|  |  |  |
| --- | --- | --- |
| PROJECT NAME | **Fresenius Medical Care - Conversational AI** 501 | |
| Specify the official name of the project to be undertaken.  TEAM MEMBERS List the names and roles of all team members.   |  |  |  | | --- | --- | --- | | ID | NAME | ROLE | | 1 | Bharath Sreekumar | Software Developer | | 2 | Dhanushree Neelapu | System Engineer | | 3 | Palak Tandel | IT Project Manager and Data Architect | | 4 | Ritesh Pachgade | Business Systems Analyst and Data Analyst | | 5 | Shaunak Dhande | Software Architect | | 6 | Vineet Chheda | Software Quality Assurance Analyst and Software Developer | | | |
| BACKGROUND | | Why is this project important? What led to the initiation of this project? |
| Provide a brief overview of the project context and rationale. | | The project is important because it addresses a significant challenge at **Fresenius Medical Care**—the inefficiency of manually handling HR inquiries. This manual process is not only time-consuming but also detracts from other vital activities, such as patient care, which is a core focus of the organization. By implementing a **Conversational AI system**, the project aims to streamline and automate routine HR tasks, leading to several key benefits:   * **Increased efficiency**: Automating routine HR inquiries reduces the time and effort needed to manage these processes manually. * **Enhanced employee experience**: With routine inquiries being handled more efficiently, employees can get faster and more accurate responses to their HR-related questions. * **Focus on critical activities**: Automating HR inquiries allows the team to focus on more critical, patient-focused activities, improving overall care quality.   This project was initiated to improve the operational workflow and enhance the overall employee experience at Fresenius Medical Care. |

|  |  |
| --- | --- |
| MISSION | What is the primary goal the team aims to achieve? |
| State the fundamental purpose of the team. | The team's main goal is to deploy an MVP (Minimum Viable Product) Conversational AI system within the given time frame, specifically designed to automate and streamline repetitive HR tasks. The MVP will focus on addressing the most common and time-consuming HR inquiries, providing **instant** and **accurate** responses to enhance **employee engagement**, **satisfaction**, and **operational** **efficiency**. |

|  |  |
| --- | --- |
| OBJECTIVES | What are the key deliverables? What are the success criteria? |
| List specific, measurable outcomes the team should achieve. | Key Deliverables (MVP):  1. **Conversational AI Chatbot**: MVP chatbot focused on handling **three major HR inquiries** (e.g., general HR questions, payroll, and benefits). 2. **Basic HR Knowledge Base**: Integrated knowledge base to provide accurate answers to the selected inquiries. 3. **User Interface**: Simple, intuitive interface for employees to interact with the chatbot. 4. **Testing & Validation**: Comprehensive testing and user acceptance testing (UAT) for the three key questions. 5. **Documentation**: Complete documentation of MVP architecture and processes.  Success Criteria (MVP):  1. **Chatbot Functionality**: Successfully handles the **three main HR inquiries** without human intervention. 2. **Employee Feedback**: Positive feedback from users on the ease of use and accuracy of responses. 3. **Efficiency Gains**: Reduced time HR staff spend on the three inquiries, improving response times. 4. **Accuracy**: Achieve a **90%+ accuracy rate** in answering the key questions. 5. **Scalability Ready**: The MVP is designed to allow future integration with Workday and PolicyTech. 6. **Uptime**: Minimum **90% operational uptime** for employee access.  Post-MVP Expansion:  * **Workday Integration**: Expansion to integrate with Workday for real-time employee data handling. * **PolicyTech Embedding**: Embedding with PolicyTech to answer policy-related inquiries accurately. |

# Delivery Schedule

List the feature deliveries planned.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Sprint | Dates:  Start/Finish | Deliverables/Features/Tasks |
| 1 | Sprint 09.1 | Sept 4th / Sept 17th | Infrastructure setup – Getting Access to all services in AWS. |
| 2 | Sprint 09.2 | Sept 18th / Oct 1st | Implementing Lex and creating a knowledge base in AWS Bedrock after clarifying the scope. |
| 3 | Sprint 10.1 | Oct 2nd / Oct 15th | Integrating Lambda |
| 4 | Sprint 10.2 | Oct 16th / Oct 29th | Troubleshooting and Bug fixing |
| 5 | Sprint 10.3 | Oct 30th / Nov 12th | End-to-end Testing |
| 6 | Sprint 11.1 | Nov 13th / Nov 26th | Deployment and launch |
| 7 | Sprint 11.2 | Nov 27th / Dec 11th | Post MVP expansion |

|  |  |
| --- | --- |
| COMMUNICATION PLAN | What are the preferred communication channels and frequency? Who are the key stakeholders to be informed? |
| Outline how the team will communicate internally and with stakeholders. | **Preferred Communication channels:**   1. Microsoft Teams (FMC)  * Have meetings with the mentor and discuss the updates, **weekly**.  1. Microsoft Teams (UMD- 501 Fresenius Conversational AI Team)  * Discuss the doubts and immediate updates, **regularly**.  1. Microsoft Outlook (FMC)  * Communication with Margo and Zuwen through email, **twice a month**.  1. Offline (Classroom)  * Meet in the class, **twice a week**, and discuss the work.   **Key Stakeholders:**   * Kanti Singh, Dir Software Development (Direct Supervisor) * Margo Weeks, Dir Data Analytics * Zuwen Kuang, SVP Data & Analytics |

|  |  |
| --- | --- |
| RISK MANAGEMENT | What potential obstacles could arise? How will these be managed or mitigated? |
| Identify potential risks and mitigation strategies. | |  |  | | --- | --- | | **Obstacle** | **Mitigation** | | Resource Availability | - Resource Planning  - Contingency Planning | | Delayed Access to AWS Services | - Daily Follow-up  - Weekly Stand-up calls | | Delays in Sprint Completion | - Buffer Time  - Task Prioritization | | Team Skill Gaps | - Training Sessions  - Mentorship | | Scope Creep | - Strict Change Management  - Clear Documentation | | Data Privacy and Compliance Issues | - Compliance Checks  - Legal Consultation | | Technical Integration Challenges | - Pilot Testing  - Expert Consultation | |

Appendix: Deliverables (Backlog Chart)

|  |  |  |  |
| --- | --- | --- | --- |
| **Release** | **Sprint** | **Features** | **Product Backlog** |
| 1 | Sprint 09.1 | 1. Set up AWS infrastructure (Lex, Bedrock, Lambda, S3). | 1. Refine initial access setup for multiple AWS environments (dev, prod). |
|  |  | 2. Finalize the scope for MVP (questions, documents, and success criteria). | 2. Address any permissions issues with IAM policies. |
|  |  | 3. Test access to all AWS services (Lex, Bedrock, Lambda, and S3). | 3. Ensure cross-service communication is functional between AWS services (e.g., Lex and Lambda). |
|  |  | 4. Configure IAM roles for secure access. |  |
|  | Sprint 09.2 | 1. Implement Lex for conversational AI. | 1. Refine intents and slot filling after initial testing. |
|  |  | 2. Create a knowledge base in AWS Bedrock. | 2. Enhance integration between Lex and the knowledge base. |
|  |  | 3. Define and configure intents and slot filling in Lex. | 3. Add support for more complex queries and edge cases in Lex. |
|  |  | 4. Set up an S3 bucket for storing and linking data to the Bedrock knowledge base. |  |
|  |  | 5. Test the knowledge base and output from Lex interactions. |  |
| 2 | Sprint 10.1 | 1. Set up boto3 Lambda layer for easy interaction with AWS services. | 1. Add error handling to Lambda functions for various scenarios. |
|  |  | 2. Identify Lambda functions required for backend operations. | 2. Refine Lambda configuration to improve response times. |
|  |  | 3. Test the Lambda functions to ensure successful execution. | 3. Expand Lambda functions to handle more complex interactions with Lex and Bedrock. |
|  | Sprint 10.2 | 1. Identify and fix any integration issues between Lex, Bedrock, Lambda, and S3. | 1. Optimize performance across services, particularly with Lambda execution time. |
|  |  | 2. Troubleshoot access, performance, and permission issues in AWS services. | 2. Address scalability issues to handle larger datasets or more user interactions. |
|  |  | 3. Fix bugs identified during testing and setup. | 3. Fine-tune permissions and security settings for all AWS services. |
| 3 | Sprint 10.3 | 1. Conduct comprehensive end-to-end testing across all AWS services (Lex, Lambda, Bedrock, S3). | 1. Add additional test cases for less common scenarios. |
|  |  | 2. Set up test environments and develop test cases for each component. | 2. Perform stress testing to ensure the system can handle high loads. |
|  |  | 3. Document the system architecture and testing process for future reference. | 3. Improve test automation to speed up future tests and deployments. |
|  |  | 4. Perform user acceptance testing (UAT) to verify system functionality. |  |
|  | Sprint 11.1 | 1. Finalize and deploy MVP Version 1. | 1. Set up continuous monitoring and logging to track any post-deployment issues. |
|  |  | 2. Deploy the system into production. | 2. Address any deployment bugs or errors found during production rollout. |
|  |  | 3. Ensure all integrations (Lex, Bedrock, Lambda, S3) are functioning smoothly in the production environment. | 3. Prepare rollback strategies in case of critical issues. |
|  |  | 4. Monitor system performance post-launch. |  |
|  | Sprint 11.2 | 1. Knowledge transfer and documentation for the next phase. | 1. Refine knowledge base updates for more advanced queries. |
|  |  | 2. Update intents, slots, and the knowledge base based on feedback from MVP. | 2. Add new features and expand capabilities for Version 2. |
|  |  | 3. Implement and test MVP Version 2. | 3. Enhance monitoring and logging for long-term system maintenance. |