Project BRD: Employee Data Management System

1. Introduction

This document outlines the Business Requirements Document (BRD) for an Employee Data Management System. This system will ingest, process, and store employee data from various sources on AWS S3 and Kafka. The processed data will be used to generate reports and identify potential leave abuse.

2. Business Needs

The system aims to:

- Manage employee data efficiently.
- Track employee leave history and quotas.
- Identify employees with potentially excessive leave requests.
- Monitor employee communication for potential misuse.

3. System Requirements

3.1 Data Sources

- Employee data (.csv): employee_data.csv (daily)
- Employee timeframe data (.csv): employee_timeframe_data.csv (daily with incremental updates) and employee_timeframe_data_*.csv (daily incremental files)
- Employee leave quota data (.csv): employee_leave_quota_data.csv (yearly)
- Employee leave calendar data (.csv): employee_leave_calendar_data.csv (yearly on Jan 1st)
- Employee leave data (.csv): employee_leave_data.csv (daily at 7:00 UTC)
- Message data (JSON): streamed messages from Kafka (real-time)
- Reserved word list (JSON): marked_word.json
- Vocabulary list (JSON): vocab.json

3.2 Data Processing and Storage

- **Employee Data:**
  - Create an append-only table with employee ID, age, and name (scheduled at 7:00 UTC daily).
- **Employee Timeframe Data:**
  - Create an incremental table with employee ID, start date, end date, designation, salary, and status.
    - Handle duplicates by keeping the record with the highest salary and corresponding designation.
    - Convert timestamps to dates.
    - Mark ongoing designations as "ACTIVE" and others as "INACTIVE".
    - Ensure continuity between records for the same employee.
  - Schedule the pipeline to run at 7:00 UTC daily.
- **Employee Leave Data:**
  - Create separate yearly append-only tables for leave quota and leave calendar data.
  - Create a daily updated table (at 7:00 UTC) for tracking leave applications.
- **Employee Reporting:**
  - Generate a daily table (at 7:00 UTC) showing currently active employees by designation and count.
  - Identify employees with potential leave exceeding 8% of working days (excluding holidays and weekends, ignoring duplicates and cancellations). This table should

be updated by 7:00 UTC daily.
    - Create a monthly report (on the 1st at 7:00 UTC) identifying employees who used more than 80% of their leave quota. Generate a text file for each manager with the details (no actual emails). Ensure no duplicate reports are generated if the job fails.
- **Employee Communication Monitoring:**
    - Design a streaming system that flags messages containing reserved words from the marked_word.json file.
    - Maintain a history of flagged messages and employee communication activity.
    - Track the number of flagged messages sent and received by each employee.
    - Deduct 10% salary for each flagged message (represented by a separate column).
    - Implement a monthly cooldown period where strikes are removed, and salaries are restored.
    - Flag employees with 10 strikes as "INACTIVE" and exclude them from the cooldown process.

4. Technical Requirements

- The system should be built following Data Warehouse (DWH) principles for scalability and fault tolerance.
- Utilize AWS services for data storage, processing, and streaming.
- Use an EC2 instance to install a database if necessary.
- Develop an Entity-Relationship (ER) diagram for the system design.

5. Success Criteria

- Successful data ingestion from all sources.
- Accurate processing and storage of employee data.
- Generation of reports with employee leave information and potential misuse cases.
- Functional employee communication monitoring system with strike tracking and salary deductions.

6. Non-Requirements

- Sending actual emails for leave quota reports.
- Implementing functionalities beyond the scope of this document.

This BRD provides a high-level overview of the project requirements. Further technical details and design specifications will be documented during the development phase.