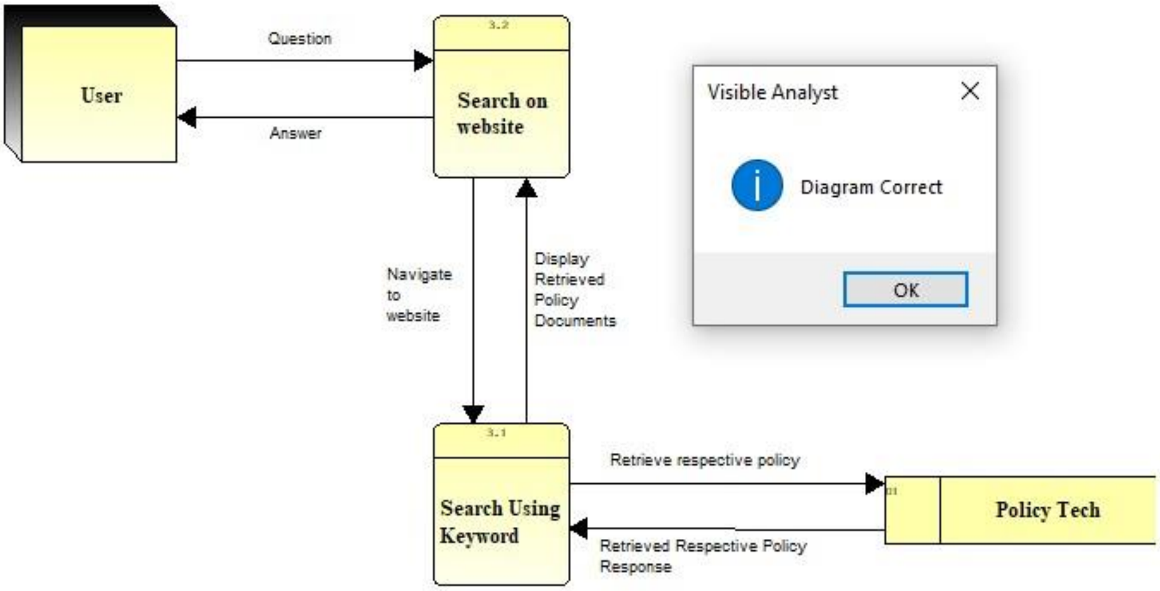


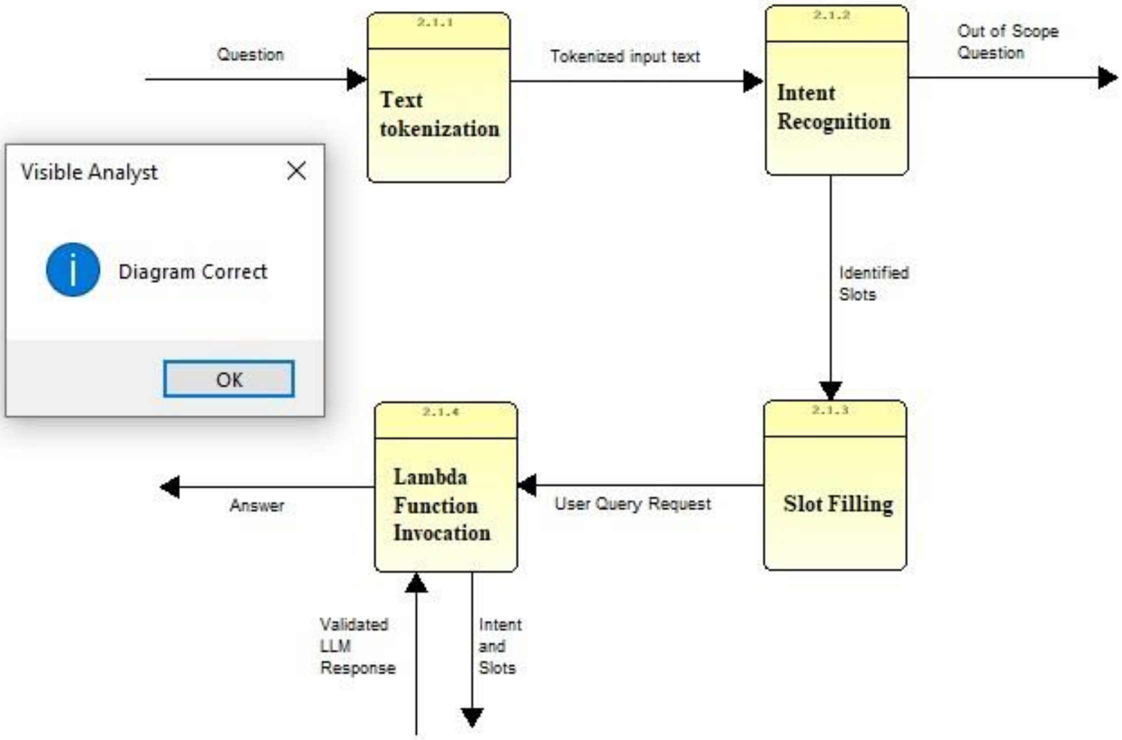
Visible Analyst

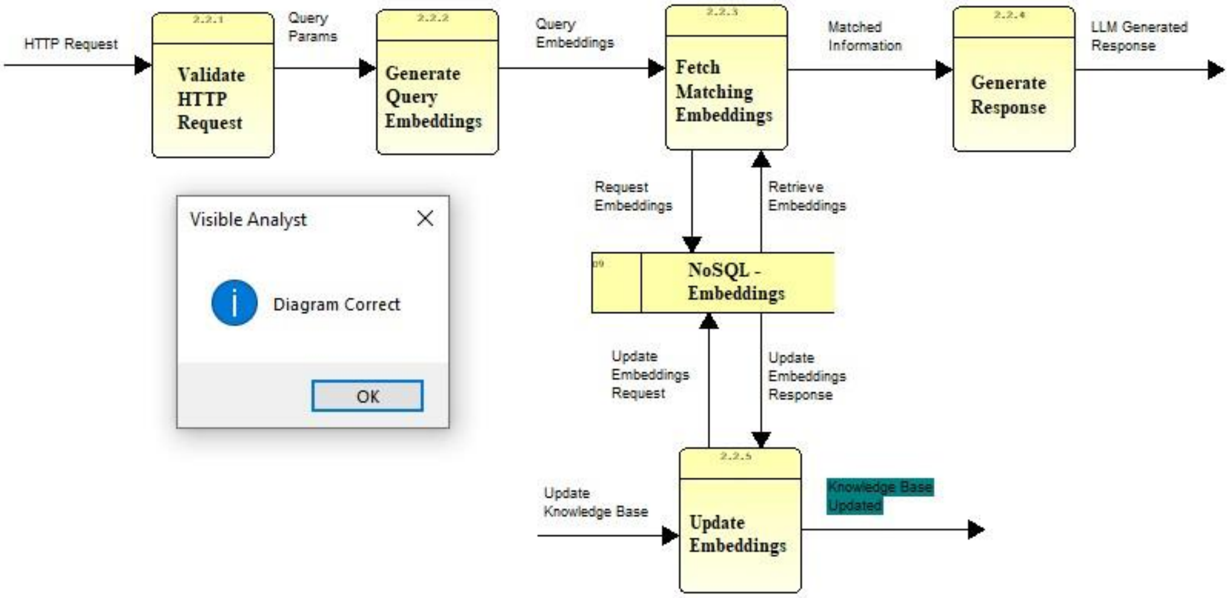
Diagram Correct

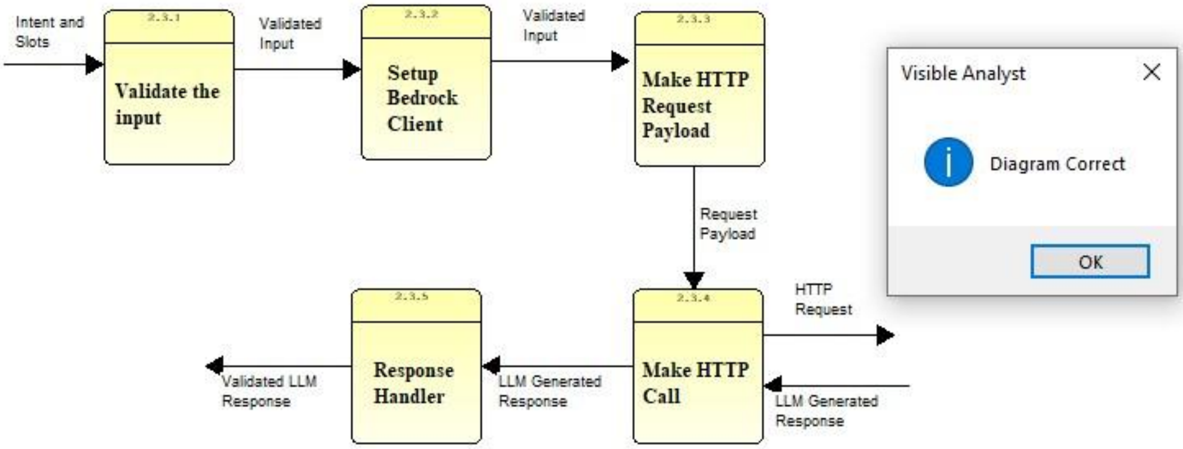

Visible Analyst
Diagram Correct
OK
Out of Scope Question
1.1
Managing Tickets
User
Question
Answer
Ticket Request
Ticket Response
D3
Tickets
D1
Policy Tech
D2
Workday
Unanswered Ticket
Answered Ticket
Document Request
Requested Document
Employee Information Request
Employee Information
1.2
Retrieve Document Information
Policy Document Request
Policy Document
Human Resources
Employee Information Request
Employee Information
1.3
Retrieve Employee Information

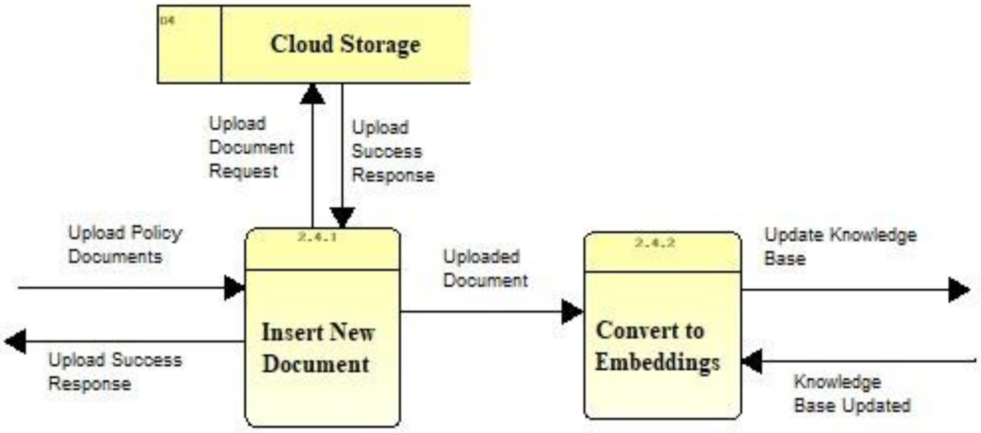

User
2.2
LLM Model
D4
Cloud Storage
Upload Document Request
Upload Success Response
Visible Analyst
Diagram Correct
OK
Update Knowledge Base
Knowledge Base Updated
Question
Answer
HTTP Request
LLM Generated Response
2.4
New Policy Onboarding
Upload Policy Documents
Upload Success Response
2.1
Chatbot Interface
Intent and Slots
2.3
Logic Handler
Out of Scope Question
Validated LLM Response
Human Resources


3.2
User
Question
Answer
Search on website
Visible Analyst
Diagram Correct
OK
Navigate to website
Display Retrieved Policy Documents
3.1
Search Using Keyword
Retrieve respective policy
Retrieved Respective Policy Response
D1
Policy Tech


Question
2.1.1
Text tokenization
Tokenized input text
2.1.2
Intent Recognition
Out of Scope Question
Visible Analyst
Diagram Correct
OK
Identified Slots
2.1.4
Lambda Function Invocation
2.1.3
Slot Filling
Answer
User Query Request
Validated LLM Response
Intent and Slots


HTTP Request
2.2.1
Validate HTTP Request
Query Params
2.2.2
Generate Query Embeddings
Query Embeddings
2.2.3
Fetch Matching Embeddings
Matched Information
2.2.4
Generate Response
LLM Generated Response
Request Embeddings
Retrieve Embeddings
NoSQL - Embeddings
Update Embeddings Request
Update Embeddings Response
2.2.5
Update Embeddings
Update Knowledge Base
Knowledge Base Updated
Visible Analyst
Diagram Correct
OK


Intent and Slots
2.3.1
Validate the input
Validated Input
2.3.2
Setup Bedrock Client
Validated Input
2.3.3
Make HTTP Request Payload
Request Payload
2.3.4
Make HTTP Call
HTTP Request
LLM Generated Response
LLM Generated Response
2.3.5
Response Handler
Validated LLM Response
Visible Analyst
Diagram Correct
OK



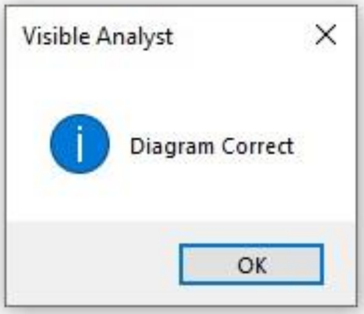
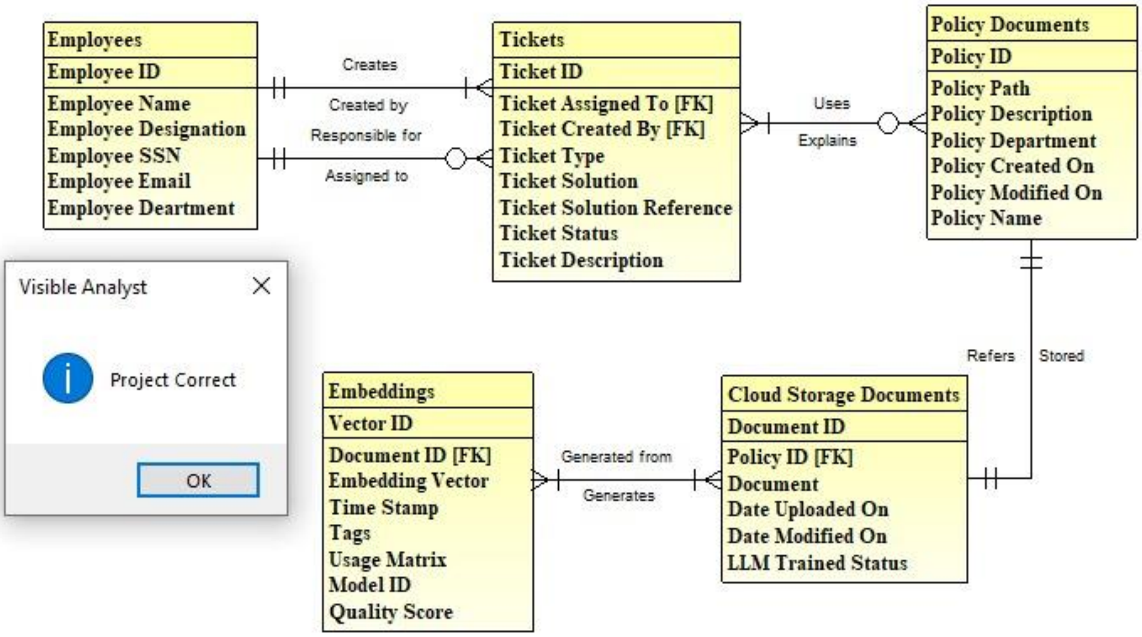

Visible Analyst

Diagram Correct



Visible Analyst

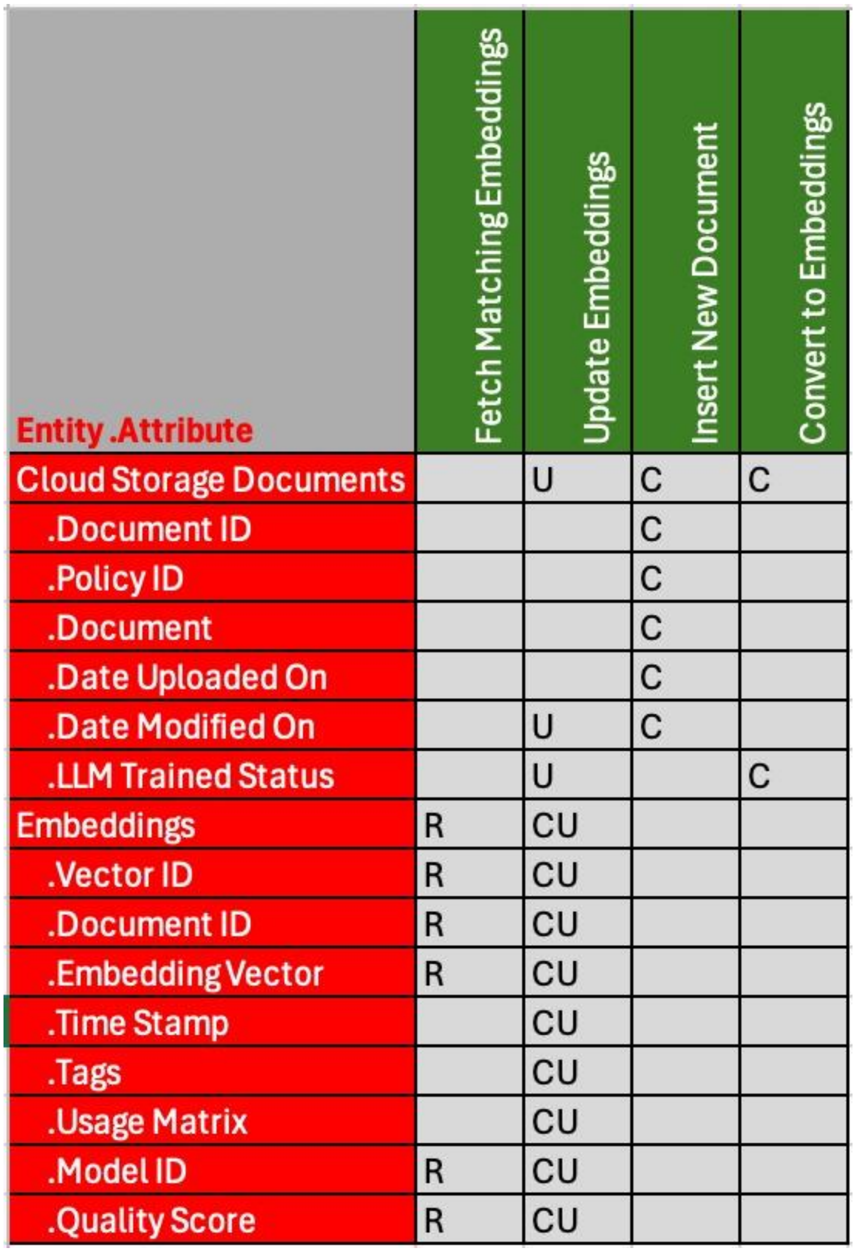
Project Correct

c. Synchronized System Model (CRUD Matrix)

The matrix is divided into two main databases: "Cloud Storage Documents" and "Embeddings". Each section lists various attributes like Document ID, Policy ID, Date Uploaded On, Vector ID, Embedding Vector, etc.

Each column of the CRUD matrix corresponds to these operations, showing whether attributes are read (R), created (C), updated (U), or converted/used (CU). This matrix is essential for understanding the lifecycle of documents and embeddings within your database system, detailing how data flows and is manipulated through various operations, crucial for both managing current data processes and planning future database schema modifications or expansions.

This structured approach facilitates effective management and operation of the data processes, ensuring that each component functions optimally within your overall system architecture. It is particularly useful for teams to ensure consistency in how data is handled across different parts of the system and to help new developers understand the system's data operations quickly.



Entity .Attribute	Fetch Matching Embeddings	Update Embeddings	Insert New Document	Convert to Embeddings
Cloud Storage Documents		U	C	C
.Document ID			C	
.Policy ID			C	
.Document			C	
.Date Uploaded On			C	
.Date Modified On		U	C	
.LLM Trained Status		U		C
Embeddings	R	CU		
.Vector ID	R	CU		
.Document ID	R	CU		
.Embedding Vector	R	CU		
.Time Stamp		CU		
.Tags		CU		
.Usage Matrix		CU		
.Model ID	R	CU		
.Quality Score	R	CU		

6. Physical System Design

a. The Candidate System Solutions

Here's a brief description of the AWS services used in the chatbot system:

Amazon Lex:

Role: Serves as the entry point for user queries.

Functionality: Provides speech recognition and NLP capabilities, allowing it to convert voice or text input into structured queries.

Integration: Forwards queries to AWS Lambda or other services for further processing.

AWS Lambda:

Role: Acts as a processing engine, managing the logic flow.

Functionality: Executes functions in response to events, handling communication between Lex and other services.

Integration: Directs queries to AWS Bedrock, OpenSearch, or Amazon S3, and combines the responses.

AWS Bedrock:

Role: Provides NLP capabilities through its models.

Functionality: Houses various pre-trained models like Anthropic Claude V2 and Meta LLaMA 3 8B Amazon Titan, and Mistral AI which generate responses or process queries. We will be using either Claude or Llama as both help in NLP Text processing and as per client requirements.

Integration: Receives queries from Lambda, performs text generation or RAG, and returns the responses.

Model Processing:

- Claude V2:
- Provides comprehensive NLP capabilities, including text generation, understanding, and processing.
- Focuses on safe, responsible AI, ensuring ethical handling of sensitive information.
- Returns responses back to Lambda for further processing or delivery.
- LLaMA 3-8B:
- Excels in instruction-following tasks, generating nuanced and accurate responses for structured queries.

- Handles various NLP tasks, including natural language understanding and text generation.
- Returns responses back to Lambda for further processing or delivery.

Amazon S3:

Role: Stores the enterprise knowledge corpus and documents for embedding.

Functionality: Provides data storage, allowing Sage Maker or other services to retrieve and process information.

Integration: Lambda retrieves documents or data from S3, feeding it into processing pipelines.

Amazon SageMaker:

Role: Facilitates model training, embedding processing, and deployment.

Functionality: Converts documents from S3 into embeddings and supports NLP tasks.

Integration: Can work directly with S3 for data processing, or through Lambda for model-based responses.

Amazon OpenSearch:

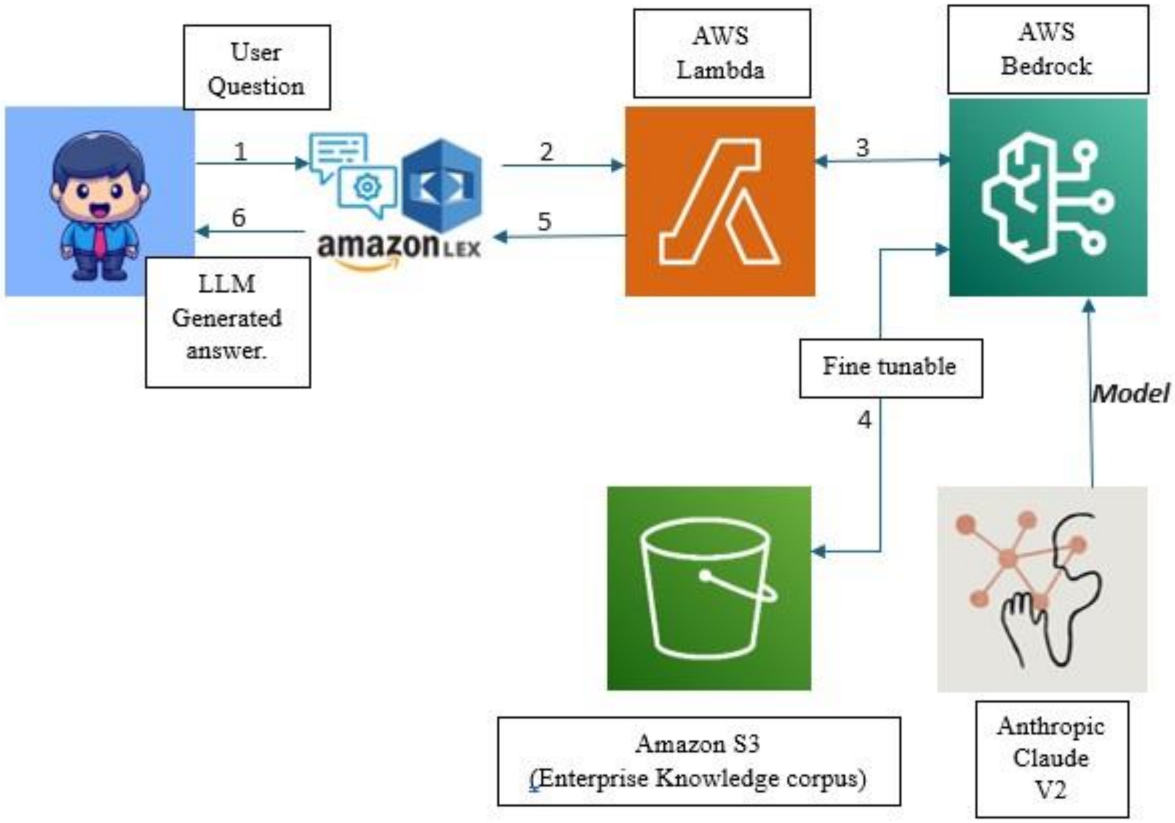
Role: Acts as a vector database, storing and retrieving embeddings.

Functionality: Handles data retrieval for RAG processing, ensuring efficient response generation.

Integration: Communicates with Lambda for embedding retrieval.

Now, we have 3 architectures for our candidate solutions:

1. Bedrock - Claude V2



1. User Interaction:

- **Query Submission:** An employee submits an HR-related query to the chatbot through Amazon Lex, which converts the input into structured data.
- **Lex to Lambda:** Amazon Lex forwards the query to AWS Lambda for processing.

2. Processing:

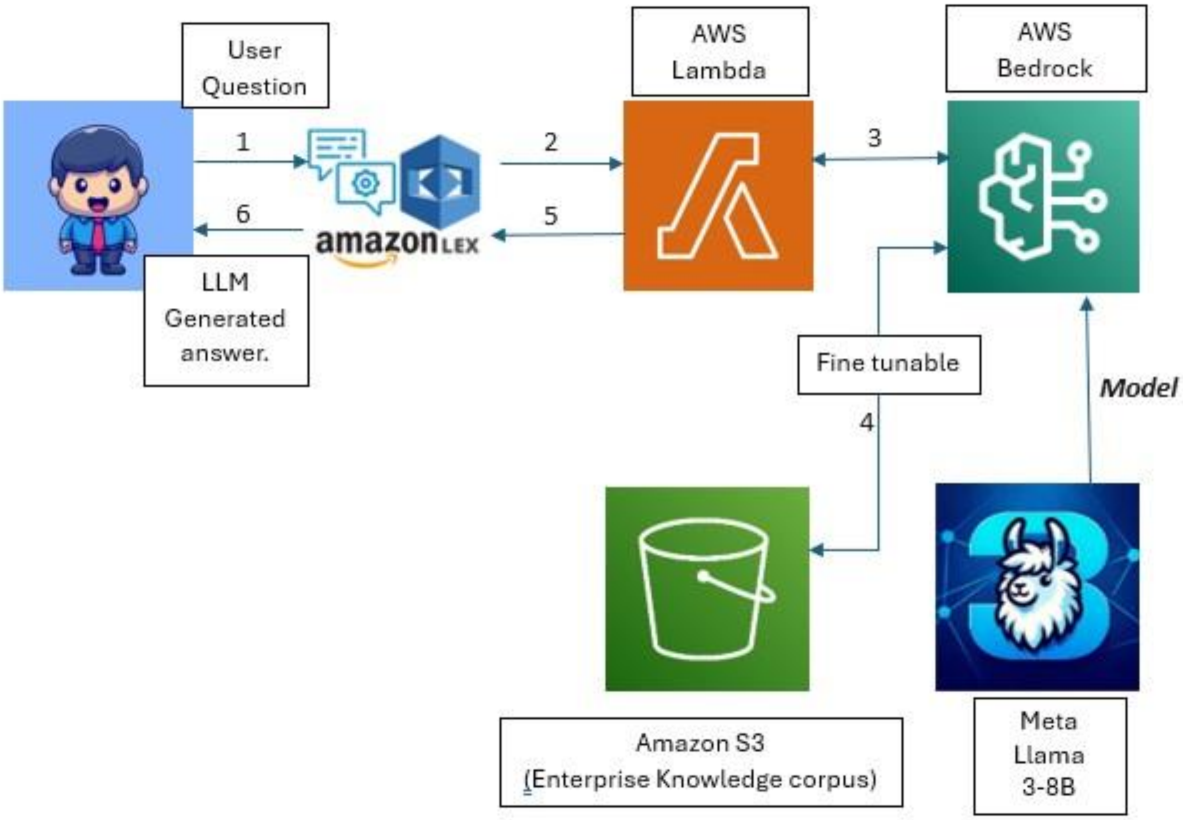
- **Lambda to Bedrock**: Lambda decides which model to use based on the query type using Anthorpic's claude, then routes it to AWS Bedrock.
- **Claude V2:** Handles general NLP tasks, including text generation and response processing.
- **RAG Implementation**: Lambda retrieves relevant information.
- **From Bedrock OpenSearch:** For embeddings or vectors needed for response generation.

- **From S3:** For documents from the enterprise knowledge corpus.

3. Response Generation:

- **Lambda Processing:** Lambda combines the results from Bedrock, OpenSearch, or S3 to generate a comprehensive response.
- **Lex Delivery**: Lambda forwards the response back to Amazon Lex, which delivers it to the employee.

2. Bedrock - Llama 3-8b



1. User Interaction:

- **Query Submission**: An employee submits an HR-related query to the chatbot through Amazon Lex, which converts the input into structured data.
- **Lex to Lambda:** Amazon Lex forwards the query to AWS Lambda for processing.

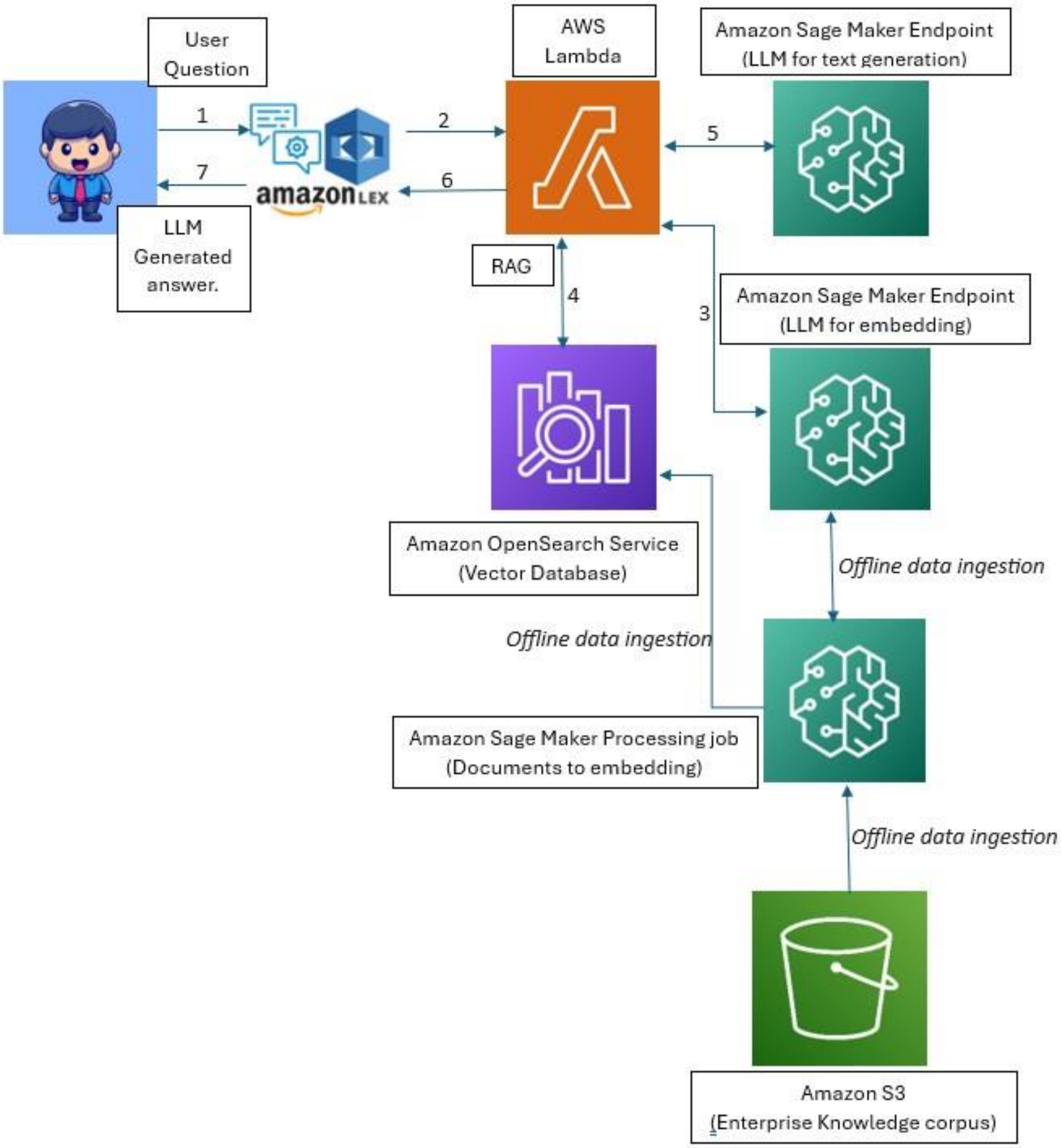
2. Processing:

- **Lambda to Bedrock:** Lambda decides to route the query to AWS Bedrock, utilizing the LLaMA 3-8B model.
- **Instruction-Based Processing**: LLaMA 3-8B handles instruction-following tasks, generating accurate and nuanced responses.
- **RAG Implementation:** Lambda retrieves relevant information.
- **From bedrock to OpenSearch:** For embeddings or vectors needed for response generation.
- **From S3:** For documents from the enterprise knowledge corpus.

3. Response Generation:

- **Lambda Processing:** Lambda combines the results from Bedrock, OpenSearch, or S3 to generate a comprehensive response.
- **Lex Delivery:** Lambda forwards the response back to Amazon Lex, which delivers it to the employee.

3. SageMaker


User Question
1
7
LLM Generated answer.
amazon LEX
2
6
AWS Lambda
5
Amazon Sage Maker Endpoint (LLM for text generation)
RAG
4
3
Amazon Sage Maker Endpoint (LLM for embedding)
Amazon OpenSearch Service (Vector Database)
Offline data ingestion
Offline data ingestion
Amazon Sage Maker Processing job (Documents to embedding)
Offline data ingestion
Amazon S3 (Enterprise Knowledge corpus)

1. User Interaction:

- **Query Submission:** An employee submits an HR-related query to the chatbot through Amazon Lex, which converts the input into structured data.
- **Lex to Lambda:** Amazon Lex forwards the query to AWS Lambda for processing.

2. Processing:

- **Lambda Calls SageMaker:** AWS Lambda decides to route the query to Amazon SageMaker, utilizing its various capabilities:
- **SageMaker Endpoints:** For text generation or embedding processing.
- **SageMaker Processing:** Converts documents from S3 into embeddings.

RAG Implementation:

- **OpenSearch Retrieval:** Lambda can retrieve embeddings or vectors from OpenSearch to aid in response generation.
- **S3 Retrieval:** SageMaker can also access documents from S3 for embedding or text generation.

3. Response Generation:

- **Lambda Processing:** Lambda combines the results from SageMaker, OpenSearch, and S3 to generate a comprehensive response.
- **Lex Delivery:** Lambda forwards the response back to Amazon Lex, which delivers it to the employee.

Additional Options: If not chatbot!

Dovetail Ticketing: Employees can choose to submit a ticket via Dovetail for more direct HR support.

PolicyTech Access: Employees can browse the PolicyTech repository for HR documents and policies.

Candidate System Matrix

Characteristics	**Anthropic Claude V2**	**Meta Llama 3 8B**	**Amazon Sage Maker**
Business Benefits	Provides comprehensive NLP capabilities with a focus on safe, responsible AI.	Excels at instruction-following NLP tasks, providing accurate responses.	Offers NLP capabilities and embedding processing.
Integration	Seamlessly integrates with AWS Bedrock.	Smooth integration via AWS Bedrock.	Integrates with Amazon S3 for data storage and processing.
Scalability	Scales efficiently across Anthropic's cloud.	Scales effectively across AWS infrastructure.	Can scale across AWS infrastructure.
Security	Adheres to responsible AI guidelines, ensuring ethical handling of sensitive data.	Adheres to AWS security protocols, including IAM roles and encryption.	Follows AWS security protocols, including IAM roles and policies.
NLP Capabilities	Provides broad NLP capabilities, including text generation and understanding.	Offers specialized instruction-based NLP, generating nuanced responses.	Covers various NLP tasks, including text generation and embeddings.
Support & Maintenance	Regular updates and monitoring from Anthropic.	Strong support from AWS Bedrock.	Provides updates and monitoring from AWS.

Ease of Customization	Limited customization options.	Offers some flexibility in fine-tuning.	Highly customizable, supporting training and deploying custom models.
Latency	Moderate response time.	Quick responses for instruction-based tasks.	Depends on model and instance choice, generally consistent.
Documentation & Support	Regular updates and support from Anthropic.	Strong documentation and community support.	Comprehensive AWS documentation and monitoring tools.
Monitoring & Logging	Supports basic logging and monitoring.	Limited in-built tools but can integrate third-party solutions.	AWS CloudWatch and other AWS tools support monitoring.
Healthcare Compliance	Designed with compliance considerations.	Can comply with relevant regulations.	Can be set up to comply with healthcare regulations.
User Interface & Experience	Provides consistent responses, and easy integration with chat interfaces.	Delivers structured responses, suitable for employee queries.	Depending on implementation can offer varied interfaces.
Cost Structure	Flexible pricing aligned with Anthropic cloud offerings.	Competitive pricing, scales with usage.	Depending on instance type and usage, offers scalable options.

1. Claude V2: Focuses on safe, responsible AI, providing comprehensive NLP capabilities and integration.

2. Llama 3 8B: Excels in instruction-following tasks, offering accurate and nuanced responses.
3. Sage Maker: Provides NLP and embedding capabilities, integrating seamlessly with other AWS services.

b. The Feasibility Analysis Matrix (Rationale for weights)

This Feasibility Analysis Matrix evaluates three leading AI technologies—Anthropic Claude V2, Meta LLaMA 3-8B, and Amazon SageMaker—against key criteria crucial to the successful deployment of our Conversational AI system at Fresenius Medical Care. These criteria include Operational Feasibility, Technical Feasibility, Economic Feasibility, and Schedule Feasibility, each weighed according to their impact on our strategic objectives.

Operational Feasibility (30% weight):

This criterion assesses how well each technology performs various natural language processing (NLP) tasks which are essential for our HR conversational AI. Meta LLaMA 3-8B leads with a score of 85, showcasing its superior ability in handling precise, instruction-following tasks, making it highly suitable for our needs. Claude V2 scores 80 due to its comprehensive but slightly less advanced capabilities, while SageMaker trails at 70, hindered by its rigidity in deployment options.

Technical Feasibility (30% weight):

We evaluate how well these solutions integrate with our existing AWS Bedrock infrastructure and overall system architecture. Meta LLaMA 3-8B scores the highest (85) for its seamless integration in complex environments. Claude V2, while robust, follows closely with a score of 80, and SageMaker receives a 70, facing challenges with integration compared to its counterparts.

Economic Feasibility (30% weight):

This measures cost-effectiveness, considering the pricing models and scalability. Meta LLaMA 3-8B, with its scalable pricing model, rates the best at 85, suggesting cost savings as usage increases. Claude V2 offers flexible pricing but less cost efficiency, scoring 75. SageMaker's variable costs, potentially the highest, earn it a score of 65.

Schedule Feasibility (10% weight):

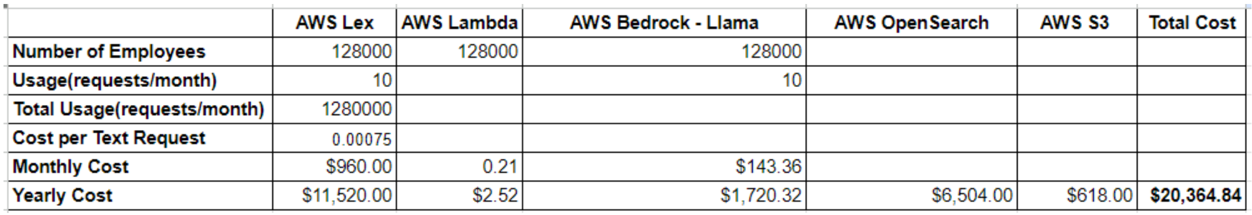
Time to deployment is critical. Meta LLaMA 3-8B, with its efficiency and quick integration, scores an 85. Claude V2 also demonstrates rapid deployment capabilities, scoring 80. However, SageMaker, requiring extensive setup for custom model training, receives the lowest score of 60.

Feasibility Criteria	**Weight**	**Anthropic Claude V2**	**Meta LLaMA 3 8B**	**Amazon SageMaker**
Operational Feasibility	30%	Provides comprehensive NLP capabilities for various tasks, slightly behind the leading edge. Score: 80	Offers strong NLP capabilities, excelling at instruction-following tasks, slightly better performance. Score: 85	Supports NLP tasks and embedding processing, less flexible in deployment. Score: 70
Technical Feasibility	30%	Seamless integration with AWS Bedrock, ensuring responsible AI handling, strong integration capability. Score: 80	Integrates smoothly into the chatbot architecture via Bedrock, excellent fit for complex systems. Score: 85	Integrates well with AWS services, good support for model training and deployment but less integrated than competitors. Score: 70
Economic Feasibility	30%	Flexible pricing aligned with Anthropic cloud offerings, suitable for varied budget ranges. NPV: -$18,714.50 BEP: 6.46 years Score: 30	Competitive pricing model, scales with usage, offering cost efficiency as usage increases. NPV: $63,288.18 BEP: 1.11 years Score: 85	Variable costs depending on instance type and usage, which might be higher than competitors. NPV: -$31,129.85 BEP: 5.92 years Score: 35

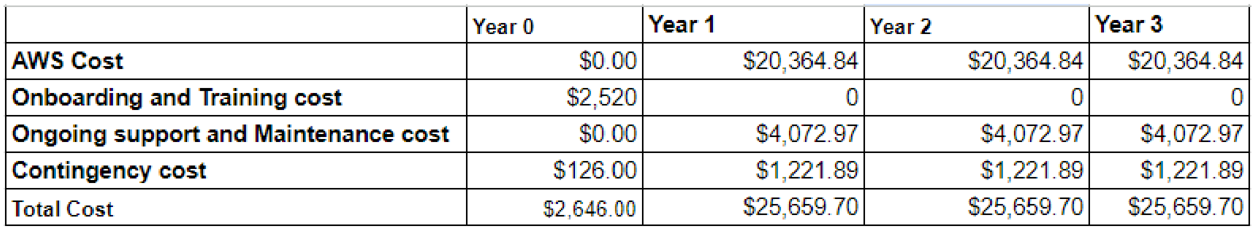
Schedule Feasibility	10%	Provides immediate access to NLP capabilities, supporting efficient processing, quick to deploy. Score: 80	Offers quick response time for instruction-based tasks, leading to efficient workflow. Score: 85	Variable implementation time, depending on model training and setup, which can delay project timelines. Score: 60
Overall Score	100%	Total Score: 65	Total Score: 85	Total Score: 58.5

The matrix highlights Meta LLaMA 3-8B as the leading candidate with an overall score of 85, proving its efficiency and alignment with our strategic goals for a streamlined, responsive, and economical HR conversational AI system. This evaluation allows us to proceed with confidence, knowing that Meta LLaMA 3-8B offers the best blend of performance, integration, cost-effectiveness, and deployment readiness, which are essential for enhancing HR operations and employee engagement at Fresenius Medical Care.

Economic feasibility for bedrock Llama 3-8B



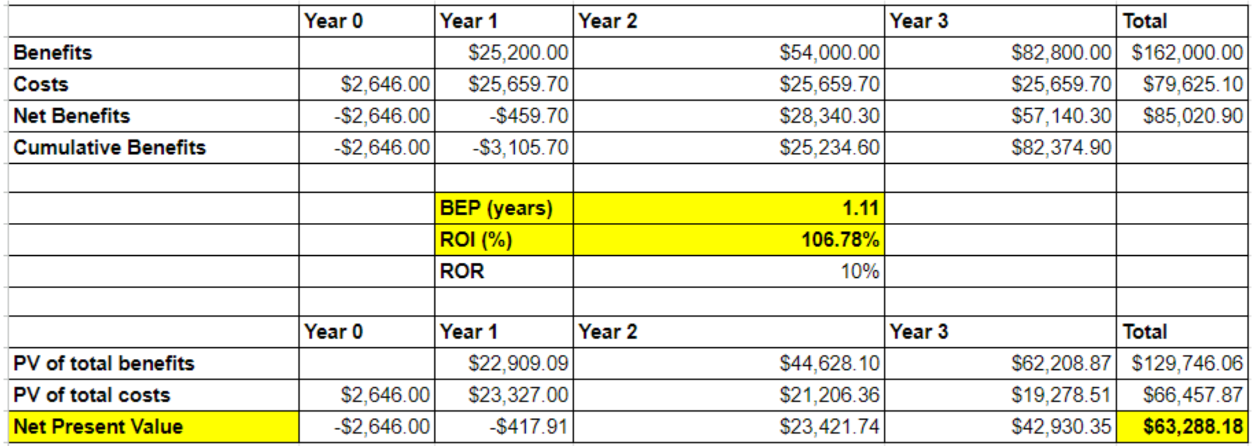
	AWS Lex	AWS Lambda	AWS Bedrock - Llama	AWS OpenSearch	AWS S3	Total Cost
Number of Employees	128000	128000	128000			
Usage(requests/month)	10		10			
Total Usage(requests/month)	1280000					
Cost per Text Request	0.00075					
Monthly Cost	$960.00	0.21	$143.36			
Yearly Cost	$11,520.00	$2.52	$1,720.32	$6,504.00	$618.00	**$20,364.84**



	Year 0	Year 1	Year 2	Year 3
AWS Cost	$0.00	$20,364.84	$20,364.84	$20,364.84
Onboarding and Training cost	$2,520	0	0	0
Ongoing support and Maintenance cost	$0.00	$4,072.97	$4,072.97	$4,072.97
Contingency cost	$126.00	$1,221.89	$1,221.89	$1,221.89
Total Cost	$2,646.00	$25,659.70	$25,659.70	$25,659.70

Assuming we save 25 hours a month and increase 5 hours on a consecutive and average hourly wage for HR is $40.

Cost Benefit Analysis

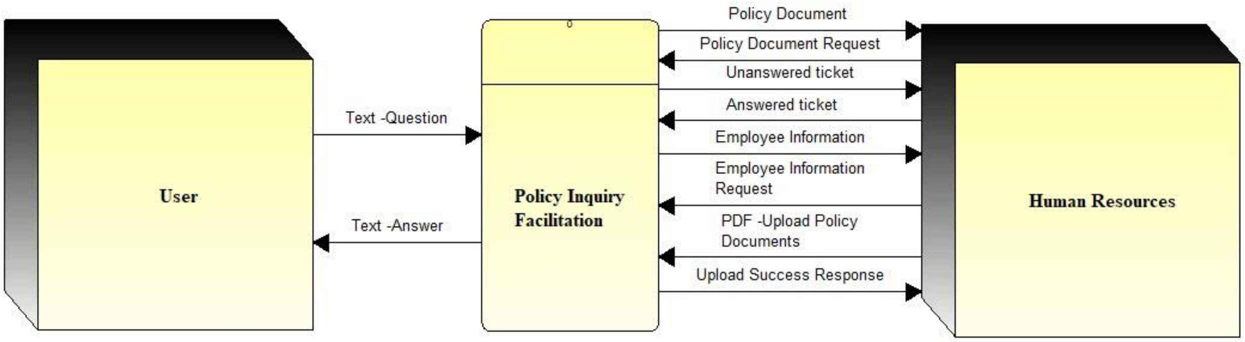


	Year 0	Year 1	Year 2	Year 3	Total
Benefits		$25,200.00	$54,000.00	$82,800.00	$162,000.00
Costs	$2,646.00	$25,659.70	$25,659.70	$25,659.70	$79,625.10
Net Benefits	-$2,646.00	-$459.70	$28,340.30	$57,140.30	$85,020.90
Cumulative Benefits	-$2,646.00	-$3,105.70	$25,234.60	$82,374.90	
		BEP (years)	**1.11**		
		ROI (%)	**106.78%**		
		ROR	10%		
	Year 0	**Year 1**	**Year 2**	**Year 3**	**Total**
PV of total benefits		$22,909.09	$44,628.10	$62,208.87	$129,746.06
PV of total costs	$2,646.00	$23,327.00	$21,206.36	$19,278.51	$66,457.87
Net Present Value	-$2,646.00	-$417.91	$23,421.74	$42,930.35	**$63,288.18**

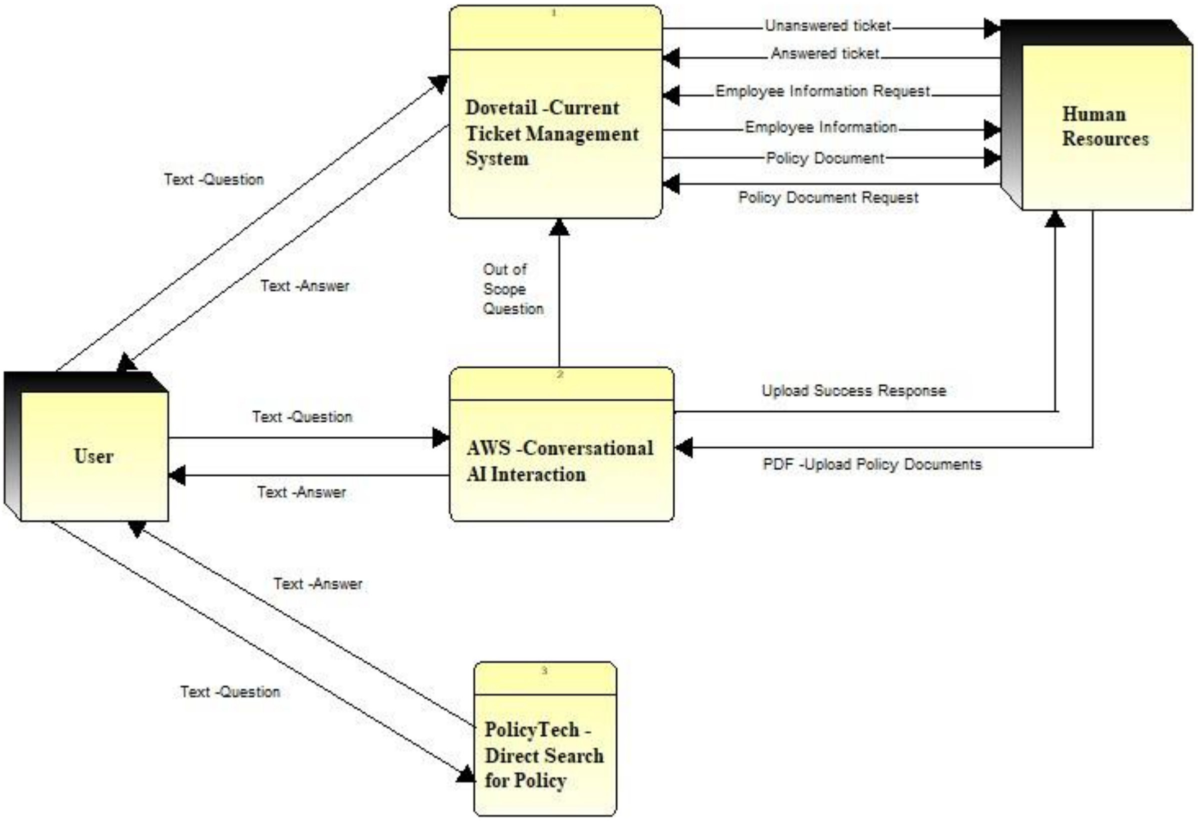
Expected Break-even point- 13 months with net present value being $63,288.

c. The Physical DFD's

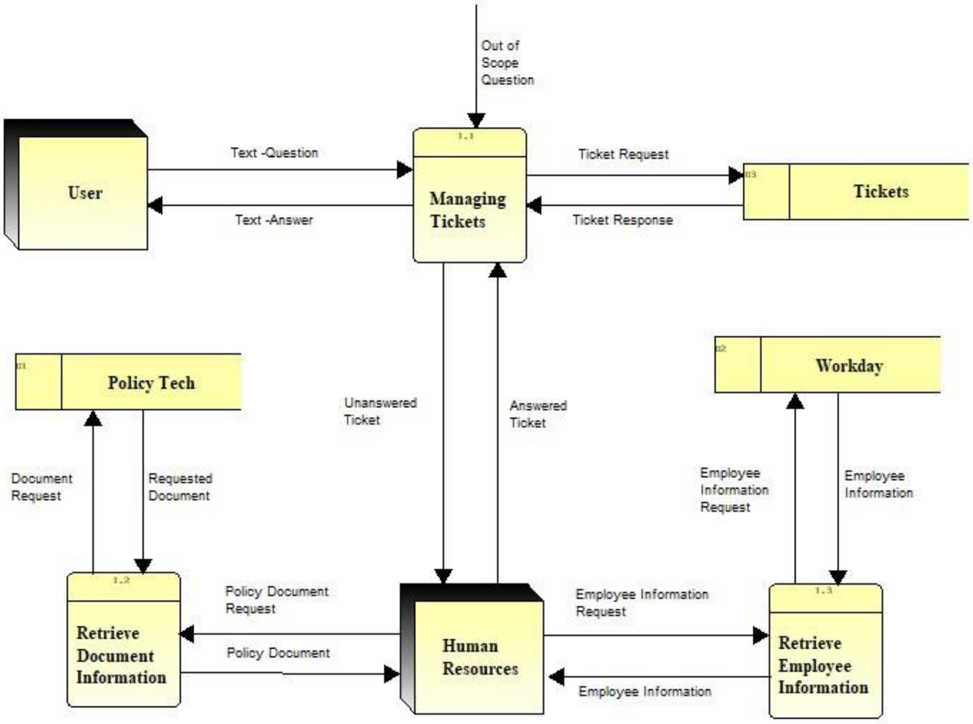
Context Level:



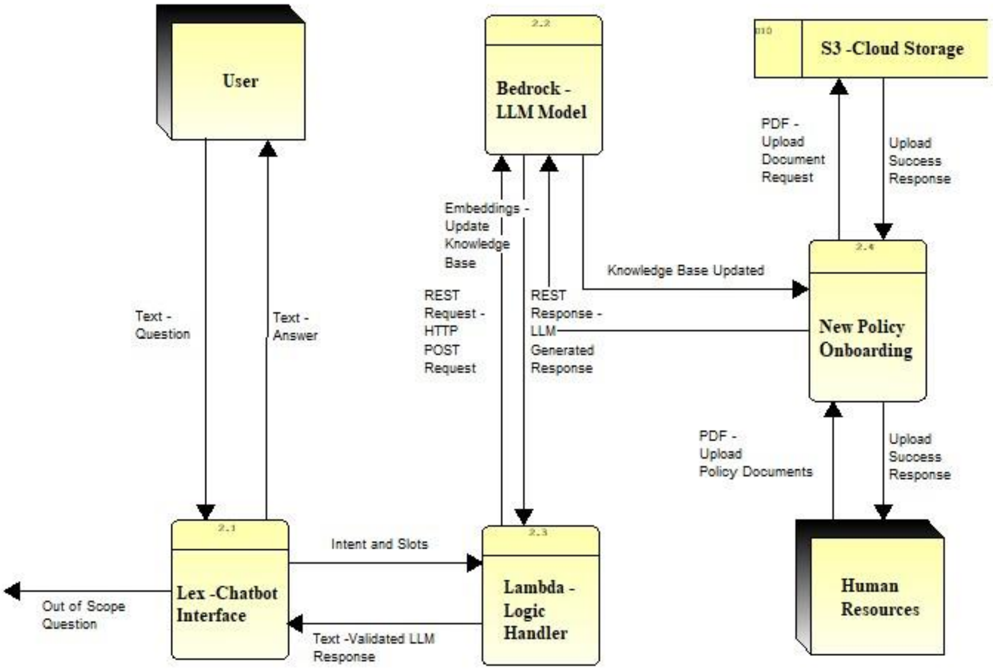
Level 0:



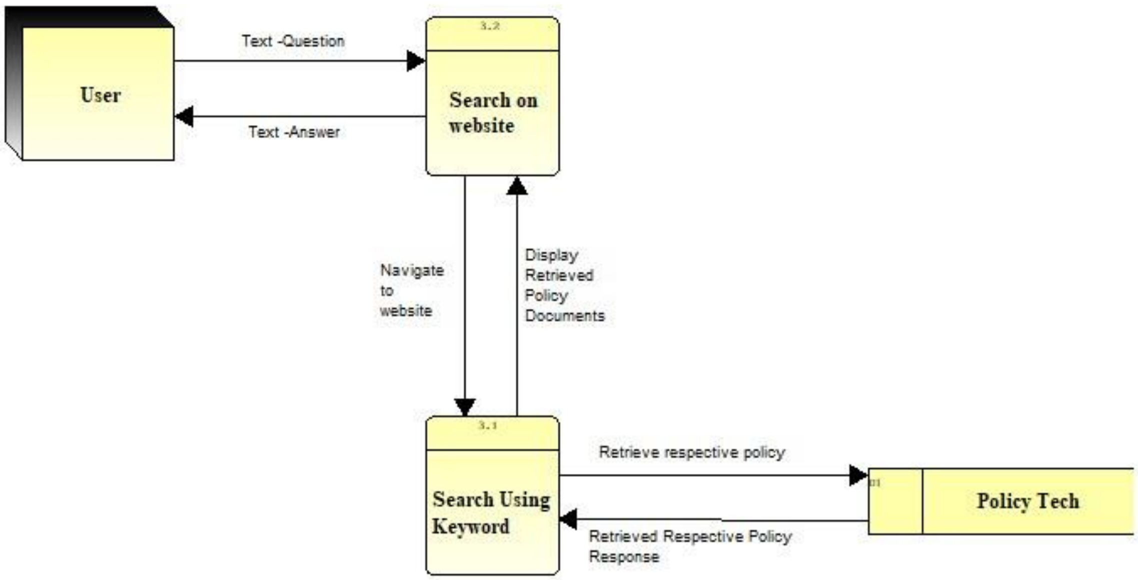
Level 1.1:



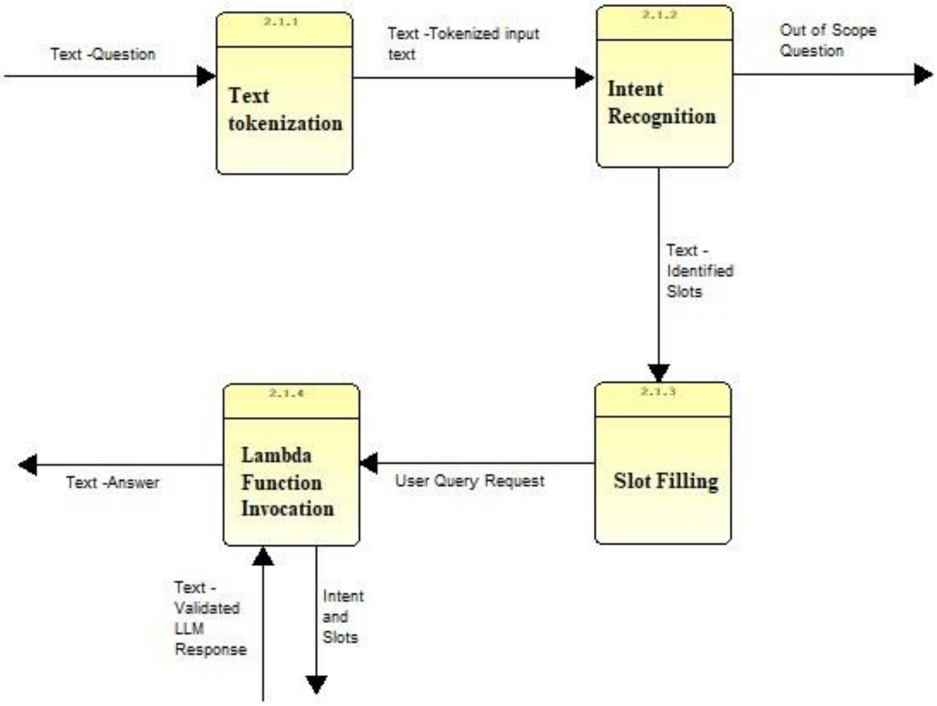
Level 1.2:



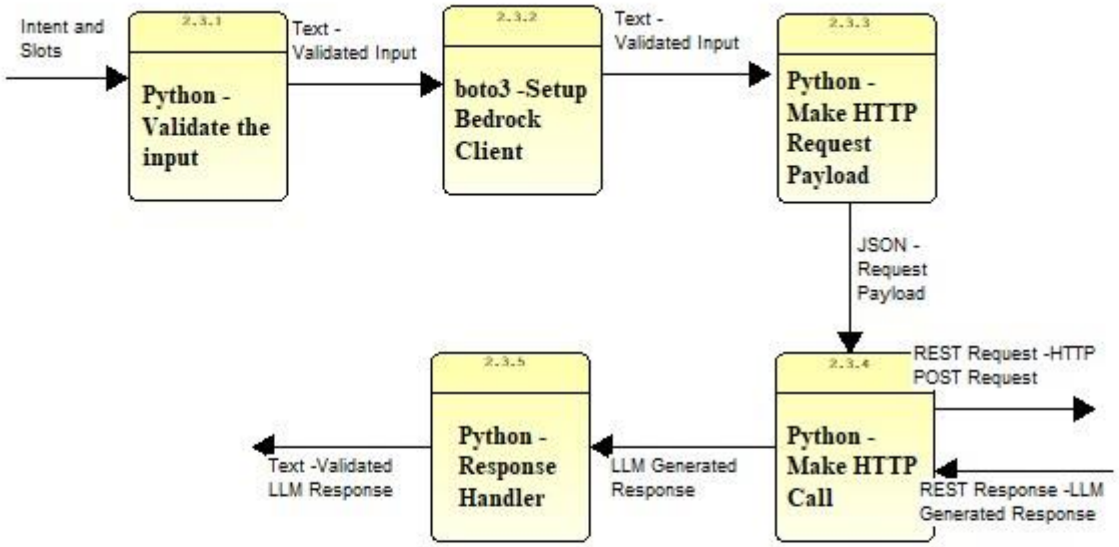
Level 1.3:



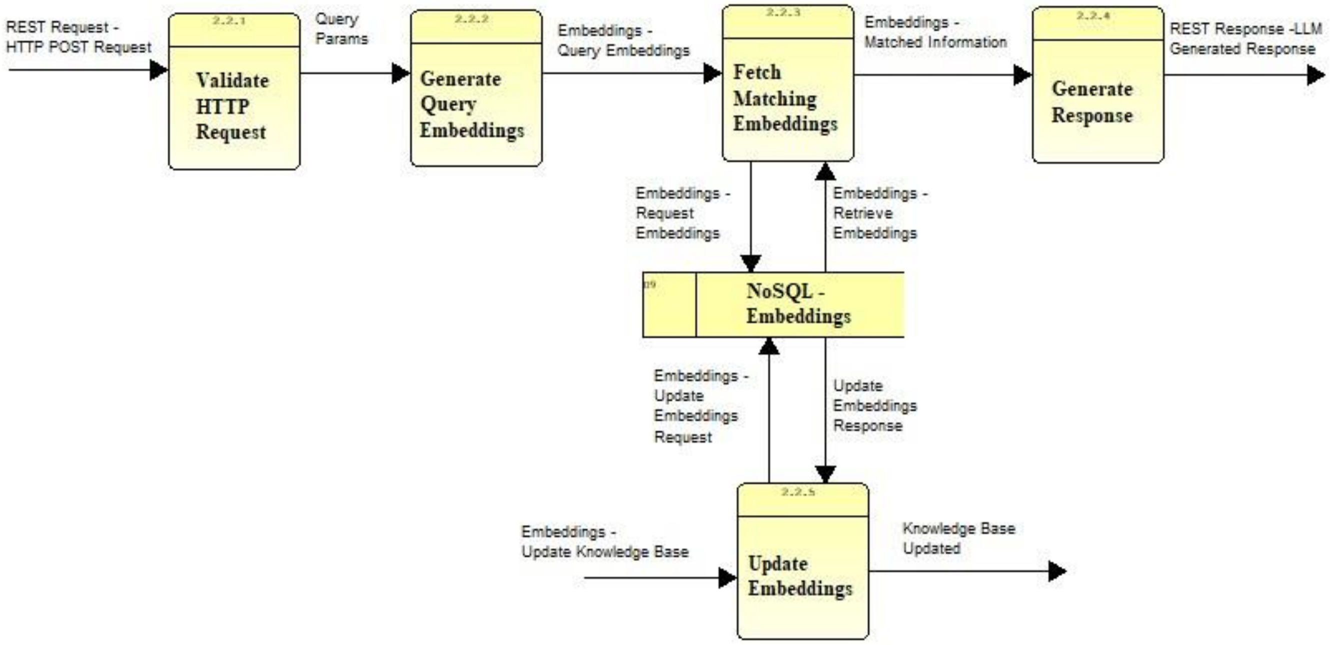
Level 2.1:



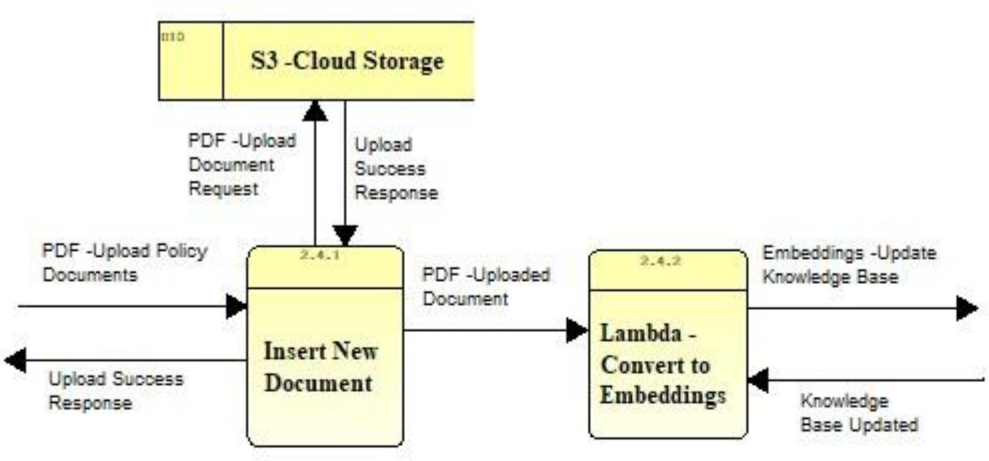
Level 2.3:


Intent and Slots
2.3.1
Python - Validate the input
Text - Validated Input
2.3.2
boto3 -Setup Bedrock Client
Text - Validated Input
2.3.3
Python - Make HTTP Request Payload
JSON - Request Payload
2.3.4
Python - Make HTTP Call
REST Request -HTTP POST Request
REST Response -LLM Generated Response
LLM Generated Response
2.3.5
Python - Response Handler
Text -Validated LLM Response

Level 2.2:


REST Request - HTTP POST Request
2.2.1
Validate HTTP Request
Query Params
2.2.2
Generate Query Embeddings
Embeddings - Query Embeddings
2.2.3
Fetch Matching Embeddings
Embeddings - Matched Information
2.2.4
Generate Response
REST Response -LLM Generated Response
Embeddings - Request Embeddings
Embeddings - Retrieve Embeddings
D9
NoSQL - Embeddings
Embeddings - Update Embeddings Request
Update Embeddings Response
Embeddings - Update Knowledge Base
2.2.5
Update Embeddings
Knowledge Base Updated

Level 2.4:


D10
S3 -Cloud Storage
PDF -Upload Document Request
Upload Success Response
PDF -Upload Policy Documents
Upload Success Response
2.4.1
Insert New Document
PDF -Uploaded Document
2.4.2
Lambda - Convert to Embeddings
Embeddings -Update Knowledge Base
Knowledge Base Updated

7. Input and Output Design

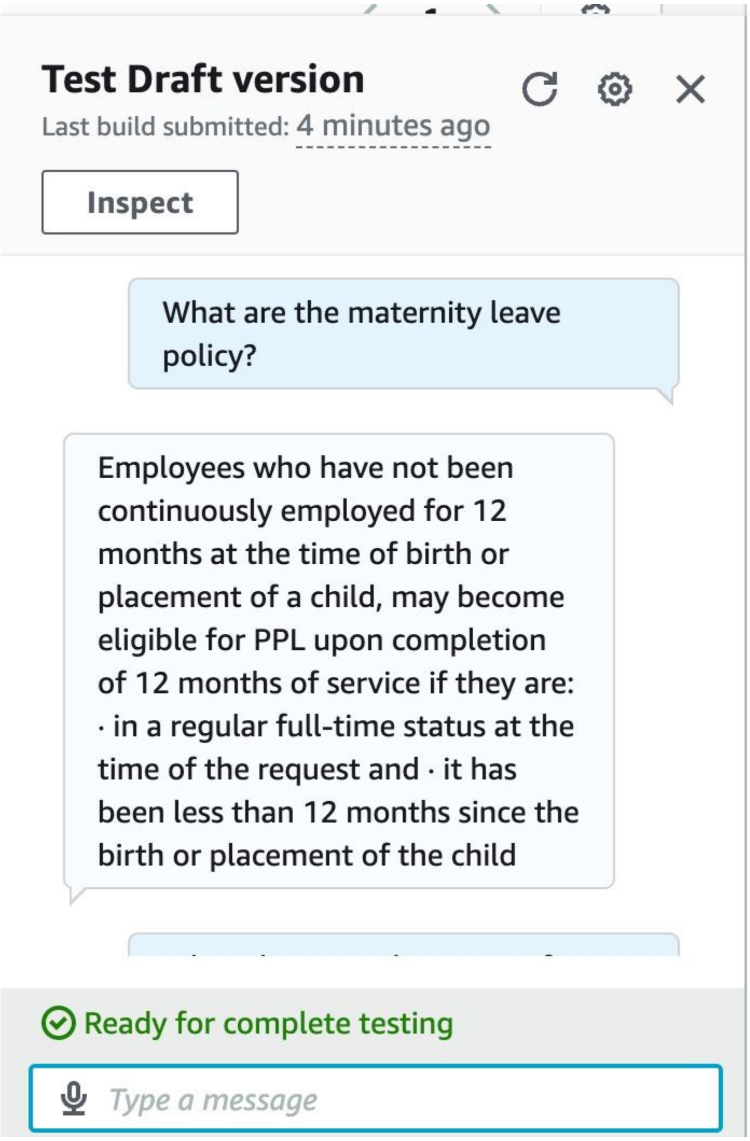
To process and send responses, the chatbot system integrates several AWS services to offer employees solutions to their HR-related questions.

Amazon Lex plays a pivotal role in processing user inputs, seamlessly integrating with other AWS services to generate and deliver comprehensive responses.

Example Responses:

1. **Maternity Leave Policy:**

The chatbot provides detailed information about eligibility for maternity leave and how it applies to employees.



Test Draft version
Last build submitted: 4 minutes ago
Inspect

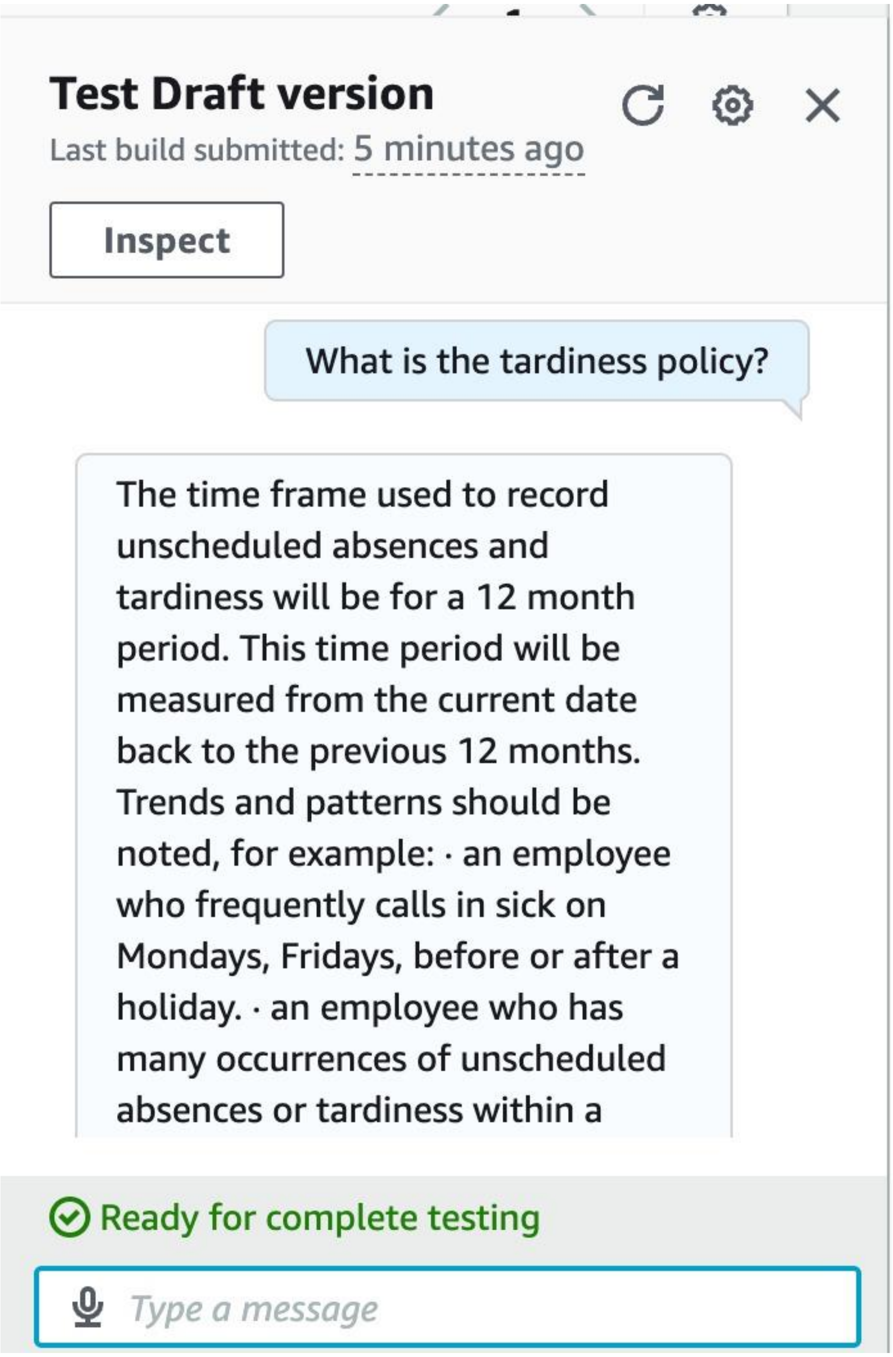
What are the maternity leave policy?

Employees who have not been continuously employed for 12 months at the time of birth or placement of a child, may become eligible for PPL upon completion of 12 months of service if they are: · in a regular full-time status at the time of the request and · it has been less than 12 months since the birth or placement of the child

Ready for complete testing
Type a message

2. Tardiness Policy:

The chatbot outlines the time frame for recording unscheduled absences and tardiness, along with relevant patterns to note.

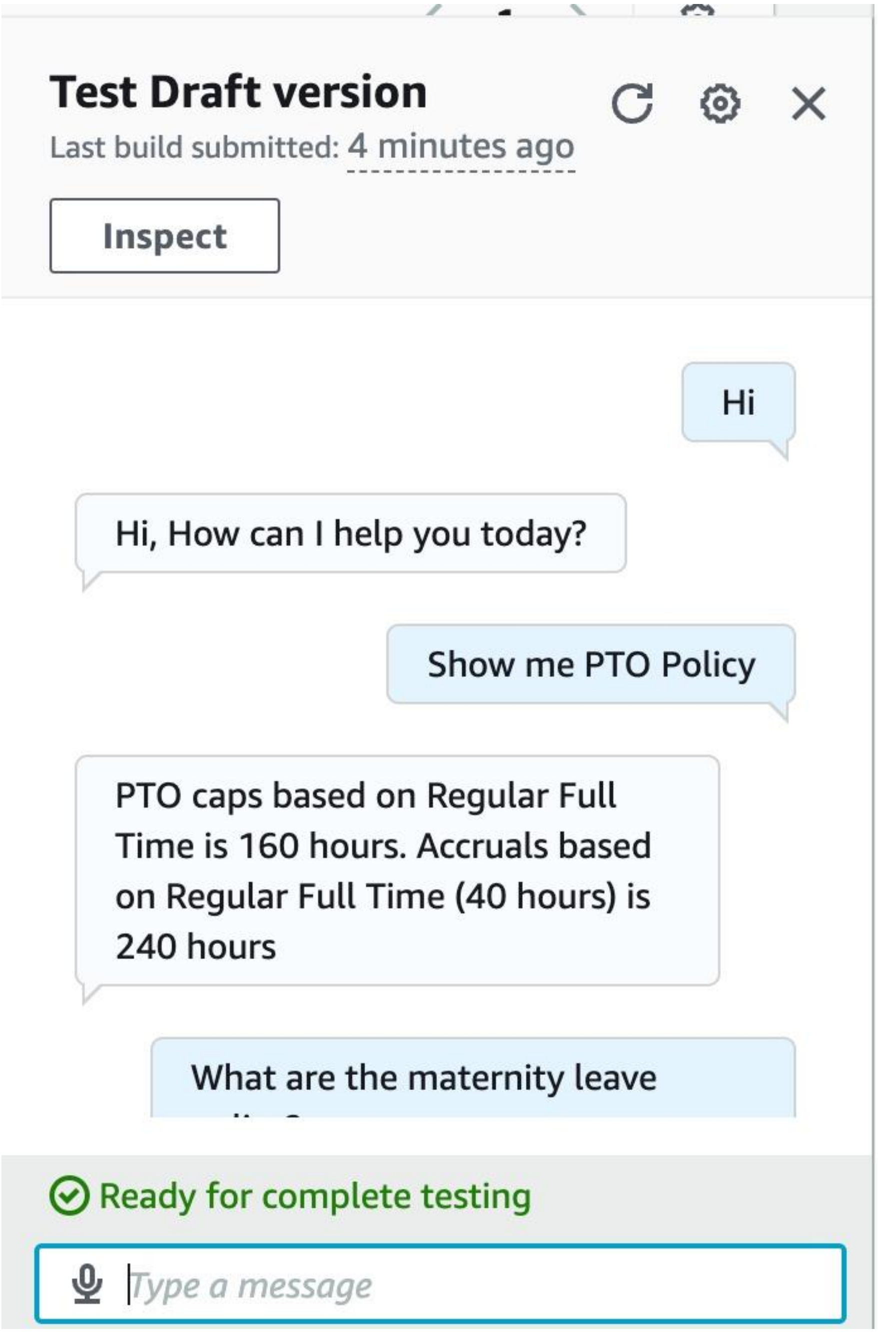


What is the tardiness policy?

The time frame used to record unscheduled absences and tardiness will be for a 12 month period. This time period will be measured from the current date back to the previous 12 months. Trends and patterns should be noted, for example: · an employee who frequently calls in sick on Mondays, Fridays, before or after a holiday. · an employee who has many occurrences of unscheduled absences or tardiness within a

3. PTO Caps:

The chatbot specifies the caps for regular full-time employees and accrual rates based on different working hours.



Test Draft version
Last build submitted: 4 minutes ago
Inspect

Hi

Hi, How can I help you today?

Show me PTO Policy

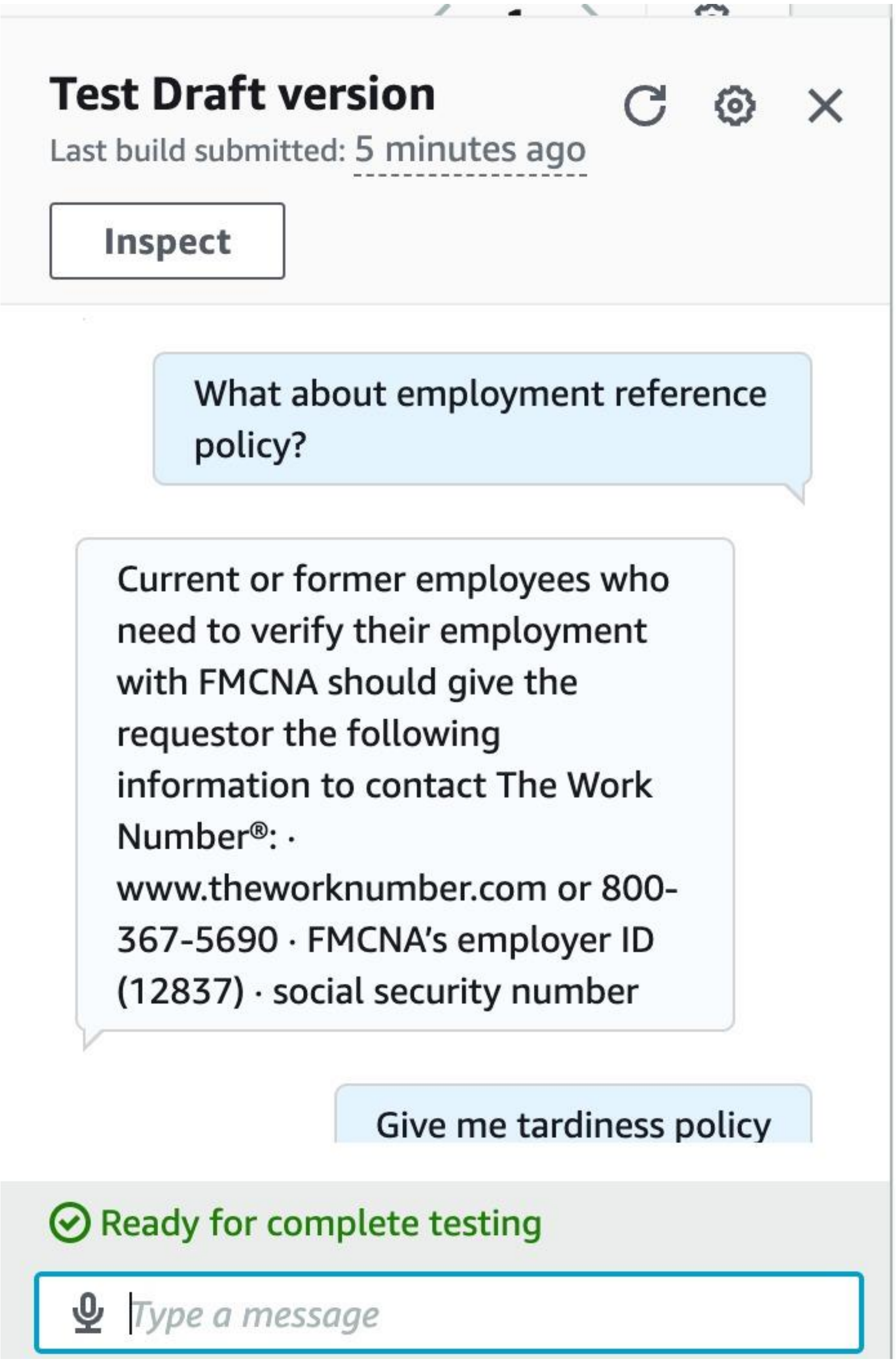
PTO caps based on Regular Full Time is 160 hours. Accruals based on Regular Full Time (40 hours) is 240 hours

What are the maternity leave

Ready for complete testing
Type a message

4. Employment Reference Policy:

The chatbot provides information on verifying employment, including contact details, and required information.



Test Draft version
Last build submitted: 5 minutes ago
Inspect

What about employment reference policy?

Current or former employees who need to verify their employment with FMCNA should give the requestor the following information to contact The Work Number®: · www.theworknumber.com or 800-367-5690 · FMCNA's employer ID (12837) · social security number

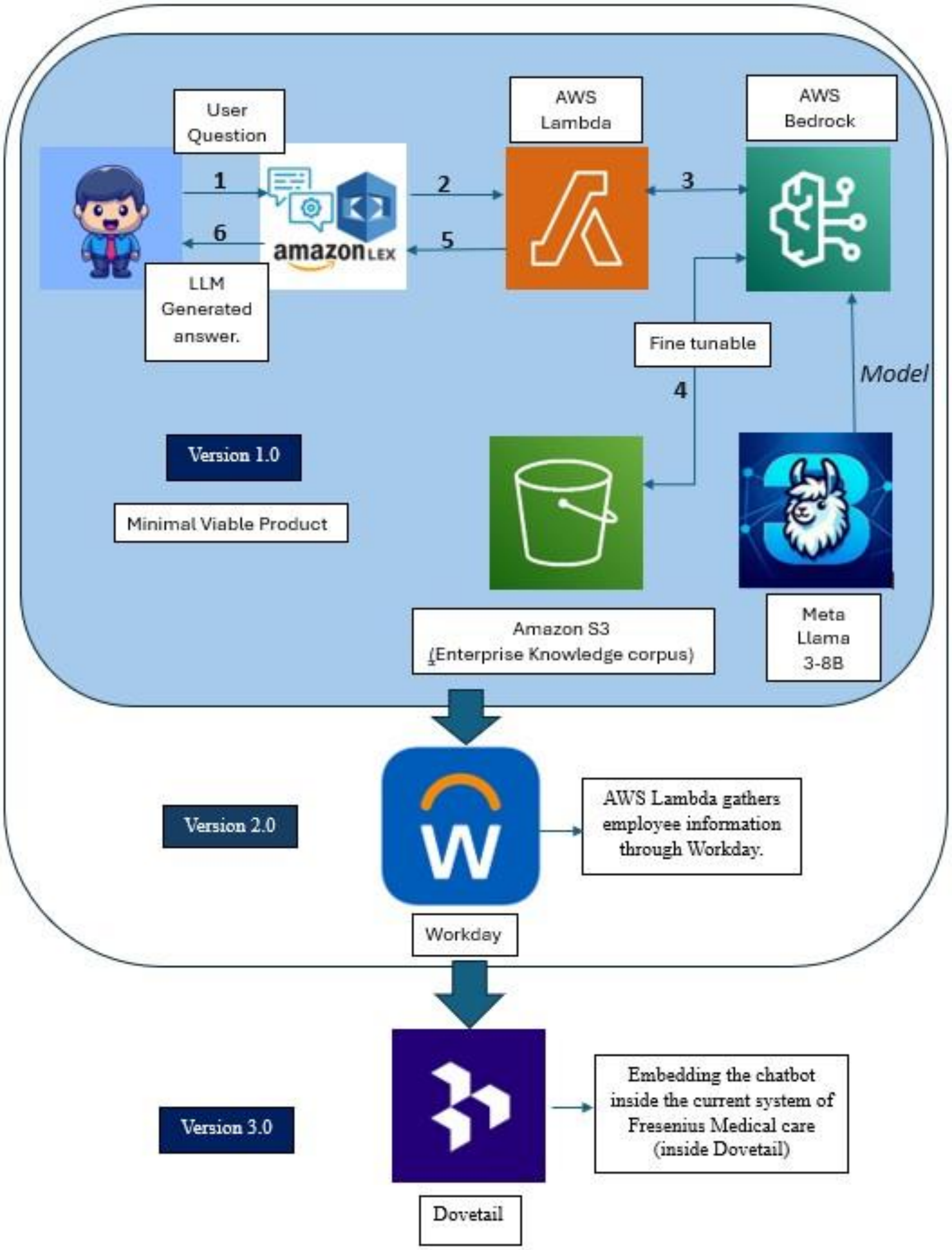
Give me tardiness policy

Ready for complete testing
Type a message

8. Implementation Plan

The client wants our team to implement the chatbot system in three distinct phases, starting with the Minimum Viable Product (MVP) and progressing to more integrated solutions.

Our client wanted us to include the whole process in our architecture, but **the focus will be solely on developing the Minimum Viable Product (MVP).**



Version 1 (MVP):

- Focus: The initial version of the chatbot serves as an MVP, integrating essential functionality.
- Workflow:

  Amazon Lex: Acts as the entry point for user queries, converting them into structured data.

  AWS Lambda: Manages processing logic, handling communication between Lex and other services.

  AWS Bedrock: Provides NLP models like Claude V2 and LLaMA 3-8B for text generation and processing. Support RAG capabilities, storing embeddings and enterprise knowledge corpus.
- Purpose: To offer immediate support for HR queries, ensuring essential functionality is available early.

Version 2 (Integration with Workday):

- Timeline: This version will be developed after MVP, time permitting.
- Integration:

  Workday: The system integrates with Workday, allowing Lambda to gather employee information.
- Benefits:

  Access to Employee Data: Provides seamless access to employee information, supporting more comprehensive HR responses.

  Enhanced Response Generation: The integration allows for more detailed responses to HR-related queries.

Version 3 (Embedding with Dovetail):

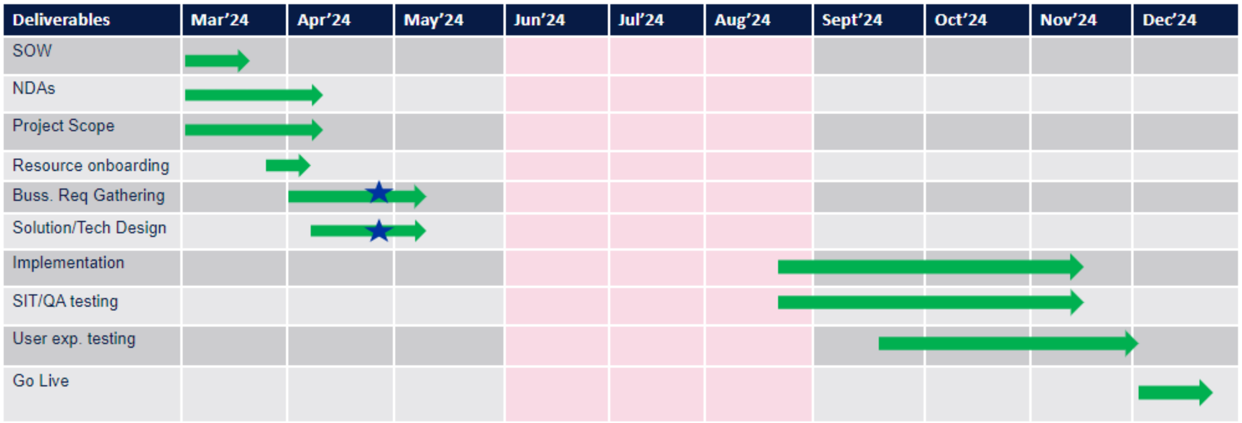
- Timeline: Developed after Version 2, focusing on deeper integration.
- Integration:

  Dovetail: Embeds the chatbot within Dovetail's existing HR ticketing system.
- Benefits:

Seamless Workflow: Allows employees to manage HR tickets directly through Dovetail, integrating responses from the chatbot.

Unified System: This integration provides a comprehensive solution, combining traditional ticketing with AI-powered support.

Here we are right now: the stars indicate our current position.



Deliverables	Mar'24	Apr'24	May'24	Jun'24	Jul'24	Aug'24	Sept'24	Oct'24	Nov'24	Dec'24
SOW	→									
NDAs	→	→								
Project Scope	→	→								
Resource onboarding	→	→								
Buss. Req Gathering		→ ★	→							
Solution/Tech Design		→ ★	→							
Implementation						→	→	→	→	
SIT/QA testing						→	→	→	→	
User exp. testing							→	→	→	
Go Live										→

Stage	Time Frame	Tasks
SOW (Statement of Work)	March 2024	- Draft and finalize the SOW, outlining the project's scope, objectives, and deliverables.
NDAs	March - April 2024	- Ensure non-disclosure agreements are signed by all team members for system access into Policy tech, AWS, etc.
Project Scope	March - April 2024	- Define the project's scope, including key milestones, deliverables, and deadlines.
Resource Onboarding	April 2024	- Recruit and onboard all resources, including technical, and managerial training sessions.

Business Requirements Gathering	April - May 2024	- Gather business requirements through stakeholder meetings, documentation, and analysis.
Solution/Tech Design	April - May 2024	- Design the technical solution, including system models, architecture, and DFDs.
Implementation	August - November 2024	- Infrastructure Setup: Set up AWS services (Lex, Lambda, Bedrock, S3, OpenSearch).
		- Lex Integration: Create a Lex bot, define intents, and integrate with Lambda.
		- Lambda Functions: Create functions for processing logic and managing communication with other services.
		- Bedrock Integration: Configure Lambda to interact with Bedrock's models (Claude V2, Llama 3-8B).
		- RAG Implementation: Retrieve data from OpenSearch and S3, supporting response generation.
		- Testing: Conduct System Integration Testing (SIT) and Quality Assurance (QA) testing.
User Experience Testing	November 2024	- Conduct user experience testing, gathering feedback from potential end-users.

Go Live	December 2024	- Launch the system, making it available to employees.

Breaking down the implementation into further elaborated steps:

Stage	**Time Frame**	**Tasks**
Infrastructure Setup	Aug 24 - Sept 5, 2024	- Set up the necessary AWS services: Amazon Lex, AWS Lambda, AWS Bedrock, Amazon S3, and OpenSearch.
		- Ensuring secure access between services, including encryption and data access controls.
Amazon Lex Integration	Sept 6 – Sept 21, 2024	- Create an Amazon Lex bot, configuring it to handle various HR-related queries.
		- Define intents to cover different HR topics, such as maternity leave, PTO policies, and tardiness.
		- Configure the bot to forward structured queries to AWS Lambda for further processing.
AWS Lambda Processing	Sept 22 - Oct 5, 2024	- Function Creation: Create Lambda functions to handle processing logic, determining which services to call based on query type.
		- Bedrock Integration: Lambda interacts with Bedrock's NLP models:

		- Claude V2: For general NLP tasks, including text generation and response processing.
		- LLaMA 3-8B: For instruction-following tasks, generating nuanced responses.
		- RAG Implementation: Lambda retrieves information from OpenSearch or S3 for response generation.
AWS Bedrock	Oct 6 - Oct 26, 2024	- Model Selection: Choose between Claude V2 or LLaMA 3-8B based on query context:
		- Claude V2: Offers comprehensive NLP capabilities, handling text generation and processing.
		- LLaMA 3-8B: Excels at instruction-following NLP tasks, generating structured responses.
		- Integration with Lambda: Models work seamlessly with Lambda, processing queries and returning responses.
Amazon S3 and OpenSearch	Oct 27 – Nov 13, 2024	- Data Retrieval: Lambda retrieves relevant information:
		- S3: Provides documents from the enterprise knowledge corpus or other stored data.

		- OpenSearch: Stores and retrieves embeddings or vector data for RAG processing.
		- Response Generation: Lambda combines information from Bedrock, S3, and OpenSearch, generating a comprehensive response.
Testing	Nov 14 - Nov 25, 2024	- SIT: Conduct System Integration Testing (SIT) to ensure seamless communication between services, identifying and resolving issues.
		- QA Testing: Perform Quality Assurance (QA) testing to ensure consistent performance and response quality.
Deployment and Launch	Nov 26 – Dec15, 2024	- Launch the system, make it available to employees, and continue monitoring performance.

Additional Considerations:

Documentation: Ensure comprehensive documentation is created at each stage, covering design, integration, and testing.

Support & Maintenance: Establish a plan for ongoing support and maintenance, ensuring consistent performance post-launch.

9. Lessons learned.

Feedback importance: We learned the value of ongoing feedback in shaping and refining system design. Regular input from users and stakeholders is important for aligning the final product with actual needs.

Integration challenges: Integrating new AI technologies like the Meta LLaMA 3-8B within AWS Bedrock revealed complexities as we researched using AWS documentation related to compatibility and ecosystems. Choosing the right components, like Lambda, Bedrock, etc., to fulfill each task will need careful evaluation. This will also emphasize the need for thorough testing and flexible planning to handle unexpected issues.

Resource management: Effective resource allocation helped us meet project timelines more efficiently. Predicting resource needs accurately is crucial for balancing quality and deadlines.

Setting Realistic Expectations: Understanding the limitations of new technologies helped manage stakeholder expectations, preventing any mismatch between promised and delivered capabilities.

Collaboration: Cross-functional collaboration was key in developing a system that considers all operational aspects, enhancing the project's overall success.

These insights will guide the next steps of the project and improve future technology implementations.