**Prompt:** {transcript} go through the transcript carefully and generate a requirement document for this. The requirement document should be in two part . The first part should contain the requirements and next part should contain the gherkin's user acceptance along with user stories. Skip the technical part from it.

**Banking Admin Application – Requirement Document**

**Business & Functional Requirements**

# **1. Objective**

Create a secure, admin-driven banking application that allows authorized administrators to manage user accounts, view transaction histories, perform deposits and withdrawals, and execute fund transfers between accounts. Version 1 will focus exclusively on the admin interface and essential back-office workflows.

# **2. Scope**

* Included in v1:
  + Admin authentication and secure access.
  + Account management (view, create, update status).
  + Transaction history per account.
  + Deposits and withdrawals (cash-based).
  + Transfers between accounts with validations and confirmation.
* Excluded from v1:
  + Customer-facing portal.
  + Multi-role access control (only ADMIN role).
  + Multi-currency transactions.
  + Mobile or third‑party integrations.

# **3. User Roles**

* Admin:  
  Can authenticate, view accounts, create new ones, post deposits and withdrawals, perform transfers, and review transaction records.

# **4. Authentication and Access Control**

* SSO login is the default authentication path.
* Fallback JWT login may be used where SSO is unavailable.
* All operations require authenticated admin access.

# **5. Core Functional Features**

# 5.1 Login

* Admins can log in securely using SSO or fallback credentials.
* Unauthorized users are denied access.

# 5.2 Dashboard

* Displays a searchable list of all accounts with summary information: account number, owner name, balance, and status.

# 5.3 Accounts Management

* Admins can:
  + View and search accounts by name, number, or status.
  + Create new accounts (optionally with an initial deposit).
  + Modify status (Active / Frozen).

# 5.4 Transactions

* Displays all account transactions.
* Filter by date, type (Debit / Credit / Deposit / Withdrawal), or amount.
* Each transaction shows date, type, amount, and reference.

# 5.5 Deposits and Withdrawals (Cash Operations)

* Admins can record cash deposits and cash withdrawals on accounts.
* Validations:
  + Deposit increases the account balance.
  + Withdrawal decreases the balance only if sufficient funds are available.
* Each operation is logged as a transaction entry (Deposit = Credit, Withdrawal = Debit) and reflected in the account balance immediately.

# 5.6 Transfers

* Admins can transfer funds between accounts.
* Validations:
  + Both accounts must share the same currency.
  + Self‑transfer is not permitted.
  + The source account must have sufficient balance.
  + Each transfer requires a unique idempotency reference to avoid duplication.
* The process creates matching debit and credit transaction entries for both accounts.

# **6. Non‑Functional Requirements**

* Security: All communication via HTTPS.
* Reliability: Prevent duplicate postings or partial transfers.
* Traceability: Every transaction should have a unique reference for review.
* Performance: Each operation should complete quickly for admin workflows.
* Error Handling: Display user‑friendly messages without technical detail.

# **7. Constraints**

* Only ADMIN role.
* Same‑currency operations only.
* No customer self‑service actions.
* No audit trails in this version.

**User Stories and Gherkin Acceptance Criteria**

## **User Story 1: Admin Authentication**

As an admin, I want to log in securely so I can access banking operations.

**Acceptance Criteria:**

Scenario:

Given valid SSO credentials

When the admin logs in through the SSO

Then access to the admin dashboard is granted

Scenario:

Given valid fallback credentials

When SSO is unavailable

Then login with fallback credentials is accepted

Scenario:

Given invalid credentials

When login is attempted

Then the system rejects the request with an appropriate error message

## User Story 2: View and Search Accounts

As an admin, I want to see and search all bank accounts efficiently.

Acceptance Criteria:

Scenario:

Given an authenticated admin

When they open the dashboard

Then the list of accounts with balances appears

Scenario:

Given a search filter

When applied

Then results show only accounts that match filters

## User Story 3: View Account Detail

As an admin, I want to see the full detail and transaction history of each account.

**Acceptance Criteria:**

Scenario:

Given an admin clicks an account

When the page opens

Then details and transactions are displayed with amounts and dates

## User Story 4: Create a New Account

As an admin, I want to create bank accounts for new customers.

**Acceptance Criteria:**

Scenario:

Given the admin opens the new‑account form

When required details are submitted

Then a new account is created with default Active status

## User Story 5: Update Account Status

As an admin, I want to freeze or activate accounts.

**Acceptance Criteria:**

Scenario:

Given an admin opens an account record

When status is changed to Frozen

Then the system blocks new transactions on that account

Scenario:

Given status is changed to Active

Then normal activity is permitted again

## User Story 6: View Transaction History

As an admin, I want to filter transactions for review.

**Acceptance Criteria:**

Scenario:

Given an admin views an account’s transactions

When date or type filters are used

Then results show only matching transactions

## User Story 7: Deposit Funds (Cash)

As an admin, I want to record a cash deposit into a user account.

**Acceptance Criteria:**

Scenario:

Given an admin enters a valid deposit amount

When the deposit is submitted

Then the account balance increases by that amount

And a Deposit transaction is recorded

## User Story 8: Withdraw Funds (Cash)

As an admin, I want to record a cash withdrawal from a user account.

**Acceptance Criteria:**

Scenario:

Given an admin selects an account with sufficient balance

When a valid withdrawal amount is entered

Then the account balance decreases accordingly

And a Withdrawal transaction is created

Scenario:

Given the balance is insufficient

Then the system blocks the withdrawal and shows an error

## User Story 9: Perform Transfers

As an admin, I want to transfer funds securely between accounts.

**Acceptance Criteria:**

Scenario:

Given valid source and destination accounts with same currency

When a valid transfer amount is confirmed

Then balances update on both accounts accurately

Scenario:

Given a duplicate transfer with the same reference

Then the system rejects it as duplicate

Scenario:

Given mismatched currencies or same account

Then system prevents the transfer and shows a validation error

## User Story 10: Error Visibility

As an admin, I want consistent and clear error responses.

**Acceptance Criteria:**

Scenario:

Given any user operation fails

Then the system returns a message with a clear code and human‑readable explanation

And no technical information is exposed