# High Level Design Document

|  |  |
| --- | --- |
| **Project Name** | Pro witty |
| **Author** | Jayanthi B |
| **Date** | 11/04/2024 |
| **Version** | 1.0 |

Version History:

|  |  |  |
| --- | --- | --- |
| **Version** | **Author** | **Updated Date** |
| V0 | Jayanthi B | 11/04/2024 |
| V0.2 | Gayathiri R | 12/04/2024 |
| V0.3 | Jayanthi B | 12/04/2024 |
| V0.4 | Jayanthi B | 14/04/2024 |
| V0.5 | Jayanth B | 18/04/2024 |

Review History:

|  |  |  |
| --- | --- | --- |
| **Version** | **Author** | **Updated Date** |
| V0 | Gayathiri R | 12/04/2024 |
| V0.3 | Gayathiri R | 14/04/2024 |
| V0.4 | Vishal | 17/08/2024 |

Approval History:

|  |  |  |
| --- | --- | --- |
| **Version** | **Author** | **Updated Date** |
|  |  |  |

|  |
| --- |
| **Table of Contents** |

1. Introduction

2. Abstract

3. System Architecture

3.1 Architecture Overview

3.2 System Architecture

4. Process

4.1 Components and Services Used

4.3 Development Methods

5. Design Considerations

5.1. Assumptions

5.2. Dependencies

6. Appendix

**1. Introduction**

The purpose of this design document is to present a detailed plan for the creation of an intelligent project assistant. This assistant aims to function as a knowledgeable companion for users engaged in project management, offering streamlined information retrieval and enhancing overall efficiency within project workflows. By providing specific details, historical records, and documentation related to projects, this assistant seeks to optimize the management process and facilitate smoother project execution.

**2. Abstract**

The primary objective of the project is to develop an intelligent project assistant capable of understanding user queries in natural language and retrieving relevant information from diverse data sources within the project ecosystem. By harnessing the power of artificial intelligence (AI) and cognitive computing, this aims to streamline the information retrieval process, empower users with timely and accurate insights, and ultimately enhance project management efficiency.

**3. System Architecture**

3.1 Architecture Overview

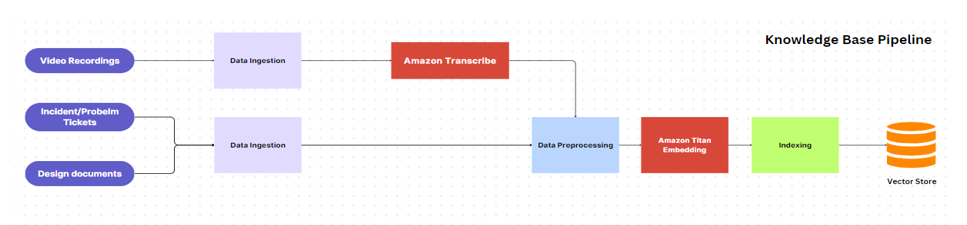
The project assistant (Pro Witty) is a comprehensive software system designed to streamline information retrieval processes and enhance project management efficiency. This model leverages advanced technologies such as Large Language Model which is a type of model Natural Language Processing (NLP), and knowledge representation to understand user queries and retrieve relevant information from diverse project data sources. The system architecture is composed of several key components working together seamlessly to deliver an intuitive and efficient user experience.

A diagram of a diagram

Description automatically generated with medium confidence

**3. System Architecture**

**Knowledge Base Pipeline**



The Knowledge Base Pipeline architecture is a systematic approach for processing and analyzing various types of input data, such as video recordings, incident problem tickets, and design documents. The pipeline begins with data ingestion, where the input data is provided to Amazon Transcribe for speech-to-text transcription. Following transcription, data preprocessing steps are applied to ensure data integrity and relevance. The preprocessed data then undergoes embedding generation using Amazon Titan Embedding, which generates vector embeddings representing the semantic meaning of the data. These embeddings are then indexed and stored in Vector DB for efficient storage and retrieval. This architecture enables the creation of a comprehensive knowledge base by converting diverse input data into structured and searchable representations, facilitating semantic understanding and information retrieval for various applications and use cases.

**Chat Pipeline**

A screenshot of a computer

Description automatically generated

The Chat-Based Pipeline architecture facilitates user interaction with the knowledge base by enabling queries and responses in a conversational format. When a user submits a query, it triggers a similarity search within the Vector Store to retrieve relevant knowledge chunks based on the semantic similarity of the query. The retrieved knowledge chunks are then processed using Amazon Titan Express to identify and extract relevant information. The extracted information is used to generate responses using Language Model (LM) techniques, such as LLM (Large Language Model). These responses are tailored to the user query and aim to provide accurate and contextual information in a conversational manner. This architecture empowers users to interact with the knowledge base dynamically, leveraging natural language understanding and response generation capabilities to facilitate effective communication and knowledge retrieval.

4.Process

The project process follows a systematic approach, starting with requirements gathering to understand end- user needs and objectives. Advanced technologies like Amazon Transcribe for speech-to-text transcription, Amazon Titan Embedding for generating vector embeddings, and Vector DB for efficient storage and indexing are utilized for semantic understanding and information retrieval. Data ingestion and preprocessing ensure data integrity and relevance, while similarity search indexing optimizes search efficiency. Amazon Titan Express, a scalable graph database, supports complex relationship queries, enhancing project management efficiency. Analysis of advanced technologies like Large Language Models (LLMs) for NLP and knowledge representation is conducted, with a focus on identifying key components and designing seamless interactions. Integration strategies for diverse data sources and mechanisms for user query understanding using NLP techniques are developed. User experience design ensures an intuitive interface, while scalability and performance optimization address system efficiency.

4.1 Components and Services Used

**1.User Interface (UI):** The UI interface component serves as the gateway for users to interact with the system, accommodating both text-based inputs such as incident/problem tickets in various formats like Word documents and Excel sheets, as well as voice-based inputs like MP3 recordings, enabling users to communicate naturally. Its primary function involves presenting information in a user-friendly format for easy comprehension and navigation. Users engage with the system by querying desired information, which prompts the component to retrieve relevant data from the vector database and provide the most suitable response, ensuring efficient communication and seamless interaction between users and the system.

**2.Data Ingestion:** Data ingestion refers to the process of importing and loading data into the system from various sources. This includes extracting data from databases, files, APIs, and streaming sources, transforming it into a usable format, and loading it into the target storage or processing system. Data ingestion pipelines are typically designed to handle large volumes of data efficiently, ensuring data integrity, consistency, and reliability. Common techniques used in data ingestion include batch processing, real-time streaming, and change data capture (CDC), depending on the nature and velocity of the data sources.

**3.Amazon Transcribe**: Amazon Transcribe is a service provided by Amazon Web Services (AWS) that converts speech into accurate text. It utilizes advanced machine learning algorithms to transcribe audio files, supporting various audio formats and languages. The service is scalable, reliable, and offers high accuracy, making it suitable for tasks such as transcribing meetings, lectures, interviews, and customer service calls. Amazon Transcribe simplifies the process of converting spoken language into written text, providing a valuable resource for applications requiring speech-to-text functionality.

**4. Data Preprocessing:** Data preprocessing involves a series of techniques for cleaning, transforming, and preparing data for analysis. This includes tasks such as removing duplicates, handling missing values, normalizing data, and feature engineering. Data preprocessing is essential for improving the quality, consistency, and relevance of data, ensuring that it is suitable for analysis and modeling tasks. Preprocessing techniques may vary depending on the specific requirements and characteristics of the data, but the goal is to prepare the data in a format that is conducive to extracting meaningful insights and patterns.

**5. Knowledge Base (KB):** The Knowledge Base serves as the repository of project-related information and documentation. The KB is continuously updated and enriched with new information to ensure its relevance and accuracy.

**6.Amazon Titan Embedding:** Amazon Titan Embedding is a service that leverages machine learning techniques to generate vector embeddings of text data. These embeddings capture the semantic meaning of words and phrases, enabling applications to understand the context and relationships between different pieces of text. By representing text data as vectors in a high-dimensional space, Amazon Titan Embedding facilitates tasks such as semantic search, recommendation systems, and natural language understanding. The service provides pre-trained models for generating embeddings from text data, as well as tools for customizing and fine-tuning models to suit specific use cases.

**7. Vector Database (Vector DB):** Vector Database (Vector DB) is a specialized database system designed for storing and indexing vector embeddings efficiently. It aggregates data from various sources such as project databases, repositories, and historical records. Unlike traditional relational databases, Vector DB is optimized for similarity search operations, enabling fast retrieval of similar vectors based on distance metrics such as cosine similarity or Euclidean distance. The database provides features for managing large-scale collections of vector data, including indexing, querying, and updating vectors in real-time. Vector DB is well-suited for applications requiring similarity search functionality, such as content recommendation systems, image search engines, and anomaly detection.

**8. Information Retrieval Engine (IRE):** The Information Retrieval Engine is tasked with retrieving relevant information from the Knowledge Base based on user queries. It employs advanced search algorithms, including keyword matching, semantic similarity, and relevance ranking, to efficiently locate and retrieve the most pertinent documents, records, or data points.

**9. Integration Layer:** The Integration Layer facilitates seamless integration with external data sources and systems, including project management tools, document repositories, and third-party APIs. It provides standardized interfaces and protocols for data exchange, ensuring interoperability and data consistency across the ecosystem.

**10. Similarity Search Indexing:** Similarity search indexing refers to methods for creating indexes and data structures optimized for similarity search operations. This includes techniques such as locality-sensitive hashing (LSH), tree-based indexing structures (e.g., kd-trees, ball trees), and approximate nearest neighbor algorithms. These methods enable efficient retrieval of similar items from large collections of data, based on similarity metrics such as distance or similarity scores. Similarity search indexing is commonly used in applications such as recommendation systems, content-based search engines, and pattern recognition tasks, where finding similar items or patterns is a key requirement.

**11. Amazon Titan Express:** Amazon Titan Express is a managed graph database service offered by AWS, based on the open-source Titan graph database. Graph databases are specialized for storing and querying relationships between data entities, making them ideal for applications with complex interconnections and dependencies. Amazon Titan Express provides a scalable and fully managed solution for storing and querying graph data, supporting features such as ACID transactions, distributed processing, and real-time analytics. The service simplifies the deployment and management of graph databases, allowing developers to focus on building applications that leverage rich, interconnected data.

**12. Natural Language Understanding (NLU):** The NLU component is responsible for parsing and understanding user queries expressed in natural language. It utilizes state-of-the-art NLP models and techniques to extract the intent and entities from user inputs, enabling model to comprehend the user's information needs accurately.

13. Machine Learning Models (ML Models): This model utilizes machine learning models to enhance its capabilities in understanding user queries, predicting user preferences, and improving search relevance. These models are trained on historical user interactions and feedback data to continuously refine and optimize Techton's performance.

**4.2 Development Methods**

**1. Setting up GitHub Repository:** Create a GitHub repository (version 3.2) to host the project code. This will serve as a central location for collaboration, version control, and tracking changes throughout the development process.

**2. Python Programming:** Utilize Python as the primary programming language for implementing the project. Python ( version 3.11) offers a wide range of libraries and frameworks suitable for tasks such as data processing, natural language processing, and web development.

**3.Integration of Advanced Technologies:** Integrate advanced technologies such as Amazon Transcribe for speech-to-text transcription, Amazon Titan Embedding for generating vector embeddings, and Vector DB for efficient storage and indexing and Utilization of Amazon Titan Express. These technologies will provide the foundation for semantic understanding and information retrieval within the project assistant software.

**4.Data Ingestion and Preprocessing:** Implement data ingestion and preprocessing pipelines to ensure the integrity and relevance of project data. This may involve cleaning and formatting raw data, extracting relevant features, and organizing data for efficient storage and retrieval. (DB information - Chroma DB )

**5. Design Considerations**

**5.1. Assumptions**

**1.Technology Stability:** The model assumes that the selected technologies such as Amazon Transcribe, Titan Embedding, Vector DB, and Amazon Titan Express are stable and reliable for the duration of the project. Any unexpected changes or disruptions to these technologies could impact the project's timeline and deliverables.

**2.Data Availability:** It assumes that project data required for development and testing purposes will be readily available and accessible. Delays or limitations in data availability could affect the project's progress and outcomes.

    **3. User Adoption:** The model assumes that end-users will adopt and utilize the project assistant software as intended. Factors such as user preferences, training needs, and organizational culture may influence user adoption and acceptance.

**4**.**Security Compliance:** It assumes that the implemented security measures are sufficient to protect project data and ensure compliance with relevant regulations and standards.

**5.2. Dependencies**

**1. Technological Dependencies:** The model is dependent on specific technologies and platforms such as Amazon Web Services (AWS) for hosting and cloud services, Python programming language for development, and GitHub for version control. Any changes, limitations, or disruptions in these dependencies could impact the project's implementation and functionality.

**2. Data Dependencies:** It relies on the availability and quality of project data for tasks such as training machine learning models, testing algorithms, and validating system performance. Dependencies on third-party data sources or data providers may introduce risks related to data availability, consistency, and reliability.

**3. Stakeholder Dependencies:** The model depends on active engagement and collaboration with end-users. Timely feedback, approval, and support from end-users are essential for driving project progress and resolving issues.

**4. Resource Dependencies:** It depends on the availability of skilled personnel, adequate funding, and necessary infrastructure to support project activities. There may be a chance that alternative resources supports project activities due to other interlocking projects.

**5.3. Risks**

|  |  |  |
| --- | --- | --- |
| **Risk** | **Overview** | **Mitigation plan** |
| Technology Dependency | The project relies heavily on specific technologies such as Amazon Transcribe, Titan Embedding, and Vector DB. Any changes or discontinuation of support for these technologies could disrupt the project's progress and functionality. | Regularly monitor the technology landscape for updates or changes related to Amazon Transcribe, Titan Embedding, and Vector DB. Maintain open communication channels with technology providers to stay informed about any potential changes or disruptions. Develop contingency plans to switch to alternative technologies if necessary. |
| Algorithm Performance | The effectiveness of the semantic understanding and information retrieval algorithms implemented using Titan Embedding and Vector DB may not meet expectations. Fine-tuning and optimization may be required to achieve desired performance levels. | Conduct thorough benchmarking and performance testing of the semantic understanding and information retrieval algorithms. Fine-tune algorithms based on performance metrics and user feedback. Consider implementing parallel processing or distributed computing solutions to improve performance scalability. |
| Data Availability | Availability of project data for ingestion and preprocessing may be limited or delayed, impacting the development timeline and testing phases. Incomplete or inconsistent data may also affect the accuracy and reliability of the project assistant software. | Establish clear communication channels with data providers to ensure timely availability of project data. Implement data quality checks and validation processes to identify and address inconsistencies or incompleteness in the data. Develop strategies for data augmentation or synthesis to supplement missing data if needed. |
| Complexity of Relationship Queries | Utilizing Amazon Titan Express for complex relationship queries introduces the risk of query optimization challenges. Ensuring efficient query execution and minimizing response times may require expertise in graph database management and optimization techniques. | Invest in training and development for team members responsible for managing and optimizing queries in Amazon Titan Express. Leverage best practices and optimization techniques for graph database management. Implement caching mechanisms or query optimization strategies to improve query performance and reduce response times. |
| Integration Issues | Integrating various components and technologies, such as speech-to-text transcription and similarity search indexing, may encounter compatibility issues or require custom integration solutions. Ensuring seamless interoperability between different modules is crucial for the project's success. | Conduct thorough integration testing to identify and resolve compatibility issues early in the development process. Establish clear communication channels between development teams working on different components to facilitate seamless integration. Consider leveraging integration frameworks or middleware solutions to streamline integration efforts. |
| User Adoption and Acceptance | The project assistant software's complexity or user interface design may not align with users' preferences or expectations. Poor user adoption and acceptance could hinder the software's effectiveness and utility within the organization | Involve end-users in the design and development process through user feedback sessions, usability testing, and prototype demonstrations. Incorporate user-centered design principles to ensure the software meets users' needs and expectations. Provide comprehensive training and support resources to facilitate user adoption and acceptance. |
| Performance Scalability | As the volume of project data grows over time, scalability issues may arise, impacting the software's performance and responsiveness. Implementing scalable architecture and optimizing resource utilization are essential to accommodate increasing data loads. | Design the software architecture with scalability in mind, leveraging cloud-based infrastructure and scalable database solutions. Implement horizontal scaling techniques such as load balancing and auto-scaling to handle increasing data loads. Monitor system performance and capacity regularly, scaling resources as needed to maintain optimal performance. |

**6. Appendix**

NA

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of Document\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*