|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Software Design Specifications**  ***Healthcare Asset Tokenization System***  **Version: [01.08]**   |  |  | | --- | --- | | Project Code | F25-269 | | Supervisor | Dr. Shahbaz Siddiqui | | Co Supervisor | - | | Project Team |  | | Submission Date |  | |   **[Instructions]**   * *No section of template should be deleted. You can write ‘Not applicable’ if a section is not applicable to your project. But all sections must exist in the final document.* * *All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced / removed in final document.* * *This’ Instruction’ section should also be removed in final document.* * *MS-Word Reviewing feature must be used to get the document reviewed by PMs or supervisors.*       **Document History**  *[Revision history will be maintained to keep a track of changes done by anyone in the document.]*   |  |  |  |  | | --- | --- | --- | --- | | Version | Name of Person | Date | Description of change | | 1.0 |  | *02/10/25* | Document Created | | 1.1 |  | *03/10/25* | ERD design | | 1.2 |  | *06/10/25* | Relations between entities | | 1.3 |  | *20/10/25* | Data Dictionary for all tables | | 1.4 |  | *23/10/25* | Sequence Diagram 1 | | 1.5 |  | *26/10/25* | Sequence Diagram 2  Sequence Diagram 3 | | 1.6 |  | *07/11/25* | State Diagram 1,2 | | 1.7 |  | *01/12/25* | State Diagram 3,5 | | 1.8 |  | *02/12/25* | State Diagram 4 |         **Distribution List**  *[Following table will contain list of people whom the document will be distributed after every sign-off]*   |  |  | | --- | --- | | **Name** | **Role** | | Dr.Shahbaz Siddiqui | Supervisor | | - | Co Supervisor | |  |  |       **Document Sign-Off**  *[Following table will contain sign-off details of the document. Once the document is prepared and revised, this should be signed-off by the sign-off authority.*  *Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.]*   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **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**

*[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly. ]*

| **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 10](#_Toc215692752)

[1.1 Purpose of Document 10](#_Toc215692753)

[1.2 Intended Audience 10](#_Toc215692754)

[1.3 Document Convention 10](#_Toc215692755)

[1.4 Project Overview 10](#_Toc215692756)

[1.5 Scope 11](#_Toc215692757)

[2 Design Considerations 12](#_Toc215692758)

[2.1 Assumptions and Dependencies 12](#_Toc215692759)

[2.2 Risks and Volatile Areas 12](#_Toc215692760)

[3 System Architecture 13](#_Toc215692761)

[3.1 System Level Architecture 13](#_Toc215692762)

[3.2 Software Architecture 14](#_Toc215692763)

[4 Design Strategy 15](#_Toc215692764)

[5 Detailed System Design 16](#_Toc215692765)

[5.1 Database Design 16](#_Toc215692766)

[5.1.1 ER Diagram 16](#_Toc215692767)

[5.1.2 Data Dictionary 16](#_Toc215692768)

[5.1.2.1 Data 1 16](#_Toc215692769)

[5.1.2.2 Data 2 17](#_Toc215692770)

[5.1.2.3 Data 3 18](#_Toc215692771)

[5.1.2.4 Data 4 19](#_Toc215692772)

[5.1.2.5 Data 5 20](#_Toc215692773)

[5.1.2.6 Data 6 21](#_Toc215692774)

[5.1.2.7 Data 7 22](#_Toc215692775)

[5.1.2.8 Data 8 23](#_Toc215692776)

[5.2 Application Design 26](#_Toc215692777)

[5.2.1 Sequence Diagram 26](#_Toc215692778)

[5.2.1.1 Asset Deposit and Token Minting 26](#_Toc215692779)

[5.2.1.2 Bank Verification Process 27](#_Toc215692780)

[5.2.1.3 Trading and Benefits Redemption 27](#_Toc215692781)

[5.2.2 State Diagram 28](#_Toc215692782)

[5.2.2.1 Asset Deposit and Token Minting 28](#_Toc215692783)

[5.2.2.2 Bank Verification 29](#_Toc215692784)

[5.2.2.3 Trading and Benefits Redemption 30](#_Toc215692785)

[5.2.3 Data Flow Diagram 31](#_Toc215692786)

[6 References 32](#_Toc215692787)

[7 Appendices 33](#_Toc215692788)

# 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 = 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/land 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 (Polygon 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* |

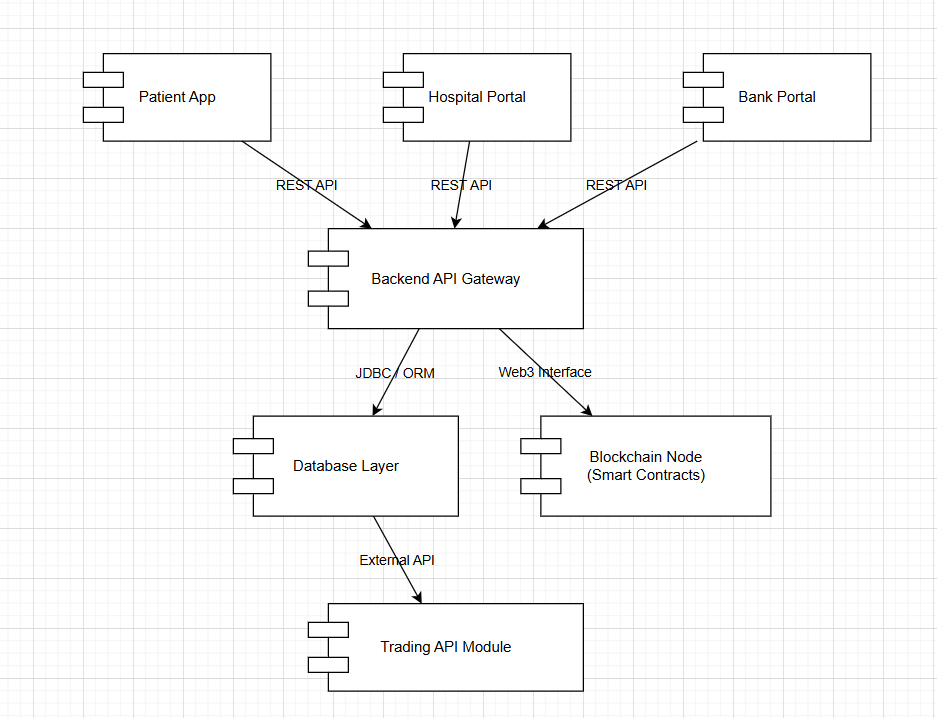
# System Architecture

## System Level Architecture

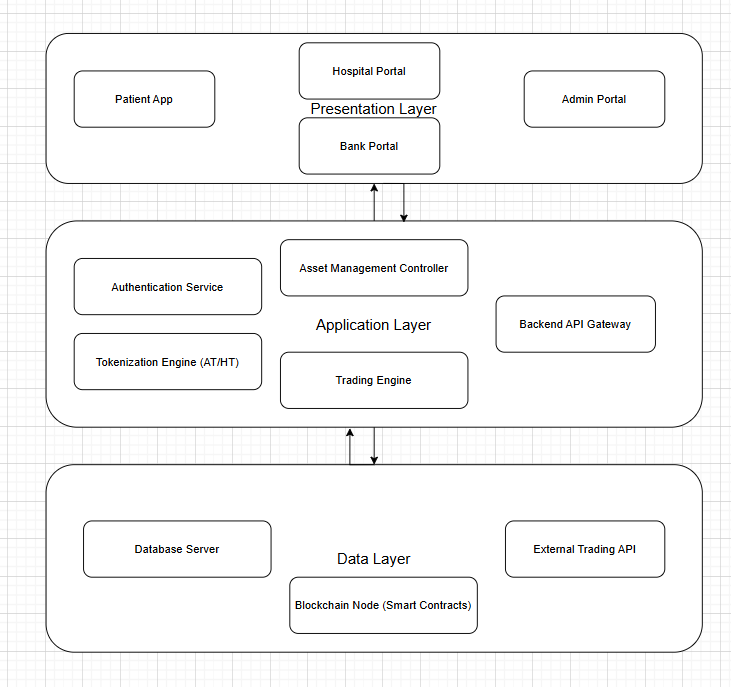
***Subsystems***

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

***Diagram***

******

## Software Architecture

**

# Design Strategy

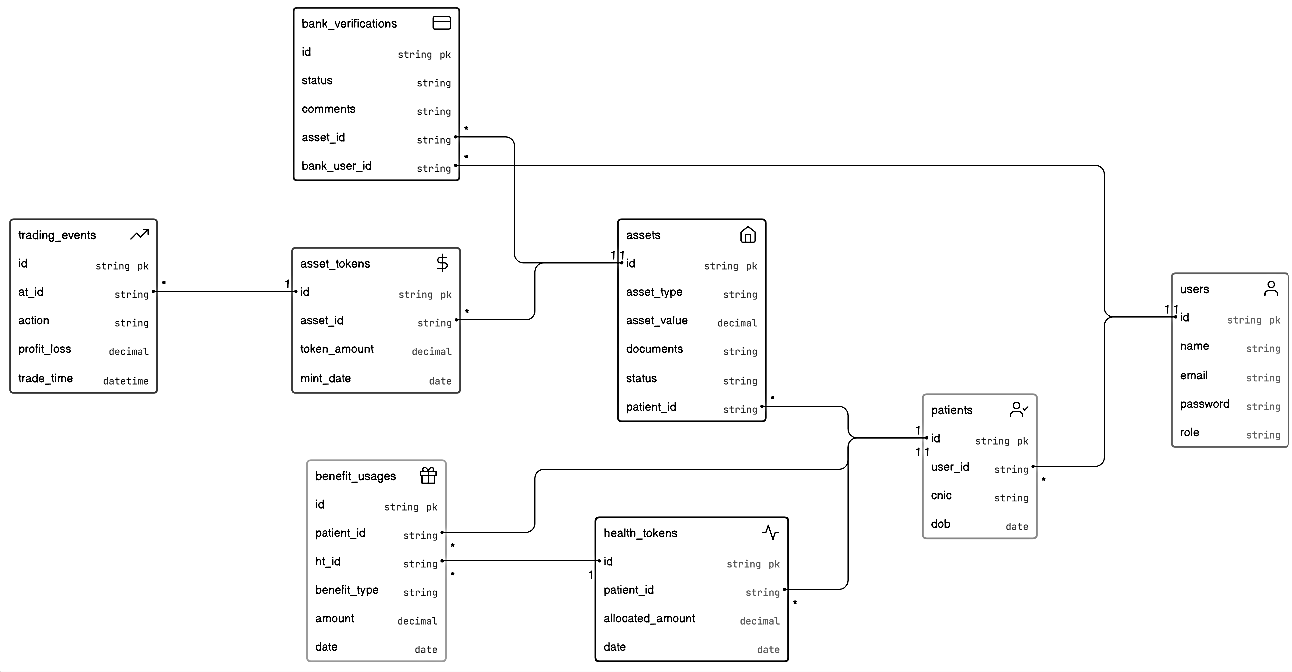
*The following design strategies were used to ensure a more robust and effective software structure:*

* *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 — BankVerification* | *Every asset must be verified* | *1–1* |
| *Asset — AssetToken* | *One asset produces one AT record* | *1–1* |
| *AssetToken — TradingEvent* | *One AssetToken can have many trades* | *1–M* |
| *Patient — HealthToken* | *Patients receive multiple HT allocations* | *1–M* |
| *HealthToken — BenefitUsage* | *HT is consumed in many benefit events* | *1–M* |

### Data Dictionary

#### Data 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **User** | | | | | | | |
| **Name** | | The system-wide account used by all participants (patient, hospital, bank, admin) | | | | | |
| **Alias** | | Account / System User | | | | | |
| **Where-used/how-used** | | All users authenticate using this table. Patients deposit assets, hospitals approve and mint tokens, banks verify deposits, and admins manage roles using this identity record. | | | | | |
| **Content description** | | Stores all basic login and identity information for every system user. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Nullable** | **Default Value** | **Key Type** |
| user\_id | unique system identifier | | Int | 11 | NO | None | PK |
| name | full name of user | | Varchar | 256 | NO | None |  |
| email | unique login email | | Varchar | 256 | NO | None |  |
| password | encrypted password | | Varchar | 256 | NO | None |  |
| role | user role (patient/hospital/bank/admin) | | Varchar | 20 | NO | None |  |

#### Data 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Patient** | | | | | | | |
| **Name** | | The individual who deposits assets and receives health tokens. | | | | | |
| **Alias** | | Patient / Client | | | | | |
| **Where-used/how-used** | | Patients register/login, deposit assets, track AT/HT balances, redeem healthcare benefits, and manage subscriptions. | | | | | |
| **Content description** | | Contains identity and demographic details of the system’s patient users. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| patient\_id | unique patient identifier | | Int | 11 | NO | None | PK |
| user\_id | reference to user record | | Int | 11 | NO | None | FK |
| cnic | national identity number | | Varchar | 15 | NO | None |  |
| dob | date of birth | | Date | 8 | NO | None |  |

.

.

#### Data 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ASSET** | | | | | | | |
| **Name** | | A fixed asset deposited by a patient. | | | | | |
| **Alias** | | Deposited Asset | | | | | |
| **Where-used/how-used** | | Patients submit asset deposit requests, hospitals review, bank verifies and tokens are minted. | | | | | |
| **Content description** | | Stores metadata and verification lifecycle for deposited assets. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| asset\_id | unique asset identifier | | Int | 11 | NO | None | PK |
| patient\_id | reference to patient | | Int | 11 | NO | None | FK |
| asset\_type | gold/land/property | | Varchar | 50 | NO | None |  |
| asset\_value | declared value | | Decimal | 10,2 | NO | None |  |

#### Data 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **BANKVERIFICATION** | | | | | | | |
| **Name** | | Record of bank’s manual verification of an asset. | | | | | |
| **Alias** | | Verification Log | | | | | |
| **Where-used/how-used** | | After hospital approves an asset, bank officers manually verify and update status here. | | | | | |
| **Content description** | | Captures verification decision and reviewer information. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| verification\_id | unique verification identifier | | Int | 11 | NO | None | PK |
| asset\_id | linked asset | | Int | 11 | NO | None | FK |
| bank\_user\_id | verifying bank officer | | Int | 11 | NO | None | FK |
| status | verified/rejected | | Varchar | 20 | NO | None |  |
| comments | verification notes | | Text | — | YES | NULL |  |

#### Data 5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ASSETTOKEN (AT)** | | | | | | | |
| **Name** | | Tokenized representation of a verified asset. | | | | | |
| **Alias** | | AT / Asset Token | | | | | |
| **Where-used/how-used** | | Minted after verification; used in trading simulations to generate profit. | | | | | |
| **Content description** | | Digital token record linked to a physical asset. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| at\_id | unique AT record | | Int | 11 | NO | None | PK |
| asset\_id | reference to asset | | Int | 11 | NO | None | FK |
| token\_amount | minted token amount | | Decimal | 10,2 | NO | None |  |
| mint\_date | mint timestamp | | Datetime | — | NO | None |  |

#### Data 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TRADINGEVENT** | | | | | | | |
| **Name** | | Simulated trading events executed by hospitals. | | | | | |
| **Alias** | | Trade Record | | | | | |
| **Where-used/how-used** | | Hospitals perform API-based trading on AT and profit/loss decides HT allocations. | | | | | |
| **Content description** | | History of all trading actions and their outcomes. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Nullable** | **Default Value** | **Key Type** |
| trade\_id | unique trading record | | Int | 11 | NO | None | PK |
| at\_id | linked asset token | | Int | 11 | NO | None | FK |
| action | buy/sell | | Varchar | 10 | NO | None |  |
| profit\_loss | trade gain/loss | | Decimal | 10,2 | NO | None |  |
| trade\_time | timestamp | | Datetime | — | NO | None |  |

#### Data 7

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HEALTHTOKEN (HT)** | | | | | | | |
| **Name** | | Healthcare benefit tokens allocated to patients. | | | | | |
| **Alias** | | HT / Health Token | | | | | |
| **Where-used/how-used** | | Patients receive monthly/annual HT based on trading profits; redeem them for OPD/Medicine/Discounts. | | | | | |
| **Content description** | | Stores allocation info for all distributed health tokens. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| ht\_id | unique HT identifier | | Int | 11 | NO | None | PK |
| patient\_id | receiver patient | | Int | 11 | NO | None | FK |
| allocated\_amount | distributed HT amount | | Decimal | 10,2 | NO | None |  |
| date | allocation timestamp | | Datetime | — | NO | None |  |

#### Data 8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **BENEFITUSAGE** | | | | | | | |
| **Name** | | Redemption of health tokens by patients. | | | | | |
| **Alias** | | Benefit Transaction | | | | | |
| **Where-used/how-used** | | Patients redeem HT for hospital benefits like OPD, medicines, discounts. | | | | | |
| **Content description** | | Captures every instance where HT tokens were consumed. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| benefit\_id | unique benefit usage record | | Int | 11 | NO | None | PK |
| patient\_id | consuming patient | | Int | 11 | NO | None | FK |
| ht\_id | associated health token | | Int | 11 | NO | None | FK |
| benefit\_type | OPD/Medicine/etc | | Varchar | 50 | NO | None |  |
| amount | HT consumed | | Decimal | 10,2 | NO | None |  |
| date | usage timestamp | | Datetime | — | NO | None |  |

#### 

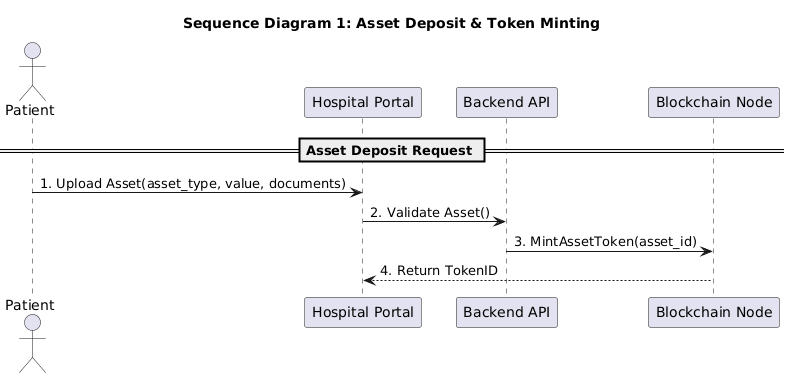
## Application Design

*[A detailed application design should include the following:*

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

### Sequence Diagram

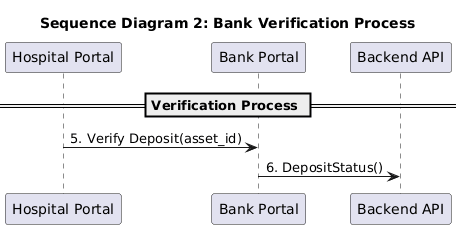
#### Asset Deposit and Token Minting



*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 to the backend to check if everything is correct.*
* *The Backend asks the Blockchain to create a digital token for the asset.*
* *The Blockchain creates the token and sends back a Token ID.*
* *The Hospital Portal receives the Token ID and updates the patient’s record confirming the asset is tokenized.*

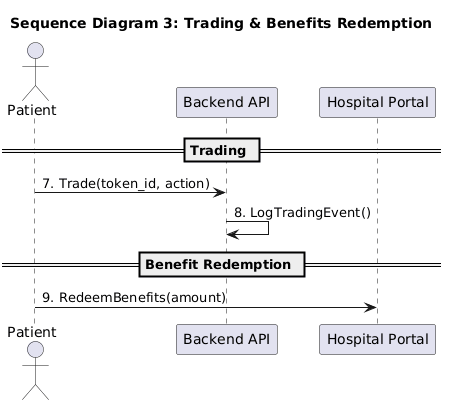
#### Bank Verification Process



*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

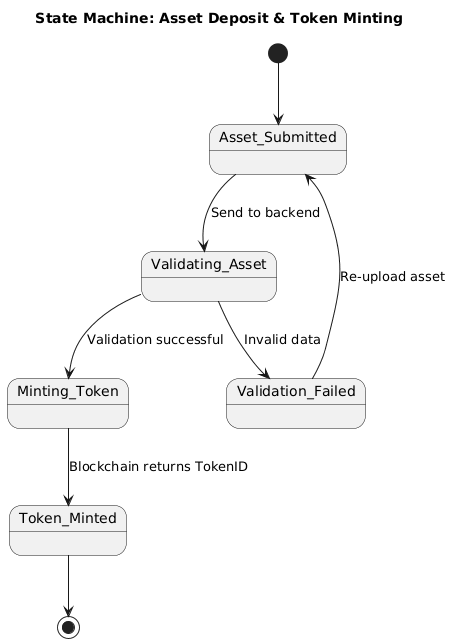


*There are two workflows in this diagram:*

* *Trading Flow*
* *The patient starts a trade (choose buy/sell) using their token.*
* *The Backend processes the trade using the trading module.*
* *The Backend logs the trade result (profit/loss) into the system.*
* *Benefits Redemption Flow*
* *The patient redeems health benefits (OPD, discount, etc.) using their health tokens.*
* *The Hospital Portal checks the patient’s balance and verifies usage.*
* *The system deducts the required HT amount and confirms the benefit redemption.*

### 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*

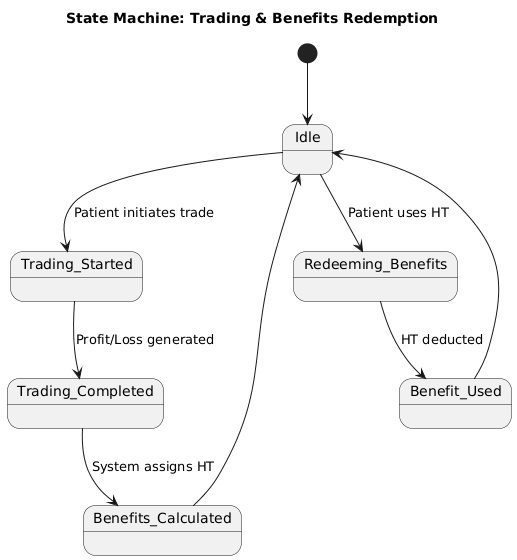
#### Bank Verification

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 shows how the Healthcare Asset Tokenization system handles asset deposits, verification, trading, and benefit redemption. Patients first submit their asset details, which are stored in the Asset Records. The Hospital Portal sends these assets to the backend for validation, and once verified, the system mints digital Asset Tokens and saves them in the token store. The Hospital then forwards the asset to the Bank Portal for manual verification, and the bank’s decision is logged in the verification store. Patients can initiate trades on their tokens, and the trading engine records profit or loss, generates Health Tokens if applicable, and updates token balances. Finally, patients redeem healthcare benefits using their Health Tokens, and the system deducts the used amount and confirms the transaction through the Hospital Portal*

# References

* Ethereum Documentation, Smart Contracts, 2024.
* Polygon Developer Docs, Token Standards, 2024.
* Healthcare Tokenization Models – IEEE Journals.
* Web3.js & Ethers.js Documentation.
* Financial Custodian Models – Banking Standards Publication.

# Appendices

*[Include supporting detail that would be too distracting to include in the main body of the document.]*