Requirements Document

for

BlockPe

Version 1.0

Prepared by

Group: 1

Chinmay Hiran Pillai	**200298**	**chinmay20@iitk.ac.in**
Raj Vardhan Singh	**200760**	**rajvs20@iitk.ac.in**

Course: CS731

Date: 07/04/2024

Index

1. Functional Requirements:

1.1 User Account Creation:

- The system shall implement user account creation functionality for both employers/managers and employees/contractors.
- Employers/managers and employees/contractors shall be required to provide essential details such as company name, tax compliance information, and financial information during account creation.
- User Account Creation: Users (managers or contractors) shall be able to create their accounts by providing necessary details such as username, email, and bank account number. Upon creation, each user shall be assigned a unique identifier.
- User Authentication: The system shall support user authentication mechanisms to allow users to securely log in using their credentials.

1.2 Contract Agreement:

- The system shall allow employers/managers to create contracts specifying the contractor, duration of work, interval between payments, and money offered per interval of work.
- Employees/contractors shall have the ability to review and agree or reject the terms of the contract.
- In order to accept the contract, the contractors shall be required to provide their financial details, including preferred currency and banking information.
- Upon the contractor accepting the contract, the final contract shall again be sent back to the manager for approval.
- The manager should then be able to accept or reject the final contract.
- Contract Creation: Managers shall be able to initiate the creation of contracts by specifying details such as contractor, duration, payment terms, and nature of work. Contractors shall receive contract proposals and have the option to accept or reject them.
- Contract Finalisation: Once both the manager and contractor agree to the contract terms, the contract shall be finalised and added to both parties' profiles.

1.3 Payroll System:

- Payment calculations shall be performed by the system based on the terms outlined in the contract agreements.
- Payment disbursement shall be limited to once per interval for employees/contractors.
- Flexibility shall be provided for employees/contractors to withdraw payments on an as-needed basis.
- The system shall incorporate taxation functionality to calculate and deduct taxes from payments at source.
- Payment Transactions: The system shall facilitate secure payment transactions between users, including domestic and cross-border transfers. Payments shall be processed efficiently and accurately, with transaction details recorded for auditability.

1.4 Local Settlement:

- The system shall facilitate seamless transfer of funds between parties within the same country or currency zone.
- Transaction fees, if applicable, shall be implemented for inter-bank transfers.
- In cases where both parties are located within the same country, funds transfer shall be executed directly without currency conversion or cross-border transaction fees.

1.5 Cross-Border Settlement:

- The system shall support international transactions with currency conversion capabilities.
- Central banks and forex banks shall be utilised for accurate conversion support and low conversion fees.
- Secure and transparent transactions shall be recorded on the blockchain for non-repudiation.
- Different types of banks, including sponsor banks (central banks), member banks and forex banks shall be involved in the cross-border settlement process.
- Central banks shall handle cross-border payments, charging low international-transfer fees.
- Forex banks shall provide accurate currency conversion support and charge low conversion fees.

1.6 Off-Chain Database Storage:

- Non-critical user details, such as email and address, must be stored off-chain in a separate database.
- This Database must be accessed through a separate server.
- Servers of backend and Hyperledger should be different.
- Non-critical user details, such as email and address, shall be stored in a separate SQLite3 database for efficient retrieval. The database shall be accessed through a separate server to minimise latency and optimise performance.

2. Non-Functional Requirements:

2.1. Performance:

- The system shall efficiently handle a high volume of contract agreements, payroll calculations, and payment settlements.
- Database queries and transaction processing shall have high throughput.
- The system should optimise resource utilisation to ensure efficient use of computing resources and minimise costs.

2.2. Scalability:

- The system shall be designed to scale horizontally and vertically to accommodate a growing user base and transaction volumes.
- Scalability should be built into the system architecture to accommodate future growth in user base and transaction volume.

2.3. Usability Requirements:

- The user interface should be intuitive and user-friendly, requiring minimal training for users to perform tasks.
- Clear and concise documentation should be provided to guide users through system functionalities and processes.

2.4. Space Requirements:

- Sufficient storage capacity shall be ensured to accommodate increasing data volumes over time.
- Efficient data storage mechanisms should be implemented to optimise disk space utilisation and minimise storage costs.

2.5. Dependability Requirements:

- High availability should be maintained, with failover mechanisms to minimise downtime.
- System availability should exceed 99.99%, ensuring uninterrupted service for users.

2.6. Security Requirements:

- Data encryption should be implemented to protect sensitive information stored on the blockchain and in the database.
- A scalable hyperledger system shall be implemented to securely store contract details, financial information, and transaction records.
- Access controls should be enforced to prevent unauthorised access to system resources and data.
- The system shall ensure efficient and secure management of assets, users, bank accounts, and contracts.

2.7. Operational Requirements:

- The system must be built to run on the Linux operating system and all its different distributions.

2.8. Development Requirements:

- The development process should follow best practices for software engineering, including version control and code reviews.
- Documentation should be comprehensive and up-to-date, covering system architecture, APIs, and development guidelines.

2.9. Seamless Integration:

- The system shall seamlessly integrate local banking systems (IBIBI, ADFC and YESBI) with central banking systems (USD and INR) within the blockchain network.

- The integration shall ensure interoperability and smooth communication between different banking entities to facilitate efficient financial transactions.

2.10. Transparency and Auditability:

- The smart contract logic shall enforce business rules and logic transparently on the blockchain platform.
- All transactions and contract interactions shall be recorded on the blockchain ledger, providing a transparent and immutable audit trail for regulatory compliance and accountability.