|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Software Design Specifications**  ***Healthcare Asset Tokenization System***  **Version: [1.8]**   |  |  | | --- | --- | | Project Code | F25-269 | | Supervisor | Dr. Shahbaz Siddiqui | | Co Supervisor | - | | Project Team | Alishba Hassan, Nimil Zubair, Ayesha Nasir | | Submission Date | 07-12-2025 | |     **Document History**   |  |  |  |  | | --- | --- | --- | --- | | Version | Name of Person | Date | Description of change | | 1.0 | Ayesha Nasir | *02/10/25* | Document Created | | 1.1 | Nimil Zubair | *03/10/25* | ERD design | | 1.2 | Alishba Hassan | *06/10/25* | Relations between entities | | 1.3 | Nimil Zubair | *20/10/25* | Data Dictionary for all tables | | 1.4 | Ayesha Nasir | *23/10/25* | Sequence Diagram 1 | | 1.5 | Alishba Hassan | *26/10/25* | Sequence Diagram 2  Sequence Diagram 3 | | 1.6 | Ayesha Nasir | *07/11/25* | State Diagram 1,2 | | 1.7 | Alishba Hassan | *01/12/25* | State Diagram 3,5 | | 1.8 | Nimil Zubair | *02/12/25* | State Diagram 4 |         **Distribution List**   |  |  | | --- | --- | | **Name** | **Role** | | Dr.Shahbaz Siddiqui | Supervisor | | - | Co Supervisor | |  |  |       **Document Sign-Off**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Version** | **Sign-off Authority** | **Project Role** | **Signature** | **Sign-off Date** | | 1.8 | Supervisor | Supervisor | Dr. Shahbaz | 4/12/2025 | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

**Document Information**

| **Category** | **Information** |
| --- | --- |
| Customer | FAST-NU |
| Project | Healthcare Asset Tokenization System |
| Document | Software Design Specification |
| Document Version | 1.8 |
| Status | Draft |
| Author(s) | Ayesha Nasir  Alishba Hassan  Nimil Zubair |
| Approver(s) | Dr. Shahbaz Siddiqui |
| Issue Date |  |
| Document Location |  |
| Distribution | Advisor  Project Coordinator’s Office (through Advisor) |

**Definition of Terms, Acronyms and Abbreviations**

| **Term** | **Description** |
| --- | --- |
| AT | Asset Token |
| HT | Health Token |
| API | Application Programming Interface |
| DApp | Decentralized Application |
| Blockchain | Distributed Ledger for tokenization |
| Patient | End user depositing assets or using subscription card |
| Hospital Admin | Manages tokenization, trades, benefits |
| Bank Officer | Verifies assets physically |

**Table of Contents**

[1 Introduction 9](#_Toc216031330)

[1.1 Purpose of Document 9](#_Toc216031331)

[1.2 Intended Audience 9](#_Toc216031332)

[1.3 Document Convention 9](#_Toc216031333)

[1.4 Project Overview 9](#_Toc216031334)

[1.5 Scope 10](#_Toc216031335)

[2 Design Considerations 11](#_Toc216031336)

[2.1 Assumptions and Dependencies 11](#_Toc216031337)

[2.2 Risks and Volatile Areas 11](#_Toc216031338)

[3 System Architecture 12](#_Toc216031339)

[3.1 System Level Architecture 12](#_Toc216031340)

[3.2 Software Architecture 13](#_Toc216031341)

[4 Design Strategy 14](#_Toc216031342)

[5 Detailed System Design 15](#_Toc216031343)

[5.1 Database Design 15](#_Toc216031344)

[5.1.1 ER Diagram 15](#_Toc216031345)

[5.1.2 Data Dictionary 15](#_Toc216031346)

[5.1.2.1 Data 1 15](#_Toc216031347)

[5.1.2.2 Data 2 16](#_Toc216031348)

[5.1.2.3 Data 3 17](#_Toc216031349)

[5.1.2.4 Data 4 18](#_Toc216031350)

[5.1.2.5 Data 5 19](#_Toc216031351)

[5.1.2.6 Data 6 20](#_Toc216031352)

[5.1.2.7 Data 7 21](#_Toc216031353)

[5.1.2.8 Data 8 22](#_Toc216031354)

[5.2 Application Design 25](#_Toc216031355)

[5.2.1 Sequence Diagram 25](#_Toc216031356)

[5.2.1.1 Asset Deposit and Token Minting 25](#_Toc216031357)

[5.2.1.2 Bank Verification Process 26](#_Toc216031358)

[5.2.1.3 Trading and Benefits Redemption 27](#_Toc216031359)

[5.2.2 State Diagram 28](#_Toc216031360)

[5.2.2.1 Asset Deposit and Token Minting 28](#_Toc216031361)

[5.2.2.2 Bank Verification 29](#_Toc216031362)

[5.2.2.3 Trading and Benefits Redemption 30](#_Toc216031363)

[5.2.3 Data Flow Diagram 31](#_Toc216031364)

[DFD Level 1A - Asset Deposit & Tokenization: 32](#_Toc216031366)

[DFD Level 1B - Trading & Benefits Redemption: 33](#_Toc216031368)

[6 Reference 34](#_Toc216031369)

[7 Appendices 35](#_Toc216031370)

# Introduction

## Purpose of Document

*This Software Design Specification (SDS) describes the complete* ***architectural, database, component-level,*** *and* ***interaction-level*** *design of the* Healthcare Asset Tokenization and Benefits Distribution System*.*

*This system allows patients to:*

* *Deposit real-world assets*
* *Convert them into Asset Tokens (AT)*
* *Receive Health Tokens (HT) from hospital profits*
* *Use HT for medical benefits, discounts, or insurance-like coverage*

*The SDS explains:*

* *How each subsystem interacts*
* *Database structure*
* *Smart contract roles*
* *Module-level designs*
* *Sequence & state diagrams*
* *Technical decisions taken in alignment with the SRS*

*This document ensures all developers can follow a consistent blueprint for system implementation.*

## Intended Audience

* *Fast NU*
* *Jury*
* *Supervisor (Dr. Shahbaz Siddiqui)*
* *Students of Fast NU*
* *Our Team(Designer, Developer, Tester)*
* *Potential Users of this product*

## Document Convention

* *Font Family = Arial*
* *Font Size = 14 and 12 for headings, 10 for the rest of the content*

## Project Overview

*The Healthcare Asset Tokenization System is a blockchain-powered financial health solution enabling patients to deposit assets or pay monthly subscriptions. Hospitals convert verified assets into blockchain-based Asset Tokens (AT).*

*Hospitals trade AT using a simulated trading API to generate profits.  
Profits are distributed to patients in the form of Health Tokens (HT).*

*Patients can redeem HT for:*

* *OPD discounts*
* *Medication vouchers*
* *Medical checkups*
* *Emergency medical coverage*
* *Insurance-like annual benefits*

## Scope

***Included Functionalities***

* *Patient asset deposit & subscription module*
* *Hospital admin portal (mint AT, trade, allocate HT)*
* *Bank verification module*
* *Blockchain-based token lifecycle*
* *Medical benefits engine*
* *Smart contract integration*
* *Trading API simulation*
* *Transaction logging*
* *Role-based access control*

***Excluded Functionalities***

* *Physical verification of assets (handled manually by bank)*
* *Automated checking of gold values (manually entered)*
* *Real-world insurance claim automation*
* *Actual stock market trading (only simulated via API)*

# Design Considerations

## Assumptions and Dependencies

* *Bank provides manual verification, system only stores result*
* *Blockchain network available 24/7*
* *Smart contract deployment environment active (Ethereum testnet)*
* *Hospital Admin has trading privileges*
* *Users (patients) understand basic digital dashboards*
* *API services must respond within reasonable limits*

## Risks and Volatile Areas

|  |  |  |
| --- | --- | --- |
| *Risk* | *Description* | *Mitigation* |
| *Blockchain congestion* | *Slower transactions* | *Use async confirmations* |
| *Trading API failure* | *Profit calculation blocked* | *Store & retry strategy* |
| *Incorrect asset valuation* | *Because manual entry* | *Require 2-step hospital+bank approval* |
| *Smart contract bug* | *Could freeze tokens* | *Multiple audits & testnets* |
| *Data privacy* | *Medical + financial data* | *Encrypt sensitive fields* |

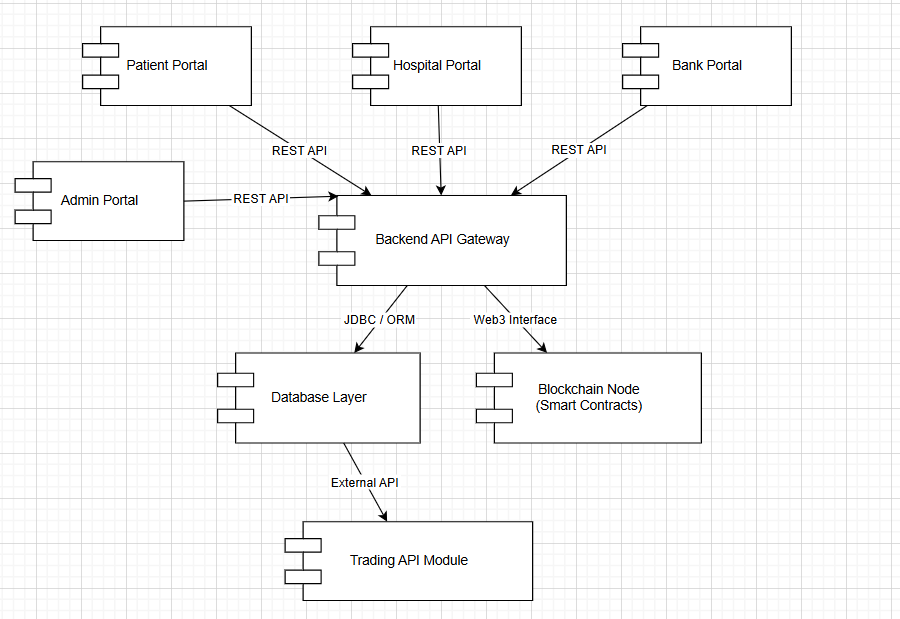
# System Architecture

## System Level Architecture

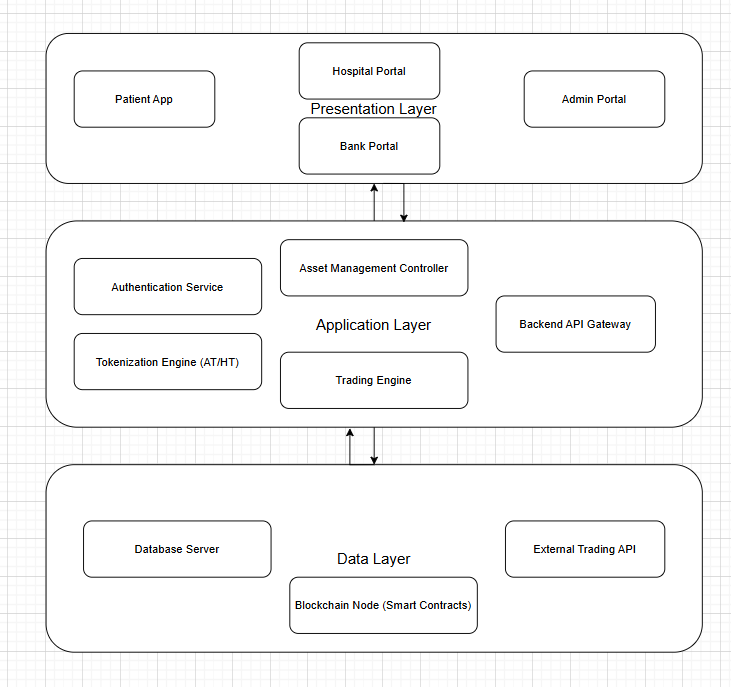
***Subsystems***

* *Patient Portal*
* *Hospital Portal*
* *Bank Portal*
* *Admin Portal*
* *Blockchain Module*
* *Trading API Integration*
* *Authentication Server*
* *Benefits & HT Engine*
* *Database Layer*

***Diagram***

******

## Software Architecture

**

# Design Strategy

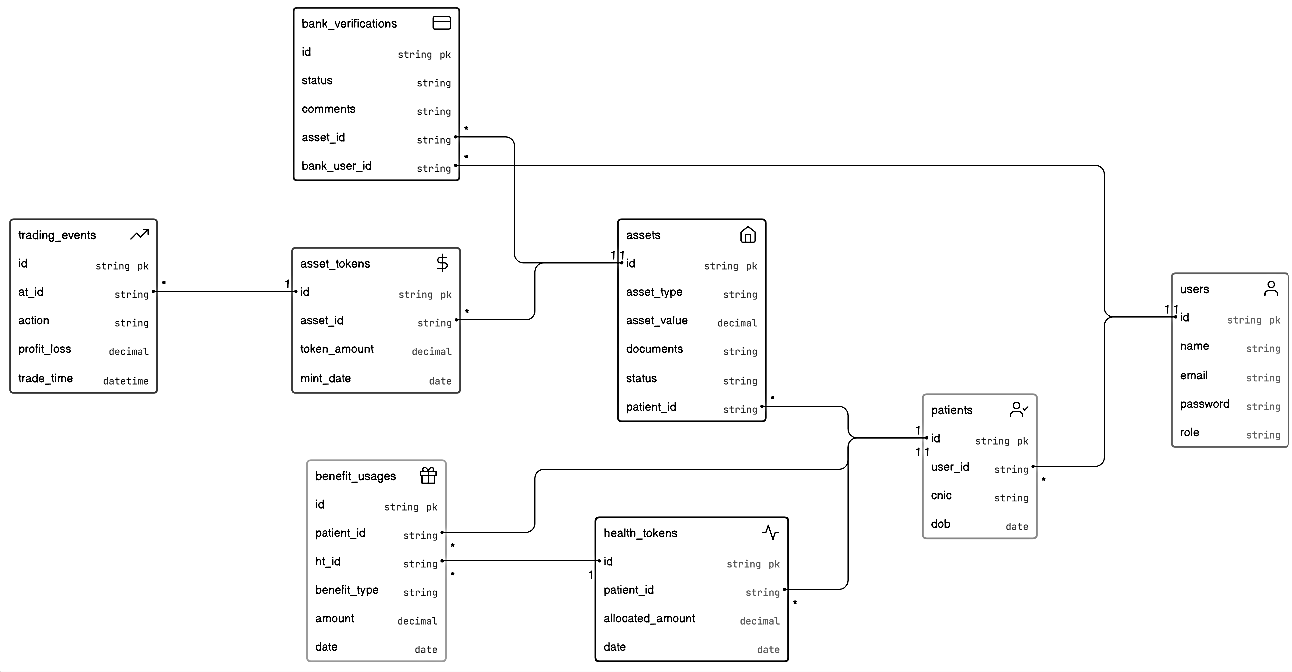
*The following design strategies were used to ensure a more robust and effective software structure:*

* *Modular microservice-like structure*
* *Smart contracts store immutable token logic*
* *API-first approach for extensibility*
* *DB normalization to avoid redundancy*
* *Use Web3.js/Ethers.js to interact with smart contracts*
* *Diagrams included to maintain clarity*
* *Use DTOs and Controllers for clean separation*
* *Use JWT for secure access handling*

# Detailed System Design

## Database Design

### ER Diagram



***Logical Schema:***

|  |  |  |
| --- | --- | --- |
| Entity | Relationship | Cardinality |
| *User — Patient* | *One user becomes a patient profile* | *1–1* |
| *Patient — Asset* | *A patient can deposit many assets* | *1–M* |
| *Asset — Bank Verification* | *Every asset must be verified* | *1–1* |
| *Asset — Asset Record* | *One asset produces one AT record* | *1–1* |
| *Asset Token — Trading Event* | *Many Asset Tokens can have many trades* | *M–M* |
| *Patient — Health Token* | *A Patient receive multiple HT allocations* | *1–M* |
| *Health Token — Benefit Usage* | *HT is consumed in many benefit events* | *1–M* |

### Data Dictionary

#### Data 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Documents** | | | | | | | |
| **Name** | | Patient Documents | | | | | |
| **Alias** | | KYC Docs | | | | | |
| **Where-used/how-used** | | Used to store KYC documents such as CNIC images, medical files, and ownership documents submitted by the patient. | | | | | |
| **Content description** | | Holds document type, file location and verification details. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| document\_id | Document ID | | UUID | — | NO | None | PK |
| patient\_id | Linked patient | | UUID | — | NO | None | FK |
| documentType | Type (CNIC, Ownership, Medical, etc.) | | VARCHAR | 100 | NO | None |  |
| file\_url | URL of uploaded file | | VARCHAR | 255 | NO | None |  |
| uploadedAt | Upload timestamp | | TIMESTAMP | — | NO | now() |  |
| verifiedAt | Verified timestamp | | TIMESTAMP | — | YES | NULL |  |
| rejectionReason | Reason if rejected | | TEXT | — | YES | NULL |  |

#### Data 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Patient** | | | | | | | |
| **Name** | | Patient | | | | | |
| **Alias** | | User Patient Profile | | | | | |
| **Where-used/how-used** | | Patients use the platform to submit assets, complete KYC, subscribe to plans, view health tokens, redeem benefits, make payments, and receive notifications. | | | | | |
| **Content description** | | Stores complete patient personal, contact, medical and system-level metadata. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| id | Unique Patient ID | | UUID | — | NO | None | PK |
| name | Patient full name | | VARCHAR | 255 | NO | None |  |
| phone\_num | Patient phone number | | VARCHAR | 20 | NO | None |  |
| email | Patient email | | VARCHAR | 255 | NO | None |  |
| address | Residential address | | TEXT | — | YES | NULL |  |
| hospital\_id | Linked hospital | | UUID | — | YES | NULL | FK |
| has\_asset | Whether patient has active asset deposits | | BOOLEAN | — | NO | false |  |
| has\_subscription | Whether patient has subscription | | BOOLEAN | — | NO | false |  |
| dob | Date of birth | | DATE | — | NO | None |  |
| blood\_group | Blood type | | VARCHAR | 5 | YES | NULL |  |
| kyc\_status | pending/approved/rejected | | VARCHAR | 30 | NO | pending |  |
| createdAt | Creation time | | TIMESTAMP | — | NO | now() |  |
| walletAddress | Blockchain wallet address | | VARCHAR | 255 | YES | NULL |  |
| mfa | Multi-factor enabled | | BOOLEAN | — | NO | false |  |

.

.

#### Data 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **KYC** | | | | | | | |
| **Name** | | KYC Record | | | | | |
| **Alias** | | Patient KYC Status | | | | | |
| **Where-used/how-used** | | Tracks KYC submission and approval process for each patient. | | | | | |
| **Content description** | | Stores completion %, timestamps, status and rejection reasons. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| kyc\_id | KYC ID | | UUID | — | NO | None | PK |
| patient\_id | Linked patient | | UUID | — | NO | None | FK |
| completionPercentage | % progress | | INT | — | NO | 0 |  |
| submittedAt | KYC submitted timestamp | | TIMESTAMP | — | YES | NULL |  |
| status | pending/approved/rejected | | VARCHAR | 30 | NO | pending |  |
| rejectionReason | Reason if rejected | | TEXT | — | YES | NULL |  |

#### Data 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AssetDeposit** | | | | | | | |
| **Name** | | Asset Deposit | | | | | |
| **Alias** | | Patient Asset Submission | | | | | |
| **Where-used/how-used** | | Represents assets deposited by patients (gold, land, property) before verification and tokenization. | | | | | |
| **Content description** | | Stores asset metadata, value, weight, and approval–rejection lifecycle. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| deposit\_id | Deposit ID | | UUID | — | NO | None | PK |
| patient\_id | Patient who submitted | | UUID | — | NO | None | FK |
| assetType | Gold/Land/Property | | VARCHAR | 50 | NO | None |  |
| assetValue | Declared value | | DECIMAL | 18,2 | NO | None |  |
| weight | Weight (for gold) | | DECIMAL | 10,2 | YES | NULL |  |
| status | submitted/pending/approved/rejected | | VARCHAR | 30 | NO | submitted |  |
| submittedAt | When submitted | | TIMESTAMP | — | NO | now() |  |
| approvedAt | Approval time | | TIMESTAMP | — | YES | NULL |  |
| rejectedAt | Rejection time | | TIMESTAMP | — | YES | NULL |  |
| rejectionReason | Reason if rejected | | TEXT | — | YES | NULL |  |

#### Data 5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Patient Token Balance** | | | | | | | |
| **Name** | | Patient Token Balance | | | | | |
| **Alias** | | Token Wallet Summary | | | | | |
| **Where-used/how-used** | | Tracks total AT (Asset Tokens) and HT (Health Tokens) associated with a patient. | | | | | |
| **Content description** | | Holds aggregated balances and update timestamps. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| balance\_id | Balance ID | | UUID | — | NO | None | PK |
| patient\_id | Linked patient | | UUID | — | NO | None | FK |
| totalAT | Total Asset Tokens | | DECIMAL | 20,8 | NO | 0 |  |
| totalHT | Total Health Tokens | | DECIMAL | 20,8 | NO | 0 |  |
| lastUpdated | Timestamp | | TIMESTAMP | — | NO | now() |  |

#### Data 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Transaction** | | | | | | | |
| **Name** | | Transaction | | | | | |
| **Alias** | | System Transaction Log | | | | | |
| **Where-used/how-used** | | Logs all on-chain and off-chain financial actions by any user. | | | | | |
| **Content description** | | Tracks token movement, payments, sender/receiver wallet addresses and blockchain metadata. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| transaction\_id | Transaction ID | | UUID | — | NO | None | PK |
| user\_id | Actor (patient/admin/hospital/staff) | | UUID | — | NO | None | FK |
| timestamp | Time | | TIMESTAMP | — | NO | now() |  |
| type | type of transaction | | VARCHAR | 50 | NO | None |  |
| token\_id | Token involved | | UUID | — | YES | NULL | FK |
| amount | Amount | | DECIMAL | 18,2 | YES | NULL |  |
| description | Details | | TEXT | — | YES | NULL |  |
| senderWalletAddress | Sender wallet | | VARCHAR | 255 | YES | NULL |  |
| receiverWalletAddress | Receiver wallet | | VARCHAR | 255 | YES | NULL |  |
| blockNumber | Blockchain block number | | BIGINT | — | YES | NULL |  |
| status | success/pending/failed | | VARCHAR | 30 | NO | pending |  |

#### Data 7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Subscription Plan (Hospital)** | | | | | | | |
| **Name** | | Subscription Plan | | | | | |
| **Alias** | | Hospital Subscription Model | | | | | |
| **Where-used/how-used** | | Hospitals create subscription plans offered to patients. | | | | | |
| **Content description** | | Stores name, monthly fee and list of benefits. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| subs\_id | Plan ID | | UUID | — | NO | None | PK |
| hospital\_id | Hospital offering plan | | UUID | — | NO | None | FK |
| subscription\_name | Name of subscription | | VARCHAR | 255 | NO | None |  |
| amount\_per\_month | Monthly cost | | DECIMAL | 18,2 | NO | None |  |
| features | JSON/Text list of plan perks | | TEXT | — | YES | NULL |  |

#### Data 8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Patient Subscription** | | | | | | | |
| **Name** | | Patient Subscription | | | | | |
| **Alias** | | Subscription Request | | | | | |
| **Where-used/how-used** | | Tracks which patient enrolled in which plan and subscription lifecycle. | | | | | |
| **Content description** | | Holds activation, expiry and status. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| subs\_req\_id | Subscription Record ID | | UUID | — | NO | None | PK |
| subscription\_id | Linked plan | | UUID | — | NO | None | FK |
| patient\_id | Linked patient | | UUID | — | NO | None | FK |
| startDate | Starting date | | DATE | — | NO | None |  |
| endDate | Ending date | | DATE | — | YES | NULL |  |
| status | active/expired/cancelled | | VARCHAR | 30 | NO | active |  |

* + - 1. **Data 9**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Payment History** | | | | | | | |
| **Name** | | Payment History | | | | | |
| **Alias** | | Patient Payment Log | | | | | |
| **Where-used/how-used** | | Stores subscription payments, fee deductions, and billing records for patients. | | | | | |
| **Content description** | | Tracks payment amount, method, status and invoice file link. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| payment\_id | Payment ID | | UUID | — | NO | None | PK |
| patient\_id | Linked patient | | UUID | — | NO | None | FK |
| timestamp | When paid | | TIMESTAMP | — | NO | now() |  |
| amount | Amount in PKR | | DECIMAL | 18,2 | NO | None |  |
| subs\_id | Subscription | | UUID | — | YES | NULL | FK |
| status | success/pending/failed | | VARCHAR | 30 | NO | pending |  |
| payment\_method | credit card / wallet / bank | | VARCHAR | 50 | YES | NULL |  |
| invoiceUrl | URL of invoice | | VARCHAR | 255 | YES | NULL |  |

* + - 1. **Data 10**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Notification** | | | | | | | |
| **Name** | | Notification | | | | | |
| **Alias** | | System Notification | | | | | |
| **Where-used/how-used** | | Tracks messaging between system entities such as admins, hospitals and patients. | | | | | |
| **Content description** | | Stores sender, receiver, body text, timestamp and read state. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| noti\_id | Notification ID | | UUID | — | NO | None | PK |
| sender\_id | Sender | | UUID | — | NO | None | FK |
| receiver\_id | Receiver | | UUID | — | NO | None | FK |
| notification\_text | Message content | | TEXT | — | YES | NULL |  |
| timestamp | Sent time | | TIMESTAMP | — | NO | now() |  |
| read | Whether read | | BOOLEAN | — | NO | false |  |

* + - 1. **Data 11**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity** | | | | | | | |
| **Name** | | Activity | | | | | |
| **Alias** | | User Activity Log | | | | | |
| **Where-used/how-used** | | Tracks all meaningful actions performed by users inside the platform. | | | | | |
| **Content description** | | Stores name, type, status and metadata like IP address. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| act\_id | Activity ID | | UUID | — | NO | None | PK |
| user\_id | User performing activity | | UUID | — | NO | None | FK |
| activity\_name | Name | | VARCHAR | 255 | NO | None |  |
| description | Short description | | TEXT | — | YES | NULL |  |
| type | Category | | VARCHAR | 100 | YES | NULL |  |
| status | success/failed/pending | | VARCHAR | 30 | NO | success |  |
| timestamp | When | | TIMESTAMP | — | NO | now() |  |
| ipAddress | IP address | | VARCHAR | 45 | YES | NULL |  |

#### Data 12

#### 

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Security Settings** | | | | | | | |
| **Name** | | Security Settings | | | | | |
| **Alias** | | User Security Profile | | | | | |
| **Where-used/how-used** | | Manages MFA, verified email status and security features for each user. | | | | | |
| **Content description** | | Stores security-related toggles and preferences. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| sec\_setting\_id | Security Setting ID | | UUID | — | NO | None | PK |
| user\_id | User | | UUID | — | NO | None | FK |
| multiFactorEnabled | MFA enabled | | BOOLEAN | — | NO | false |  |
| emailVerified | Whether email verified | | BOOLEAN | — | NO | false |  |

#### Data 13

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Patient Card** | | | | | | | |
| **Name** | | Patient Card | | | | | |
| **Alias** | | Health/Subscription Card | | | | | |
| **Where-used/how-used** | | Assigned to patients for redeeming services and storing HT balance. | | | | | |
| **Content description** | | Stores card details and secure info (CVV should be encrypted in implementation). | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| patient\_card\_id | Card ID | | UUID | — | NO | None | PK |
| patient\_id | Linked patient | | UUID | — | NO | None | FK |
| card\_id | Card type | | UUID | — | NO | None | FK |
| card\_num | Card number | | VARCHAR | 20 | NO | None |  |
| cvv | Security code | | VARCHAR | 4 | NO | None |  |
| expiryDate | Expiry | | DATE | — | NO | None |  |
| HTbalance | HT balance on card | | DECIMAL | 20,8 | NO | 0 |  |

* + - 1. **Data 14**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cards** | | | | | | | |
| **Name** | | Cards | | | | | |
| **Alias** | | Card Type | | | | | |
| **Where-used/how-used** | | Defines card types given to patients such as subscription card or asset-health card. | | | | | |
| **Content description** | | Store card metadata. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| card\_id | Card ID | | UUID | — | NO | None | PK |
| card\_name | Name of card type | | VARCHAR | 255 | NO | None |  |

* + - 1. **Data 15**

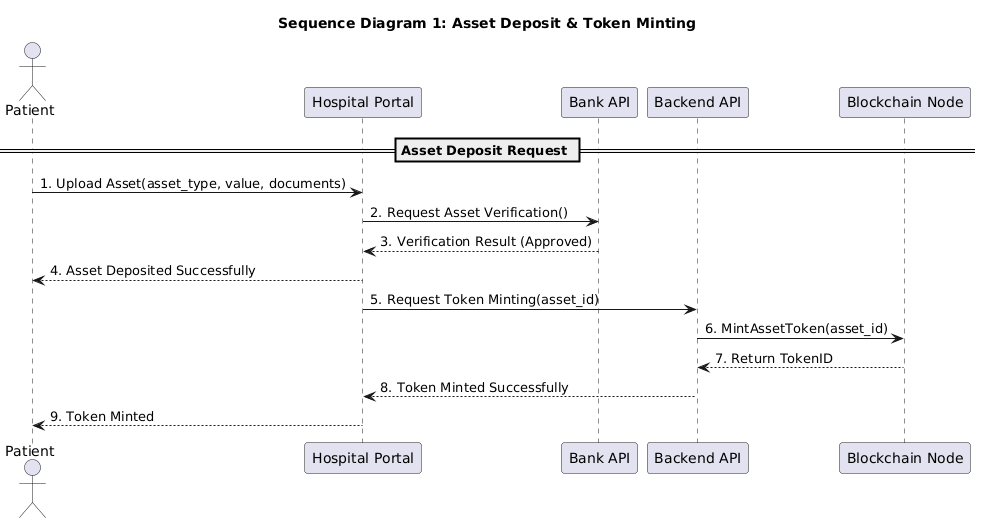
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Token** | | | | | | | |
| **Name** | | Tokens | | | | | |
| **Alias** | | Asset & Health Token Registry | | | | | |
| **Where-used/how-used** | | Used to store all blockchain-minted tokens including Asset Tokens (AT) generated from asset deposits and Health Tokens (HT) generated from trading profits or subscription rewards. Supports wallet balance updates, transfers, redemptions, and transaction logging. | | | | | |
| **Content description** | | Contains token metadata such as type, owner, value, creation time, and blockchain-specific attributes. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Nullable** | **Default Value** | **Key Type** |
| token\_id | Unique token ID | | UUID | — | NO | None | PK |
| patient\_id | Owner of token | | UUID | — | NO | None | FK |
| type | Token type (AT/HT) | | VARCHAR | 10 | NO | None |  |
| amount | Token amount/value | | DECIMAL | 20,8 | NO | 0 |  |
| createdAt | Token creation timestamp | | TIMESTAMP | — | NO | now() |  |
| updatedAt | Last update timestamp | | TIMESTAMP | — | YES | NULL |  |
| status | active/locked/expired | | VARCHAR | 20 | NO | active |  |

## Application Design

* *Detailed Sequence diagram with parameter list*
* *State Transition Diagram*
* *DFD level 1 diagram]*

### Sequence Diagram

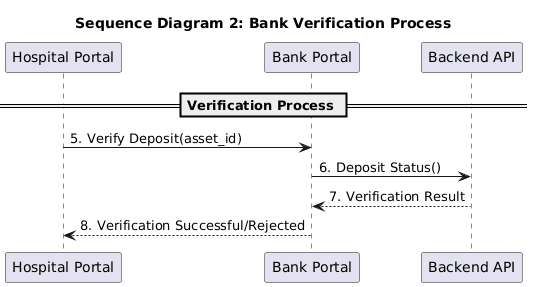
#### Asset Deposit and Token Minting



*The working of this diagram is as followed:*

* *The patient uploads an asset by sending its type, value, and documents to the Hospital Portal.*
* *The Hospital Portal sends the asset details to the Bank API for verification.*
* *The Bank API verifies the asset and returns an approval status to the Hospital Portal.*
* *The Hospital Portal notifies the patient that the asset has been deposited successfully.*
* *The Hospital Portal requests the Backend API to mint tokens for the verified asset.*
* *The Backend API asks the Blockchain to create a digital token for the asset.*
* *The Blockchain creates the token and sends back a Token ID to the Backend API.*
* *The Backend API confirms successful token minting to the Hospital Portal.*
* *The Hospital Portal notifies the patient that tokens have been minted and updates the patient's record.*

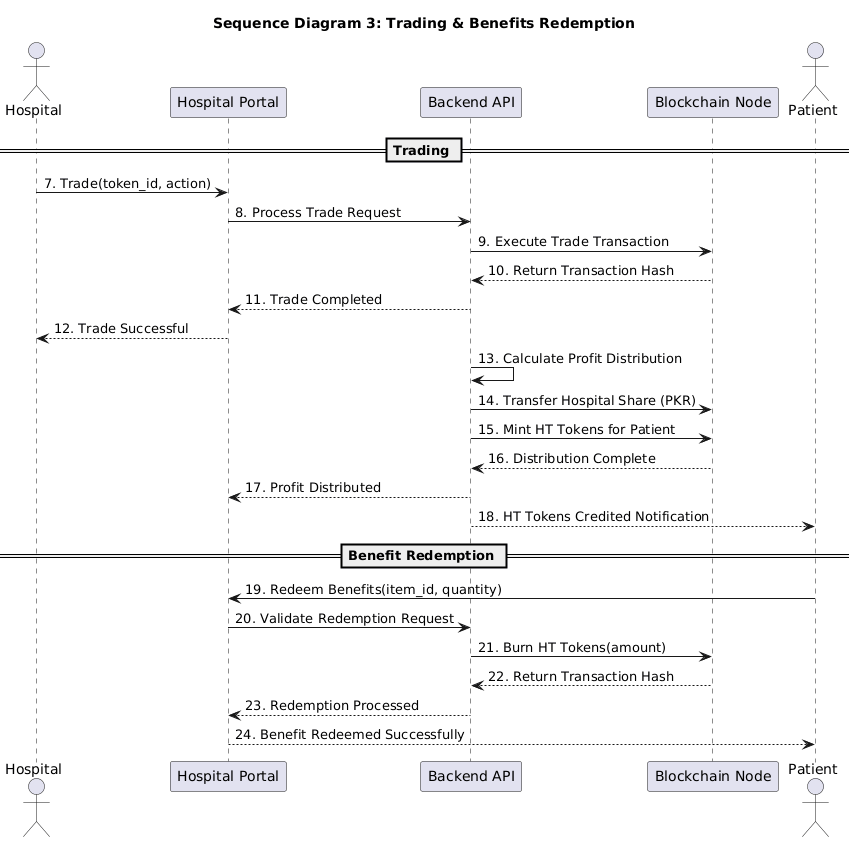
#### Bank Verification Process



*The working of this sequence diagram is as followed:*

* *The Hospital Portal sends the asset to the bank for physical/manual verification.*
* *The Bank officer checks the asset details in real life.*
* *After checking, the Bank updates the verification status in the system (verified/rejected).*
* *The Backend receives the status and updates the asset’s record accordingly.*

#### Trading and Benefits Redemption



• Trading Flow

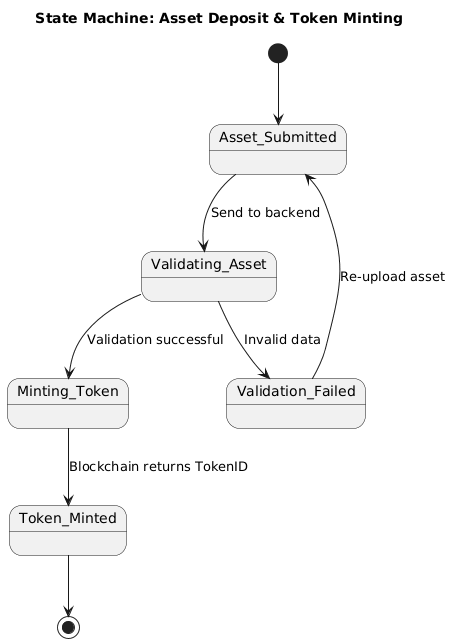
* The hospital initiates a trade (buy/sell) using asset tokens from the Hospital Portal.
* The Hospital Portal sends the trade request to the Backend API.
* The Backend API executes the trade transaction on the Blockchain.
* The Blockchain returns the transaction hash confirming the trade.
* The Backend API calculates profit distribution between hospital and patient.
* The Backend API transfers the hospital's share in PKR to the hospital account.
* The Backend API mints HT tokens for the patient as their profit share.
* The Blockchain confirms the profit distribution is complete.
* The Hospital Portal notifies the hospital that the trade was successful.
* The patient receives a notification that HT tokens have been credited to their account.

• Benefits Redemption Flow

* The patient selects and redeems health benefits (OPD, checkup, discount, etc.) using their HT tokens through the Hospital Portal.
* The Hospital Portal sends the redemption request to the Backend API.
* The Backend API validates the patient's HT balance and redemption eligibility.
* The Backend API burns the required amount of HT tokens on the Blockchain.
* The Blockchain returns the transaction hash confirming the token burn.
* The Backend API confirms the redemption has been processed.
* The Hospital Portal notifies the patient that the benefit has been redeemed successfully.

### State Diagram

#### Asset Deposit and Token Minting



* *The process starts when the patient submits an asset.*
* *System moves to Validating Asset state.*
* *If something is missing/wrong, that means Validation Failed so patient must re-upload.*
* *If everything is correct, the process moves to Minting Token.*
* *Blockchain creates the token and moves to Token Minted state.*
* *Process ends once token is successfully created*

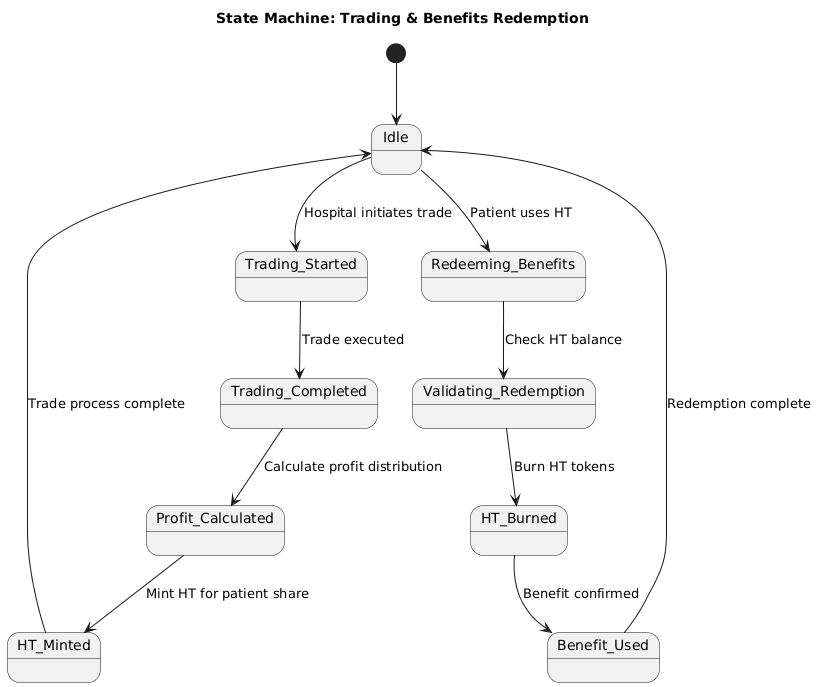
#### Bank Verification

A diagram of a bank process

AI-generated content may be incorrect.

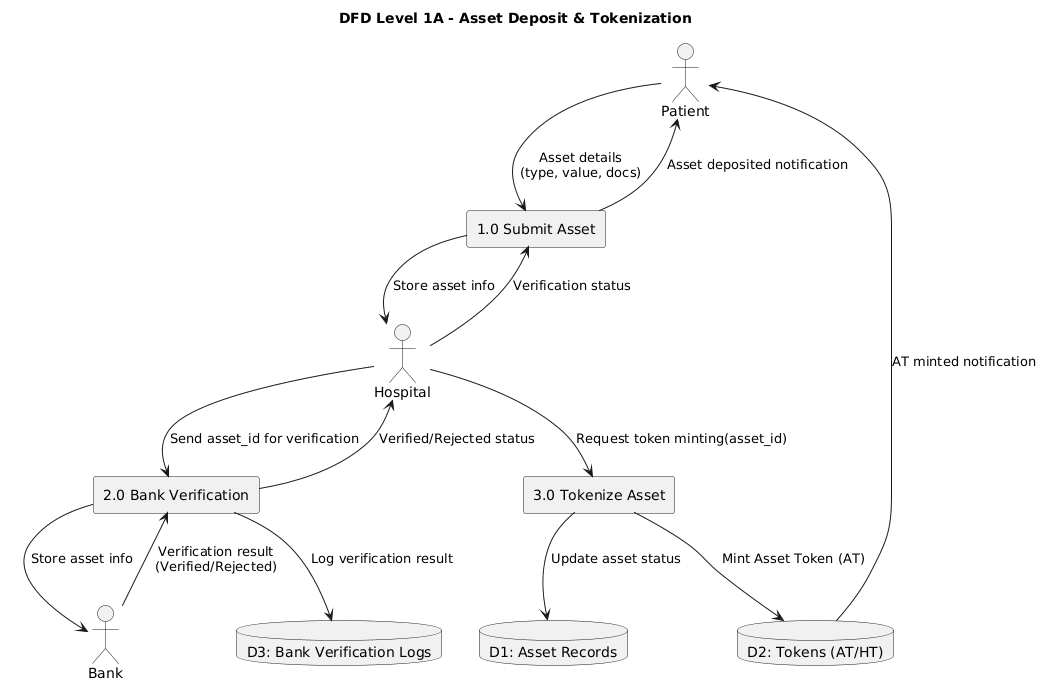
* *When the hospital forwards the asset, it enters Pending Verification.*
* *The bank starts checking and process shifts to Under Review.*
* *If everything matches documents, process shifts to Verified.*
* *If details don’t match or asset fails checks, process shifts to Rejected.*
* *Either Verified or Rejected ends the process.*

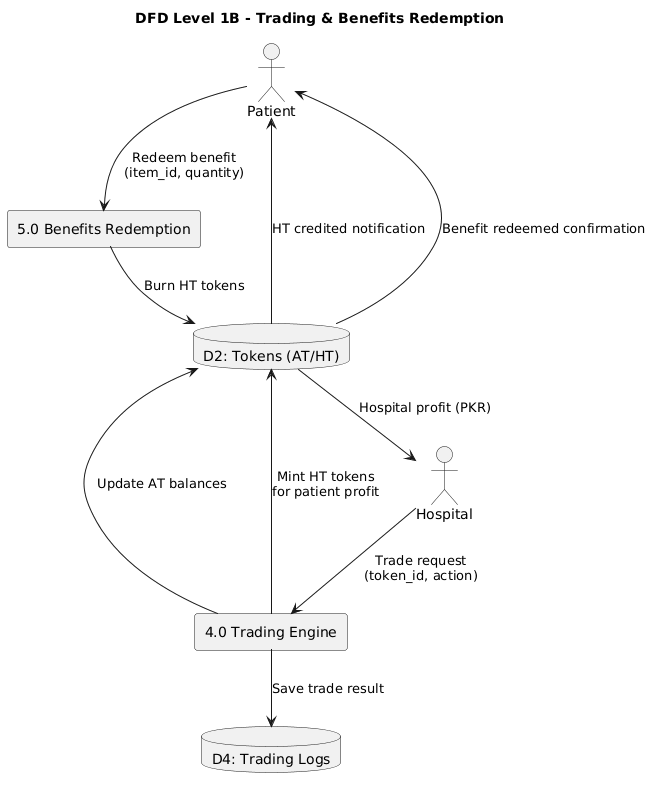
#### Trading and Benefits Redemption



* *Trading Phase*
* *System starts in Idle.*
* *Patient starts a trade and process shifts to Trading Started.*
* *Backend runs simulation and process shifts to Trading Completed.*
* *Based on profit, system calculates HT as it moves to Benefits Calculated and moves back to Idle.*
* *Benefits Redemption Phase*
* *Patient wants to redeem benefits, process moves to Redeeming Benefits.*
* *System checks balance and deducts HT and process shifts to Benefit Used.*
* *Returns to Idle.*

#### Data Flow Diagram





# The DFD Level 1 is divided into two diagrams for clarity:

# DFD Level 1A - Asset Deposit & Tokenization:

# Patients submit their asset details (type, value, documents) through the Hospital Portal, which stores the information in Asset Records. The Hospital Portal sends the asset to the Bank Portal for verification. The Bank verifies the asset and returns the verification result (Verified/Rejected), which is logged in Bank Verification Logs. The Hospital Portal notifies the patient that their asset has been deposited. Once verified, the Hospital requests token minting from the system. The system updates the asset status in Asset Records, mints Asset Tokens (AT) on the blockchain, stores them in the Tokens database, and notifies the patient that AT tokens have been minted successfully.

# DFD Level 1B - Trading & Benefits Redemption:

The Hospital initiates trading using Asset Tokens through the Trading Engine. The system executes the trade, saves the trade result in Trading Logs, and updates AT token balances. After successful trading, the system calculates profit distribution - the hospital receives their share in PKR, while the patient's profit share is converted to Health Tokens (HT) which are minted and credited to their account. The patient receives a notification of HT tokens credited. For benefit redemption, patients select and redeem healthcare benefits (OPD, checkups, discounts) using their HT tokens. The system validates the request, burns the required amount of HT tokens from the blockchain, and confirms the benefit redemption to the patient.

# Reference

* **Ethereum Documentation, Smart Contracts, 2024.**<https://ethereum.org/en/developers/docs/smart-contracts/>
* **Healthcare Tokenization Models – IEEE Journals.**  
  [https://www.researchgate.net/publication/391560069\_A\_Systematic\_Literature\_Review\_for\_Blockchain-Based\_Healthcare\_Implementations](https://www.researchgate.net/publication/391560069_A_Systematic_Literature_Review_for_Blockchain-Based_Healthcare_Implementations?utm_source=chatgpt.com)
* **Web3.js & Ethers.js Documentation.**  
  Web3.js: [https://web3js.readthedocs.io/](https://web3js.readthedocs.io/?utm_source=chatgpt.com)  
  Ethers.js: [https://docs.ethers.org/v6/](https://docs.ethers.org/v6/?utm_source=chatgpt.com)

# Appendices

* 1. *Appendix A: Tokenomics Models*
  2. *Appendix B: AT/HT Conversion Formula*
  3. *Appendix C: Sequence Diagrams*
  4. *Appendix D: Ledger Screenshots*