Software Design Document for AI-powered Knowledge Organizer (AIKO)

Version 1.5

Prepared by:
Devansh Parapalli, Kaustubh Warade,
Aditya Deshmukh, and Yashasvi Thool

Government College of Engineering, Nagpur

August 16, 2024

Table of Contents

List of Figures

List of Tables

Revision History

Name	Date	Change Description	Version
Initial Version	August 04, 2024	Initial release of the Software Design Document	1.0
1.1	August 06, 2024	Added requirements matrix and appendices	1.1
1.2	August 08, 2024	Added Glossary as Appendix A	1.2
1.3	August 10, 2024	Updated System Architecture and User Interface Design	1.3
1.4	August 12, 2024	Added Colors, Design and Typography Information	1.4
1.5	August 16, 2024	Finalized the document for submission	1.5

1. Introduction

1.1. Purpose

The purpose of this Software Design Document (SDD) is to describe the architecture and system design decisions for AIKO (AI-Powered Knowledge Organizer). AIKO is designed to address the multifaceted challenges in knowledge management and retrieval precipitated by the exponential proliferation of digital information across heterogeneous platforms. By leveraging state-of-the-art artificial intelligence and advanced data storage techniques, AIKO aims to provide an innovative solution for efficient information organization and access, thereby mitigating the cognitive load associated with managing vast quantities of disparate data.

1.2. Scope

1.2.1. Functional Scope

AIKO will encompass the following core functionalities:

1. Integration of diverse information sources into a cohesive, centralized knowledge repository
2. Multi-modal information processing and analysis, including but not limited to textual, audio, visual, and structured/unstructured datasets
3. Implementation of advanced search and retrieval mechanisms leveraging natural language processing (NLP) and semantic understanding
4. Provision of a cross-platform, user-centric interface optimized for accessibility and intuitive interaction
5. Robust data security and user privacy protection through state-of-the-art encryption and access control mechanisms

1.2.2. Application and Benefits

AIKO is designed for deployment in various contexts, including:

- Personal knowledge management ecosystems
- Organizational knowledge bases and corporate intranets
- Academic and research environments requiring sophisticated information retrieval

1.2.2.1. Key benefits

1. Substantial enhancement in productivity through streamlined information retrieval processes
2. Facilitation of data-driven decision-making via improved access to relevant knowledge assets
3. Significant reduction in cognitive load associated with managing and navigating large volumes of information
4. Promotion of knowledge discovery and cross-pollination of ideas through advanced content linking and suggestion algorithms

1.2.2.2. Objectives:

1. Develop a unified data layer capable of ingesting and harmonizing heterogeneous data sources
2. Engineer a high-performance, scalable search engine incorporating advanced NLP and machine learning algorithms for semantic analysis and context-aware information retrieval
3. Implement a multi-layered security architecture ensuring data integrity, confidentiality, and compliance with relevant data protection regulations
4. Design and develop an intuitive, responsive user interface leveraging modern web technologies and adhering to established usability heuristics
5. Create a robust API ecosystem to facilitate seamless integration with existing productivity tools and third-party applications

1.2.3. Alignment with Higher-Level Specifications

AIKO's design and functionality are in consonance with contemporary knowledge management systems and AI-powered information retrieval tools. The system adheres to industry standards for data interoperability, security protocols, and user interface design principles.

1.3. Product Overview

1.3.1. Product Perspective

AIKO is conceptualized as a standalone, yet highly integrable product designed to interface seamlessly with existing information ecosystems and platforms. The system architecture comprises the following key components:

1.3.1.1. System Interfaces:

- RESTful APIs for data ingestion, retrieval, and system administration
- Webhook system for real-time event notifications and integration with external services

1.3.1.2. User Interfaces:

- Responsive web application built on modern frontend frameworks (e.g., Svelte, Vue.js)
- Application Programming Interface (API) for power users and system administrators

1.3.1.3. Hardware Interfaces:

- Standard compatibility with web browsers and mobile devices
- Optional support for specialized hardware such as e-ink devices or smart displays

1.3.1.4. Software Interfaces:

- Integration modules for popular productivity suites
- Connectors for custom formats and data sources through extensible plugins

1.3.1.5. Communications Interfaces:

- HTTPS for secure web traffic
- WebSocket protocol for real-time, bidirectional communication
- SSL/TLS encryption for all data in transit

1.3.1.6. Memory Constraints:

- Utilization of scalable, cloud-based storage solutions (e.g., Amazon S3, Google Cloud Storage)
- Implementation of efficient data compression and deduplication techniques

1.3.1.7. Operations:

- Automated monitoring and alerting systems
- Continuous integration and deployment (CI/CD) pipeline for seamless updates

1.3.1.8. Site Adaptation Requirements:

- Configurable self-hosted deployment option for organizations with specific data sovereignty requirements
- Customizable theming and branding capabilities
- Localization and internationalization support

1.3.2. Product Functions

AIKO's core functions encompass:

1.3.2.1. Intelligent Knowledge Integration:

- Automated ingestion and categorization of information from diverse sources
- Entity recognition and relationship mapping across different content types
- Continuous learning and knowledge base enrichment through user interactions

1.3.2.2. Multimodal Information Processing:

- Natural language processing for textual content analysis
- Computer vision algorithms for image and video content understanding
- Audio processing and speech-to-text conversion for audio content

1.3.2.3. Advanced Search and Retrieval:

- Semantic search capabilities with natural language query understanding
- Context-aware result ranking and personalized recommendations
- Faceted search and filtering options for precise information discovery

1.3.2.4. Cross-Platform Accessibility:

- Real-time synchronization of data across devices and platforms
- Progressive Web App (PWA) implementation for seamless mobile experience

1.3.2.5. Personalization and Adaptive Learning:

- User behavior analysis for tailored content suggestions
- Customizable dashboards and information feeds
- Collaborative filtering for knowledge sharing within organizations

1.3.3. User Characteristics

The intended user base for AIKO encompasses:

1.3.3.1. Knowledge Workers:

- Proficiency: Advanced digital literacy
- Usage Pattern: Heavy daily use for information management and retrieval
- Key Requirements: Efficiency, accuracy, and integration with existing workflows

1.3.3.2. Researchers and Academics:

- Proficiency: High technical competence in specific domains
- Usage Pattern: Intensive use for literature review and knowledge synthesis
- Key Requirements: Comprehensive search capabilities, citation management, and collaboration features

1.3.3.3. Corporate Professionals:

- Proficiency: Varying levels of technical expertise
- Usage Pattern: Regular use for decision support and information sharing
- Key Requirements: User-friendly interface, robust security, and integration with enterprise systems

1.3.3.4. Students:

- Proficiency: Basic to intermediate digital skills
- Usage Pattern: Periodic intensive use for study and project work
- Key Requirements: Intuitive interface, multi-format content support, and collaborative features

1.3.3.5. Data Scientists and Analysts:

- Proficiency: Advanced technical skills in data manipulation and analysis
- Usage Pattern: Regular use for data exploration and knowledge extraction
- Key Requirements: API access, support for large datasets, and integration with analysis tools

1.3.4. Limitations

1.3.4.1. Data Source Constraints:

- System effectiveness is contingent upon the quality and availability of source data
- Certain proprietary data formats may have limited support

1.3.4.2. AI Model Limitations:

- Processing capabilities are bounded by the current state of AI and ML technologies
- Potential for bias in AI-driven recommendations based on training data

1.3.4.3. Connectivity Requirements:

- Full functionality is dependent on internet connectivity, with limited offline capabilities

1.3.4.4. Regulatory Compliance:

- System deployment and data handling may be subject to region-specific regulatory requirements

2. System Overview

AIKO is an innovative solution designed to revolutionize information management across multiple platforms. It addresses the challenges of information overload and fragmented knowledge management by leveraging cutting-edge artificial intelligence and advanced data storage techniques.

The system integrates diverse information sources into a cohesive knowledge bank, processes multi-modal content (text, audio, video, structured/unstructured datasets), and provides advanced search and retrieval mechanisms. AIKO offers a cross-platform, user-centric interface optimized for accessibility and intuitive interaction.

2.1. Components and Subsystems

AIKO comprises the following key subsystems and components:

2.1.1. Data Ingestion and Integration

- Responsible for ingesting and harmonizing data from various sources into one unified format
- Utilizes data connectors, APIs, and web scraping techniques for information retrieval
- Implements data transformation and normalization processes for seamless integration

2.1.2. Information Processing Subsystems

- Responsible for processing and analyzing the ingested data across multiple modalities
- Utilizes NLP, Computer Vision (Face Recognition, Object Detection), and Audio Processing algorithms
- Generates metadata, tags, and annotations for enhanced search and retrieval

2.1.3. Knowledge Base and Indexing

- Responsible for storing and indexing the processed data for efficient retrieval
- Utilizes Vector Databases, Query Processing Engines, and Indexing Algorithms along with Robust, Secure and Scalable Storage Solutions
- Implements robust security and access control mechanisms for data protection

2.1.4. Search and Retrieval Engine

- Responsible for providing advanced search capabilities and personalized recommendations
- Utilizes Semantic Search, NLP Query Understanding, and Data Ranking Algorithms
- Implements faceted search, filtering options, and result presentation customization

2.1.5. User Management and Data Synchronization Subsystems

- Responsible for managing user profiles, access controls, and synchronization across devices
- Utilizes OAuth, social login mechanisms.
- Implements real-time synchronization and conflict resolution strategies
- Provides a notification system for user alerts and updates

2.1.6. User Interface and Experience Subsystems

- Responsible for providing an intuitive, responsive, and user-friendly interface
- Utilizes modern web technologies, responsive design principles, and accessibility standards
- Split into the following components:
  - Web Application (PWA-compatible)
  - Browser Extension
  - API (with documentation) for third-party integrations as well as power users

2.2. Interactions

The subsystems within AIKO interact with each other to provide a seamless user experience and efficient information management. The data ingestion and integration subsystem feeds data into the information processing subsystems, which generate metadata and annotations for storage in the knowledge base. The search and retrieval engine retrieves relevant information based on user queries and preferences, while the user management subsystem ensures secure access and synchronization across devices. The user interface subsystem provides an intuitive interface for users to interact with the system and access the information stored in AIKO.

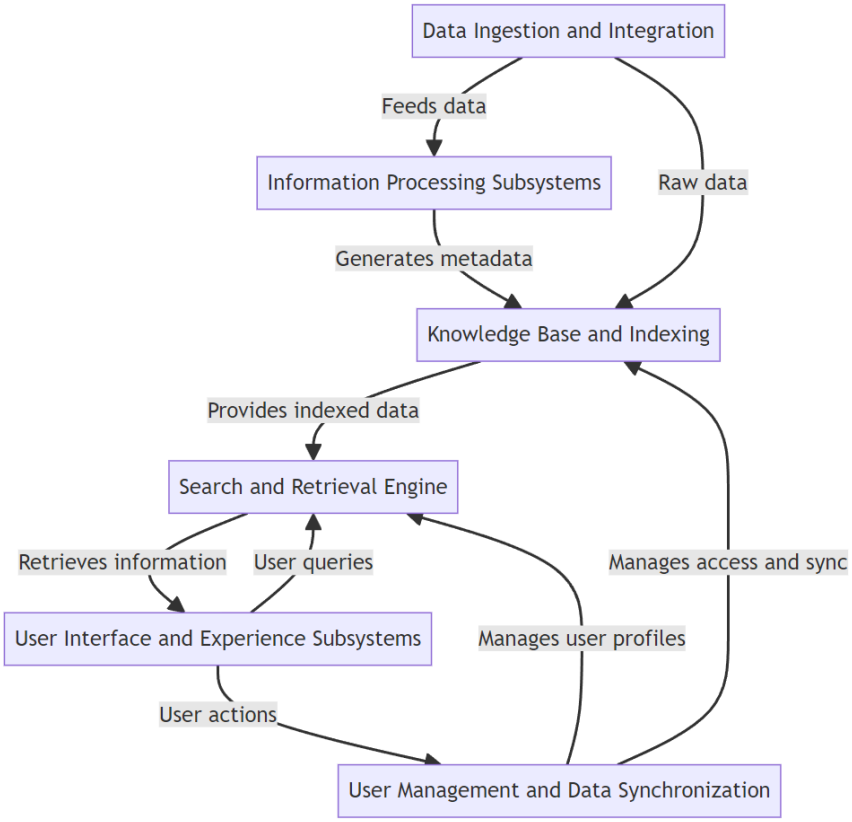

Figure 1: System Interactions

3. System Architecture

AIKO follows a modular, containerized architecture to ensure scalability, flexibility, and maintainability. The system architecture is designed to accommodate future enhancements, integrations, and customizations while maintaining high performance and reliability. The components mentioned in Section 2 are designed in a layered approach to ensure separation of concerns and ease of development and maintenance.

3.1. Architectural Design

3.1.1. Frontend Layer:

- Web Application: A responsive, user-friendly interface for accessing AIKO's features written with modern web technologies.
- Browser Extension: An optional extension for seamless integration with web browsers.

3.1.2. API Layer:

- Acts as a single entry point for all client requests and routes them to the appropriate services.
- RESTful APIs: Exposes endpoints for data ingestion, retrieval, user management, and system administration.

3.1.3. Services Layer

- Contains and coordinates the following services:
- Data Ingestion Service: Responsible for ingesting and transforming data from various sources. (Heavy use of LLMs, Transformers, and Data Wrangling techniques)
- Information Processing Service: Processes and analyzes data using NLP, Computer Vision, and Audio Processing algorithms. (Heavy utilization of Transformers, CNNs, RNNs, and Audio Processing Libraries is expected)
- Search and Retrieval Service: Provides advanced search capabilities and personalized recommendations. (Implemented using Vector Databases, Query Processing Engines, and Machine Learning Ranking Algorithms)
- User Management Service: Manages user profiles, access controls, and synchronization across devices.
- Notification Service: Sends real-time alerts and updates to users. (Implemented using WebSockets and Push Notification APIs)
- Synchronization Service: Ensures data consistency and synchronization across devices.

3.1.4. Data Layer

- Knowledge Base: Stores processed data, metadata, and annotations for efficient retrieval. (Implemented using R2, S3 or equivalent)
- Indexing and Retrieval Engine: Indexes, tags and embeds the data for fast search and retrieval.

3.1.5. Constraints and Assumptions

- The system will initially be deployed on cloud infrastructure for scalability. However, it will be designed to be deployable on-premises for organizations with specific data sovereignty requirements.
- All services and subsystems will communicate via RESTful APIs over HTTPS for security.
- The system will follow a stateless architecture to ensure scalability and fault tolerance.

3.2. Decomposition Description

AIKO is decomposed into the following components:

3.2.1. Frontend Layer

The frontend layer consists of the web application and browser extension components. The web application provides users with an intuitive interface to interact with AIKO's features, while the browser extension offers additional functionality for seamless integration with web browsers.

3.2.1.1. Web Application

- Built using modern web technologies (e.g., React, Angular, Vue.js)
- Responsive design for cross-platform compatibility
- Progressive Web App (PWA) implementation for offline access and mobile experience
- Customizable theming and branding options (saved along with user preferences)
- Integration with AIKO's API layer for data retrieval and management
- Support for multi-format content display (text, images, videos, audio) and search within content

3.2.1.2. Browser Extension

- Optional extension for popular web browsers (e.g., Chrome, Firefox)
- Provides quick access to AIKO's ingest features from the browser toolbar
- Integration with browser APIs for seamless data sharing and synchronization, particularly for web content such as articles, images, and videos
- Real-time notifications and alerts for new content suggestions and updates
- Customizable settings for user preferences and data sharing permissions

3.2.2. API Layer

The API layer serves as the primary interface for client-server communication, routing requests to the appropriate services within AIKO's architecture.

- Responsible for routing client requests to the corresponding services, that may or may not be contained within the same container/layer.
- Exposes RESTful APIs for data ingestion, retrieval, user management, and system administration.
- All requests are done using JSON payloads and follow a standardized API schema.
- Implements authentication and authorization mechanisms for secure access to the system.

3.2.3. Services Layer

The services layer is broken down into the following services:

3.2.3.1. Ingestion Service

- Responsible for ingesting and transforming data from various sources
- Utilizes transformers, data wrangling techniques, and machine learning models for data processing
- Supports batch and real-time data ingestion
- Provides data validation, normalization, and enrichment capabilities

3.2.3.2. Information Processing Service

- Processes and analyzes data using NLP, Computer Vision, and Audio Processing algorithms
- Generates metadata, tags, and annotations

3.2.3.3. Search and Retrieval Service

- Provides advanced search capabilities and personalized recommendations
- This component contains the core search engine, indexing algorithms, and ranking models as well as a API for querying the search engine

3.2.3.4. User Management Service

- Manages user profiles, access controls, and synchronization across devices, including OAuth, and social login mechanisms

3.2.3.5. Storage and Indexing Service

- Stores processed data, metadata, and annotations for efficient retrieval
- Implements robust security and access control mechanisms
- Indexes, tags, and embeds the data for fast search and retrieval

3.2.3.6. Notification Service

- Sends real-time alerts and updates to users
- Utilizes WebSockets and Push Notification APIs for real-time communication

3.2.4. Data Layer

The data layer consists of the following components:

3.2.4.1. Relational Database

The relational database is used to host nominal data such as user profiles, content metadata, system configurations, and other meta-metadata.

The database schema is provided in Appendix C.

3.2.4.2. Vector Database

The vector database is used to store embeddings and metadata for content items. This database is optimized for similarity search and retrieval of high-dimensional vectors.

The vector database schema is provided in Appendix D.

3.2.4.3. Object Storage

The object storage should be implemented keeping a fragmented storage system in mind, where a particular users data may not reside on the same server or service. Each provider or service will

be implemented based on a modular plugin-based architecture. A reference to the file present in the object storage will be stored in the relational database. For more info, refer Table 2 and Appendix C.

3.3. Design Rationale

The architectural design of AIKO is based on the following principles:

3.3.1.1. Modularity

The system is decomposed into independent services to facilitate scalability, maintainability, and extensibility. Each service will be responsible for a specific set of functionalities, allowing for easier development and deployment of new features.

3.3.1.2. Containerization

Each Service is containerized using Docker (or OCI-compatible interface) for portability and consistency across environments. Containerization also allows us to independently scale services based on demand.

3.3.1.3. RESTful APIs

All communication between services and clients is done via RESTful APIs to ensure interoperability and ease of integration. All API traffic will be routed through a load balancer for load distribution and fault tolerance.

3.3.1.4. Stateless Architecture

The system follows a stateless architecture to enable horizontal scaling and fault tolerance. Session Management is handled using JWT tokens and OAuth mechanisms.

3.3.1.5. Cloud-Native

AIKO is designed to be deployed on cloud infrastructure for scalability, reliability, and cost-effectiveness. The system will also be able to be deployed on-premises for organizations with specific data sovereignty requirements.

3.3.1.6. Security

Robust security mechanisms are implemented at each layer to protect user data and ensure compliance with data protection regulations.

3.3.1.7. User-Centric Design

The user interface and experience are designed to be intuitive, responsive, and accessible across devices. Customization options and personalization features are provided to enhance user engagement and increase quality of service.

4. Data Design

4.1. Data Description

AIKO's data model is designed to accommodate various types of information and metadata. The system uses a combination of structured and unstructured data storage to efficiently handle diverse data types while maintaining flexibility and scalability.

The following sections describe the data entities and attributes used within the system:

4.1.1. Data Entities

1. User: Represents system users with their profiles and preferences.
2. Entity: Represents the core information units stored in AIKO, including documents, images, audio files, and videos, contains the tagging and metadata information.
3. Messages: Represents the chat messages exchanged between users and the system for NLP tasks.

4.1.2. Data Storage

- Structured Data (such as User Profiles, Content Metadata) is stored in a relational database (PostgreSQL or equivalent) along with text-based search indexes
- Unstructured Data (such as Images, Audio, Video, Documents) is stored in a scalable object storage system (Amazon S3, Google Cloud Storage)
- Vector embeddings and metadata are stored in a high-performance, Vector DB

4.2. Data Dictionary

The data dictionary for AIKO includes the following key entities and attributes:

4.2.1. User

Attribute	Type	Description	Constraints
id	UUID	Unique identifier for the user	Primary Key, Foreign Key(`auth.users`)
name	TEXT	User's full name	
theme	TEXT	User's preferred theme	Default: `light`
other	JSONB	Custom user data that can be passed to the LLM when needed	
updated_at	TIMESTAMP WITH TIME ZONE	Last updated timestamp	Default: `now()`

Table 1: User Database Table Schema

4.2.2. Entity

Attribute	Type	Description	Constraints
id	UUID	Unique identifier for the document, used to link with a Vector Database	Primary Key, Default: `uuid_generate_v4()`
user	UUID	Reference to the user who owns this entity	Foreign Key(`auth.users`)
source	TEXT	Format: `source::identifier`, e.g., `gdrive::https://drive.google.com/file/d/...`	NOT NULL
type	TEXT	Document type: pdf, video, audio, text, webpage, etc.	NOT NULL
title	TEXT	Title of the entity	
tags	TEXT[]	User-defined tags for the document	Default: `{}`
processed	BOOLEAN	Indicates whether the document has been processed and added to a Vector Database	Default: FALSE
processed_at	TIMESTAMP WITH TIME ZONE	Timestamp when the document was processed and added to a Vector Database	
created_at	TIMESTAMP WITH TIME ZONE	Creation timestamp	Default: `now()`
updated_at	TIMESTAMP WITH TIME ZONE	Last updated timestamp	Default: `now()`
metadata	JSONB	Additional structured metadata specific to the content type	

Table 2: Entity Database Table Schema

4.2.3. Messages

Attribute	Type	Description	Constraints
id	UUID	Unique identifier for the message	Primary Key, Default: `uuid_generate_v4()`
user	UUID	Reference to the user who sent or received this message	Foreign Key(`auth.users`)
entity	UUID	Reference to the associated entity	Foreign Key(`public.entity`)
content	TEXT	Content of the message	
is_user_message	BOOLEAN	Indicates whether the message is from the user	Default: TRUE
created_at	TIMESTAMP WITH TIME ZONE	Creation timestamp	Default: `now()`
metadata	JSONB	Additional metadata for the message	

Table 3: Messages Database Table Schema

5. Human Interface Design

AIKO's user interface is designed to be intuitive, responsive and accessible across various devices. It follows standardized guidelines on UI Design and emphasizes simplicity and effectiveness.

5.1. User Interface & Screen Images

The user interface consists of the following distinct views:

5.1.1. Dashboard + Search View

Figure 2: Dashboard View (Desktop)

The dashboard is the main entry point for AIKO. It allows the user to search, view results and access documents. The search bar provides advanced search capabilities, and the results are displayed in a staggered grid view.

The left sidebar contains the previous searches and a set of navigaitonal links. The right sidebar contains the model settings and user metadata.

The search bar is present at the bottom of the screen. A button next to the search bar allows the user to add a new file.

The mobile view of the dashboard is optimized for smaller screens. The search bar is prominently displayed at the bottom of the screen, and the results are displayed in a single column layout. The sidebar is tucked away in a collapsible menu for better screen utilization.

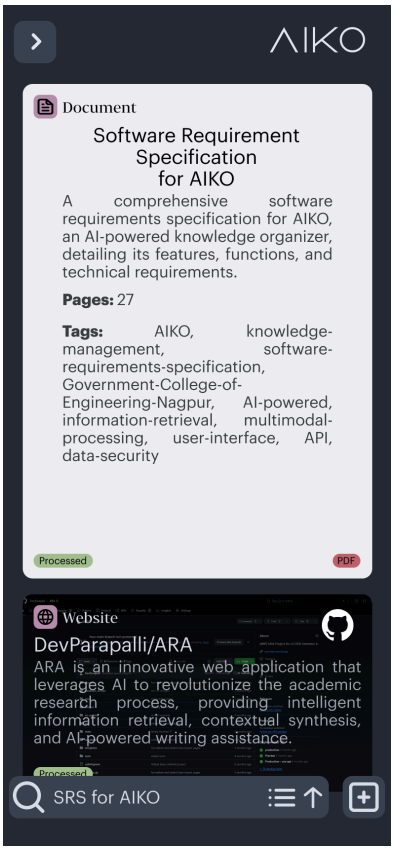

Figure 3: Dashboard View (Mobile)

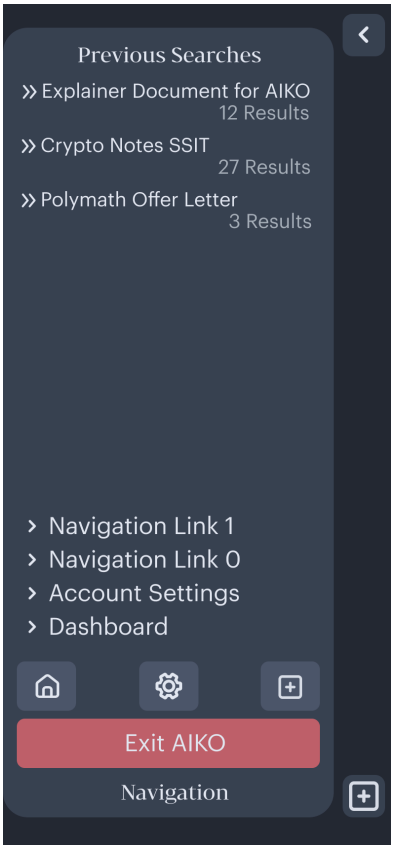

Figure 4: Sidebar (Mobile)

5.1.2. Document Chat View

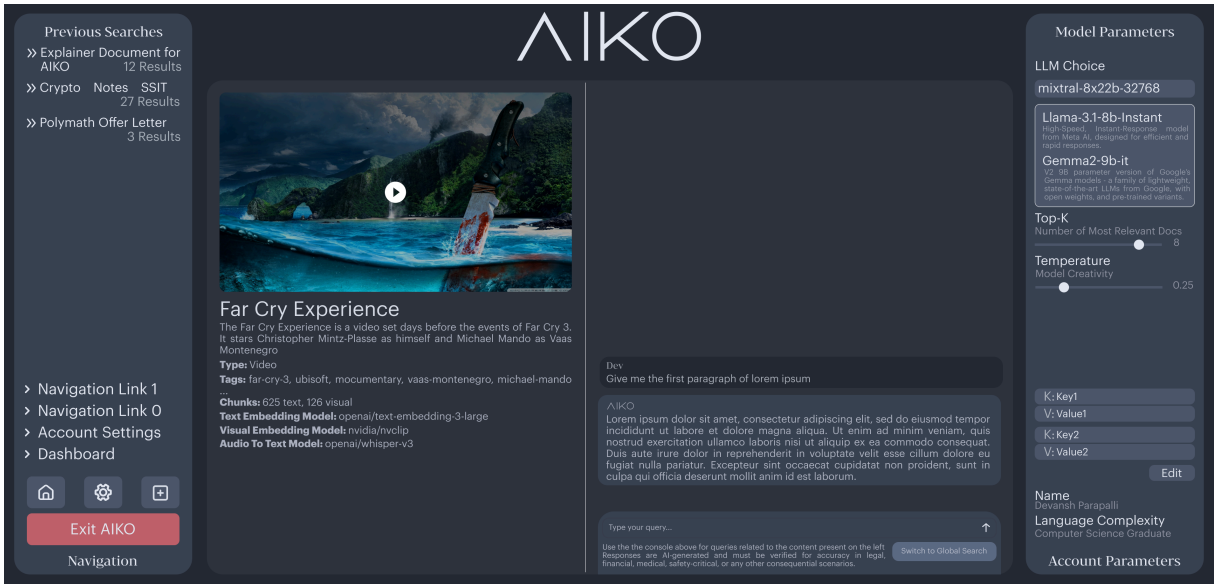

Figure 5: Document Chat View (Desktop)

The document chat view displays the content and metadata of a specific document. The right section contains a chatbot UI that allows the user to interact with the content for NLP tasks such as summarization, question-answering, etc.

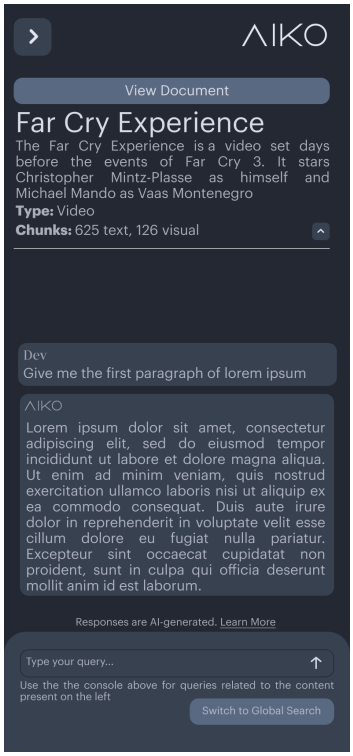

Figure 6: Document View (Mobile)

As before, the mobile view of the document chat view is optimized for smaller screens. The content is displayed in a single column layout, and the sidebar is accessible via a collapsible menu.

5.1.3. Display Cards

The following represent an example of the cards used to display search results and document previews.

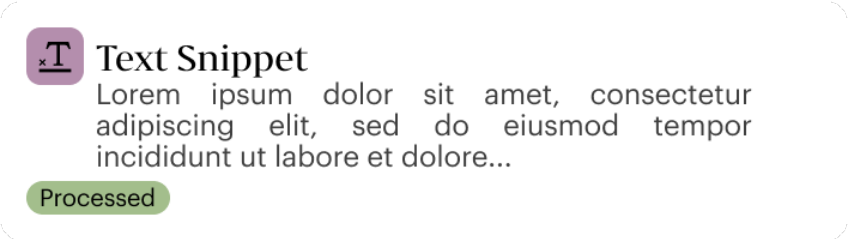

Figure 7: Text Snippet Card

The text snippet card displays a preview of the snippet and the status of the snippet's indexing.

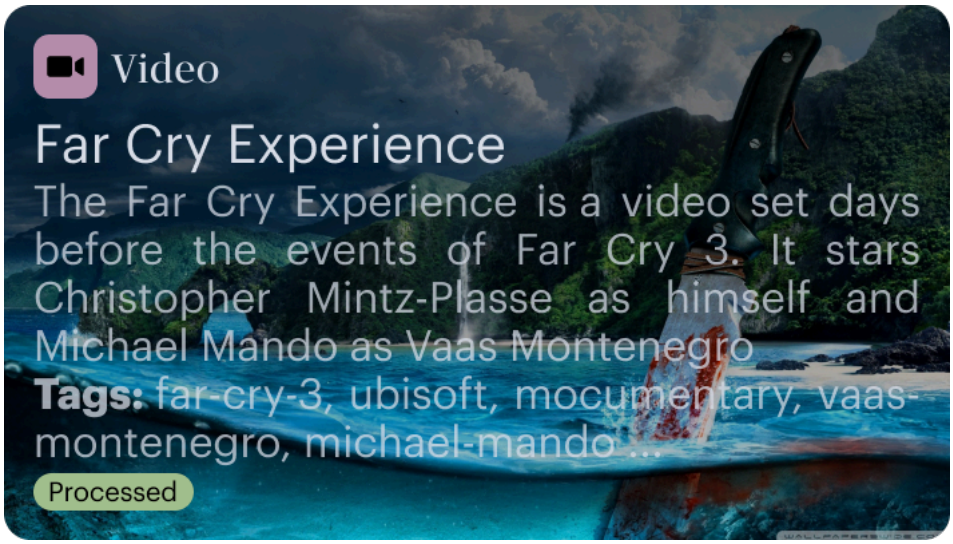

Figure 8: Media Card

The media card displays a preview of the media content and the status of the content's indexing along with a textual description and tags.

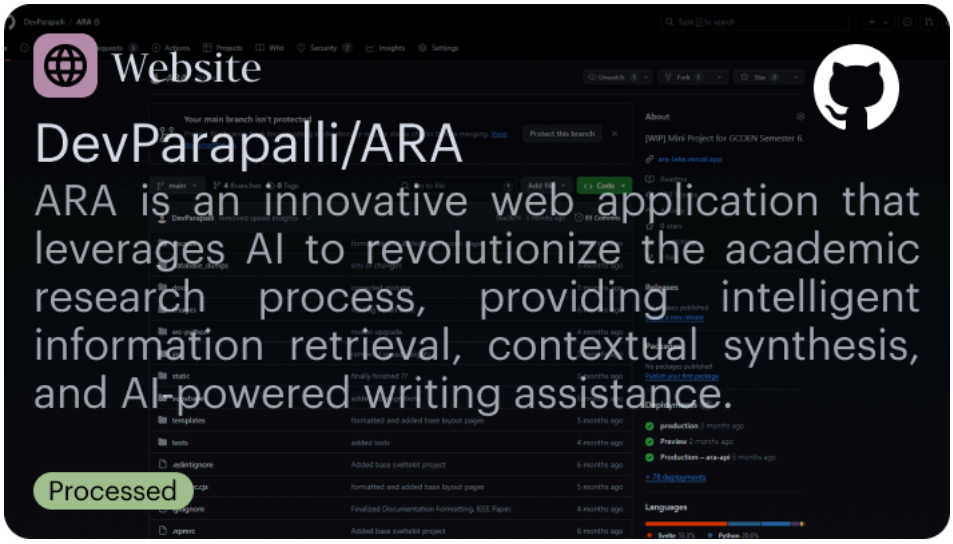

Figure 9: Website Card

The website card displays a preview of the website content and the status of the content's indexing along with a textual description and tags. The background is set to a screenshot of the website and its favicon is displayed in the top right corner.

5.2. Colors, Design and Typography

The default color palette of AIKO is based the `Nord` theme. `Nord` consists of four named color palettes providing different syntactic meanings and color effects for dark & bright ambiance designs.

All colors are numbered from `nord0` to `nord15` where each palette contains a different amount of colors. The naming convention preserves the compatibility for terminal color schemes and allows an uncomplicated use as base for such.

Figure 10: Color Palette

The first row of colors is called **Polar Night**. Polar Night is made up of four darker colors that are commonly used for base elements like backgrounds or text color in bright ambiance designs. AIKO adds another color to the palette to provide more flexibility in design. The new color is called `nord-1`.

The second row of colors is called **Snow Storm**. Snow Storm is made up of three bright colors that are commonly used for text colors or base UI elements in bright ambiance designs.

The third row of colors is called **Frost**. Frost can be described as the heart palette of Nord, a group of four bluish colors that are commonly used for primary UI component and text highlighting and essential code syntax elements. All colors of this sub-palette are used the same for both dark & bright ambiance designs.

The fourth row of colors is called **Aurora**. Aurora consists of five colorful components reminiscent of the "Aurora borealis", sometimes referred to as polar lights or northern lights.All colors of this sub-palette are used the same for both dark & bright ambiance designs.

The colors present in Figure 10 are provided as hex values in the table below:

#2e3440	#3b4252	#434c5e	#4c566a	#242933
#d8dee9	#e5e9f0	#eceff4		
#8fbcbb	#88c0d0	#81a1c1	#5e81ac	
#bf616a	#d08770	#ebcb8b	#a3be8c	#b48ead

Table 4: Hex Values of Extended Nord Palette

The proposed typography used in AIKO is based on the `BW Darius` and `Graphik` font families. The actual font used in the implementation may vary based on licensing, availability, readability, and design considerations.

Given below are the font descriptions provided by the respective foundries in the respective font families:

Designed by Alberto Romanos, **Bw Darius** is an elegant wedge serif typeface, halfway between the transitional and didone genres, with a sharper approach to terminals without falling on the stiffness of the didones. The wide skeleton, modern proportions and high contrast, all contribute to the opulent personality of this font.

Graphik was inspired by the appealing plainness seen in many of the less common 20th century European sans serifs and in the hand-lettering of classic Swiss Modern posters.First drawn as the house style for Schwartzco Inc., it was further developed for Condé Nast Portfolio and later for Wallpaper* and T, the New York Times Style Magazine. The low contrast and large x-height give the typeface great versatility.

6. Requirements Matrix

The Requirements Matrix maps the functional requirements of AIKO to the components and features that fulfill them. Each requirement contains the Description, Fulfilling Component as well as the User Interface Element that corresponds to it.

REQ-1: Multiple Input Sources and Formats

- Description: System shall support multiple input sources and formats
- Fulfilling Component: Data Ingestion Subsystem
- User Interface Element: Dashboard View, Browser Extension

REQ-2: Data Processing and Analysis

- Description: System shall process and analyze ingested data
- Fulfilling Component: Information Processing Subsystem
- User Interface Element: Background process, results visible in Dashboard View

REQ-3: Metadata Generation

- Description: System shall generate relevant metadata for all processed content
- Fulfilling Component: Information Processing Subsystem
- User Interface Element: Metadata section in Document UI

REQ-4: Advanced Faceted Search and Filtering

- Description: System will provide faceted search, filtering options and natural language queries
- Fulfilling Component: Search and Retrieval Engine
- User Interface Element: Dashboard View, Search Results

REQ-5: Multiple Authentication Methods

- Description: System shall support multiple authentication methods
- Fulfilling Component: User Management Subsystem
- User Interface Element: Login Page, User Settings

REQ-6: Real-Time Synchronization

- Description: System shall implement real-time synchronization across devices
- Fulfilling Component: Synchronization Subsystem
- User Interface Element: Background process, Dashboard UI and Document UI

REQ-7: Search Performance

- Description: System shall achieve 95th percentile of search queries returning in < 4 seconds
- Fulfilling Component: Search and Retrieval Engine
- User Interface Element: Search Results Page, Performance Metrics

REQ-8: Usability Score

- Description: System shall achieve a System Usability Scale (SUS) score of at least 80
- Fulfilling Component: User Interface Subsystem
- User Interface Element: Feedback Surveys, Usability Testing Results

Appendices

Appendix A: Glossary of Terms

- **AIKO (AI-Powered Knowledge Organizer)**: The project described in this document, an AI-powered knowledge management system designed to streamline information retrieval and organization.
- **API (Application Programming Interface)**: A set of definitions, protocols, and tools for building application software. It specifies how software components should interact, facilitating communication between different software systems.
- **CI/CD (Continuous Integration and Continuous Deployment)**: A method to frequently deliver apps to customers by introducing automation into the stages of app development. The main concepts attributed to CI/CD are continuous integration, continuous delivery, and continuous deployment.
- **CNN (Convolutional Neural Network)**: A class of deep neural networks, most commonly applied to analyzing visual imagery. They use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers.
- **Faiss (Facebook AI Similarity Search)**: A library developed by Facebook AI Research for efficient similarity search and clustering of dense vectors. It's particularly useful for tasks that require finding nearest neighbors in large datasets.
- **GDPR (General Data Protection Regulation)**: A regulation in EU law on data protection and privacy for all individual citizens of the European Union and the European Economic Area. It also addresses the transfer of personal data outside the EU and EEA areas.
- **HTTPS (Hypertext Transfer Protocol Secure)**: An extension of the HTTP protocol that is secured by SSL/TLS. It provides secure communication over a computer network, widely used on the internet for secure data transfer.
- **JSON (JavaScript Object Notation)**: A lightweight data-interchange format that is easy for humans to read and write and easy for machines to parse and generate. It's based on a subset of the JavaScript Programming Language and is commonly used for transmitting data in web applications.
- **JSONB (Binary JSON)**: A data type in some database systems that stores JSON data in a binary format. This format is typically more efficient for storage and processing compared to text-based JSON storage.
- **LLM (Large Language Model)**: A type of artificial intelligence model designed to understand, generate, and manipulate human language. These models are trained on vast amounts of text data and can perform a wide range of language-related tasks.

- **ML (Machine Learning)**: A subset of artificial intelligence that involves the development of algorithms and statistical models that enable computer systems to improve their performance on a specific task through experience, without being explicitly programmed.
- **NLP (Natural Language Processing)**: A branch of artificial intelligence that focuses on the interaction between computers and human language. NLP enables machines to understand, interpret, and generate human language, facilitating tasks such as translation, sentiment analysis, and speech recognition.
- **OAuth (Open Authorization)**: An open standard for access delegation, commonly used as a way for internet users to grant websites or applications access to their information on other websites but without giving them the passwords. It provides secure delegated access to server resources on behalf of a resource owner.
- **OCI (Open Container Initiative)**: An open governance structure for the express purpose of creating open industry standards around container formats and runtimes. It aims to establish common standards for software containers across different platforms and providers.
- **Pinecone**: A vector database designed for machine learning applications. It provides fast similarity search for high-dimensional vector data, making it useful for applications in natural language processing, computer vision, and recommendation systems.
- **PWA (Progressive Web Application)**: A type of application software delivered through the web, built using common web technologies including HTML, CSS, and JavaScript. It is intended to work on any platform that uses a standards-compliant browser, including both desktop and mobile devices.
- **R2 (Cloudflare R2 Storage)**: A rapid, reliable and distributed object storage service by Cloudflare, designed to be compatible with Amazon S3′s API. It offers benefits like reduced latency and elimination of egress fees, making it an attractive alternative for certain use cases.
- **RESTful (Representational State Transfer)**: An architectural style for designing networked applications. It relies on a stateless, client-server, cache-able communications protocol – typically HTTP. RESTful applications use HTTP requests to post, read, and delete data.
- **RNN (Recurrent Neural Network)**: A class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior, making them useful for tasks such as speech recognition and language translation.
- **S3 (Amazon Simple Storage Service)**: An object storage service offered by Amazon Web Services. It provides web services interfaces to store and retrieve any amount of data from anywhere on the web, designed for high durability, availability, and scalability.
- **SAML (Security Assertion Markup Language)**: An open standard for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider. It's commonly used for single sign-on (SSO) solutions.

- **SSL/TLS (Secure Sockets Layer/Transport Layer Security)**: Cryptographic protocols designed to provide communications security over a computer network. They are widely used for internet communications and online transactions, providing privacy, integrity, and authentication.
- **SUS (System Usability Scale)**: A simple, ten-item scale giving a global view of subjective assessments of usability. It allows you to evaluate a wide variety of products and services, including hardware, software, mobile devices, websites and applications.
- **UI (User Interface)**: The space where interactions between humans and machines occur. It's the visual part of a computer application or operating system through which a user interacts with a computer or software, including elements like buttons, menus, and icons.
- **UUID (Universally Unique Identifier)**: A 128-bit number used to identify information in computer systems. UUIDs are standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment (DCE).
- **UX (User Experience)**: The overall experience of a person using a product, especially in terms of how easy or pleasing it is to use. It encompasses all aspects of the end-user's interaction with the company, its services, and its products.

Appendix B: Links / References / Further Reading

- Nord Theme: https://www.nordtheme.com/
- BW Darius Font: https://brandingwithtype.com/typefaces/bw-darius
- Graphik Font: https://commercialtype.com/catalog/graphik
- Figma Design: https://l.parapalli.dev/aiko-figma

Appendix C: Database Schema

```sql
-- Users Table (extends auth.users)
CREATE TABLE public.users (
    id UUID PRIMARY KEY REFERENCES auth.users(id),
    name TEXT ,
    theme TEXT DEFAULT 'light',
    other JSONB,
    -- Additional fields can be added here as needed
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

COMMENT ON TABLE public.users IS 'Extends auth.users with additional user data and preferences';
COMMENT ON COLUMN public.users.other IS 'Custom user data that can be passed to the LLM when needed';

-- Entity Table
CREATE TABLE public.entity (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user UUID REFERENCES auth.users(id),
    source TEXT NOT NULL,
    type TEXT NOT NULL,
    title TEXT,
    tags TEXT[] DEFAULT '{}',
    processed BOOLEAN DEFAULT FALSE,
    processed_at TIMESTAMP WITH TIME ZONE,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    metadata JSONB
);

COMMENT ON TABLE public.entity IS 'Stores metadata about user content, with actual content stored in a Vector Database';
COMMENT ON COLUMN public.entity.id IS 'Unique identifier for the document, used to link with a Vector Database';
COMMENT ON COLUMN public.entity.source IS 'Format: source::identifier, e.g., gdrive::https://drive.google.com/file/d/...';
COMMENT ON COLUMN public.entity.type IS 'Document type: pdf, video, audio, text, webpage, etc.';
COMMENT ON COLUMN public.entity.tags IS 'User-defined tags for the document';
COMMENT ON COLUMN public.entity.processed IS 'Indicates whether the document has been processed and added to a Vector Database';
COMMENT ON COLUMN public.entity.processed_at IS 'Timestamp when the document was processed and added to a Vector Database';
```

```
COMMENT ON COLUMN public.entity.metadata IS 'Additional structured metadata specific to the content type';

-- Messages Table
CREATE TABLE public.messages (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user UUID REFERENCES auth.users(id),
    entity UUID REFERENCES public.entity(id),
    content TEXT,
    is_user_message BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP,
    metadata JSONB
)

COMMENT ON TABLE public.messages IS 'Stores LLM chat messages associated with users and entities';
COMMENT ON COLUMN public.messages.id IS 'Unique identifier for the message';
COMMENT ON COLUMN public.messages.user IS 'Reference to the user who sent or received this message';
COMMENT ON COLUMN public.messages.entity IS 'Reference to the associated entity';
COMMENT ON COLUMN public.messages.content IS 'Content of the message';
COMMENT ON COLUMN public.messages.is_user_message IS 'Indicates whether the message is from the user (true) or the system (false)';
COMMENT ON COLUMN public.messages.created_at IS 'Timestamp when the message was created';
COMMENT ON COLUMN public.messages.metadata IS 'Additional structured metadata specific to the message such as the model used or a link to previous version of the message';

-- Add any necessary indexes
CREATE INDEX idx_content_user_id ON public.entity(user_id);
CREATE INDEX idx_content_type ON public.entity(type);
CREATE INDEX idx_content_tags ON public.entity USING GIN(tags);

CREATE INDEX idx_messages_user_id ON public.messages(user_id);
CREATE INDEX idx_messages_entity_id ON public.messages(entity_id);

-- Add a trigger to update the 'updated_at' column
CREATE OR REPLACE FUNCTION update_modified_column()
RETURNS TRIGGER AS $$
BEGIN
    NEW.updated_at = NOW();
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER update_content_modtime
BEFORE UPDATE ON public.entity
FOR EACH ROW
EXECUTE FUNCTION update_modified_column();

CREATE TRIGGER update_users_modtime
BEFORE UPDATE ON public.users
FOR EACH ROW
EXECUTE FUNCTION update_modified_column();
```

Appendix D: Vector Database Schema

```
interface VectorDBMetadata {
    // Unique identifier for the document
    id: string;

    // User ID who owns this content
    user: string;

    // Title of the document
    title: string;

     // Source of the document (e.g., "gdrive::https://drive.google.com/file/d/...")
    source: string;

    // Type of content (e.g., "pdf", "video", "audio", "text", "webpage")
    type: string;

    // Set of User Defined and AIKO tags for filtering
    tags: string[];

     // Additional metadata that might be useful for specific content types, these fields are not guaranteed to be present
    metadata?: {
      [key: string]: any;
    };
  }
```

Approval:

Dr. D. J. Chaudhari
Project Guide
Assistant Professor, CSE Department
Sector-27, MIHAN Rehabilitation Colony
Khapri, Nagpur
441108
Date: August 16, 2024