BRD Analysis - 2025-08-12

Okay, here’s an analysis of the provided BRD content, focusing on identifying key requirements, potential issues, and suggesting improvements. **Overall Assessment:** The BRD is a good starting point but needs significant refinement to be truly actionable. It’s overly detailed in some areas and lacks crucial details in others. The structure is generally logical, but the flow of information could be improved. The reliance on placeholder values (e.g., "your_jwt_secret_key") is problematic. **Key Requirements (Extracted):** * **Authentication:** * Permanent JWT tokens (no expiration). * Bearer token authentication for all endpoints (except login). * Role-based access control based on username (admin vs. user). * Bcrypt password hashing with salt. * **Model Management:** * CRUD operations for AI models (Create, Read, Update, Delete – though the "Delete" functionality isn’t explicitly described). * Admin-only access to model creation and management. * Flexible model schemas (as noted in the GET ALL MODELS endpoint). * **Application Management:** * Registration of applications linked to AI models. * Auto-generation of unique app IDs. * Storage of app configuration and metadata. * **Invocation:** * Execution of registered applications. * Forwarding requests to external chat API services. * Multi-part form data handling (text, file uploads, etc.). * Storage of request/response data in MongoDB (grouped by user_id + app_id). * Audit logging of invocations. * **Guardrail Monitoring:** * Retrieval of flagged responses based on guardrail violations. * Filtering by user ID. * Extraction of content and guardrail violation information. * **Data Storage:** MongoDB for user management, chat responses, and model metadata. Grouped by user_id and app_id for chat responses. **Potential Issues & Concerns:** * **Lack of Detail in Operational Aspects:** The BRD lacks specifics on the data flow between the different APIs. For example, how does the “Invoke App” endpoint interact with the external chat API? What are the exact transformations of data? * **MongoDB Reliance:** The heavy reliance on MongoDB introduces potential consistency issues. The BRD doesn't outline the strategies for maintaining data consistency, especially given the flexible schema requirement. * **Admin-Only Access:** The BRD needs to clearly delineate the permissions and responsibilities of admin vs. developer roles. * **Guardrail Implementation:** The BRD describes the *output* of the guardrail monitoring but doesn’t explain *how* the guardrails are implemented. This is a critical omission. * **Error Handling:** The “ERROR FORMAT” section provides a generic template. This needs to be fleshed out with specific error codes, descriptions, and potential mitigation strategies. * **Performance Details:** The "PERFORMANCE" section is minimal. It needs to include specific performance targets (e.g., response times, throughput) and the strategies being used to achieve them. * **Missing Diagrams:** The BRD critically lacks diagrams. The outlined architecture would benefit immensely from a system diagram, API flow diagrams, and data flow diagrams. **Suggestions for Improvement:** 1. **Expand on Data Flow:** Add detailed diagrams illustrating the data flow between all the APIs. 2. **Define Guardrail Logic:** Clearly articulate the guardrail implementation—the rules, the thresholds, and the process for triggering violations. 3. **Document Data Consistency Strategies:** Detail how MongoDB’s flexible schema will be managed to avoid data inconsistencies. 4. **Specify Response Codes:** Expand the “ERROR FORMAT” section with a comprehensive list of error codes and their corresponding descriptions. 5. **Add Performance Metrics:** Include specific performance targets and the strategies being used to meet them. 6. **Define Roles & Permissions:** Create a detailed table outlining the permissions granted to each role (admin, developer, user). 7. **Clarify External API Integration:** Provide more details about the interaction between the internal APIs and the external chat API services. To help me refine my analysis further, can you tell me: * What is the primary business goal of this AI platform? * What type of AI models will be supported (e.g., LLMs, computer vision, etc.)? * Can you provide examples of the types of guardrail violations you anticipate?