**1. Overview**

The **ACHCR\_Validation** component is a scheduled Java application designed to process ACH (Automated Clearing House) file records stored in a database. It periodically updates the file status, generates return files, and logs the processing outcomes. The application integrates with an Oracle database and manages ACH file states through a sequence of validations and file generation tasks.

The component will be packaged as a JAR file and deployed in a **Tomcat Server** environment for scheduled execution.

**2. Functional Specifications**

**Key Responsibilities**:

1. Query the ACHHEADER table to identify records with FILE\_STATUS = 10.
2. Update the file's status to indicate processing progress.
3. Generate a return file for each record and log the result.
4. Update the database with the final processing status and the generated return file name.

**3. Technical Specifications**

**3.1 Prerequisites**

* **Environment**: Java 1.8+, Oracle 12c, Apache Tomcat 9+
* **Dependencies**:
  + JDBC Driver for Oracle (ojdbc8.jar)
  + Java Timer API
* **Configuration**:
  + Database connection settings (DB\_URL, USER, PASS)
  + File output directory path (/output/path/)

**3.2 Classes**

**1. ACHCR\_Validation**

* **Purpose**: Main class to execute scheduled tasks every 2 minutes.
* **Responsibilities**:
  + Connect to the database.
  + Fetch rows with FILE\_STATUS = 10.
  + Process each row by invoking methods to generate return files and update the database.

**2. ReturnFileGenerator\_seq**

* **Purpose**: Generates return files for processed ACH records.
* **Responsibilities**:
  + Creates a .txt file with the return data for a given ACHHEADER record.
  + Ensures the file directory exists and handles file creation exceptions.

**3.3 Database Interactions**

* **Table**: ACHHEADER
* **Key Columns**:
  + ID (Primary Key)
  + FILE\_NAME
  + FILE\_STATUS (10: Pending, 11: Processing, 12: Success, 13: Failure)
  + LAST\_STATUS\_TIME
  + RETURN\_FILENAME

**3.4 Workflow**

1. **Query ACHHEADER**:
   * Fetch records where FILE\_STATUS = 10.
2. **Update Status to Processing**:
   * Set FILE\_STATUS = 11 and update LAST\_STATUS\_TIME.
3. **Generate Return File**:
   * Invoke ReturnFileGenerator\_seq to create a .txt file.
4. **Update Final Status**:
   * On success, set FILE\_STATUS = 12 and update RETURN\_FILENAME.
   * On failure, set FILE\_STATUS = 13.

**4. Deployment Instructions**

1. **Build and Package**:
   * Use Maven/Gradle to compile and package the code into a JAR file (e.g., ACHCR\_Validation.jar).
2. **Tomcat Deployment**:
   * Place the JAR file in the **lib** folder of the Tomcat server.
   * Include a scheduled job configuration (if using an external scheduler) or ensure the application is set to run continuously.
3. **Configuration Files**:
   * Ensure a properties/configuration file (e.g., db.properties) is included for database connection and file paths.
4. **Logging**:
   * Enable log output for troubleshooting and monitoring purposes using System.out or integrate with logging frameworks (e.g., Log4j).

**5. Configuration Details**

**Example db.properties**:

properties

Copy code

DB\_URL=jdbc:oracle:thin:@localhost:1521:orcl

USER=SYS as SYSDBA

PASS=Root@123

OUTPUT\_FILE\_PATH=/output/path/

**6. Key Error Handling**

1. **Database Connectivity**:
   * Log connection issues and retry on failure.
2. **File Generation**:
   * Handle IOException and return null if file creation fails.
3. **Data Integrity**:
   * Ensure database updates are atomic (use transactions if necessary).

**7. Post-Deployment Validation**

* Verify connectivity to the Oracle database.
* Check log files for successful record processing and return file generation.
* Confirm that records are updated correctly in the ACHHEADER table.

**End of Document**