# Software Requirements Specification

## for

## CROSS BORDER PAYMENT SYSTEM

## Prepared by

| | | | |
|---|---|---|---|
| MOHIL | 200596 | CSE | mohil20@iitk.ac.in |
| Aryan Thada | 210205 | CSE | aryant21@iitk.ac.in |

**Course: CS731**

**Mentor TA:** **SUMIT LAHIRI**

**Date:** **07-04-2024**

# 1 Introduction

## 1.1 Product Scope

- ***BORDERPAY.io*** *is a Hyperledger based cross border payment system that allows user to do transactions in secure and efficient way.*
- *The app helps in cross border payments and local payments.*
- *It will reduce the extra cost deduction as tax.*
- *It is also secure as it is implemented on Hyperledger technology so it is a private blockchain.*

## 1.2 Intended Audience and Document Overview

*This document is intended for the users ,developers (the project team), course instructor (prof. Angshuman Karamakar) and to all the TA's.*

# 2 Overall Description

## 2.1 Product Overview

*This app is developed for secure and efficient transactions using the hyperledger technology as it's a private blockchain the data of the organisations is kept private and secure. Also reduce transaction fees deducted by many kind of intermediate banks.*

## 2.2 Product Functionality

- *Allow transaction between employer and employees in safe way.*
- *This app will allow to have a payment contract agreement so that the employer and employee trust each other.*
- *The app will have a payroll system, this is a system that automatically calculates the payment that employee must recieve at regular intervals (as per the contract made). employee can raise an advance request and recieve an advance payment if employer approves it.*
- *This app will provide different settlement type like local settlement and cross border settlement.*

## 2.3 Design and Implementation Constraints

***Language Requirements:*** *The Language used for implementing smart contracts is GO.*
*Backend: In Backend We have Hyperledger fabric in golang.*

*Frontend: ReactJS*
*Testing : javascript.*
***Tools: VSCode, Docker,CURL,GOLang,JQ,HyperLedgerFabric***

# 3 Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

*First time users are required to sign-up and create a new account by providing some basic details and creating a password for the same. Old users can simply sign-in with their pre-existing account. They will be provided a user ID which would be visible to other clients along with other public details.*

*Upon logging in, the user can initiate contract creation, approve for the contract, link account with their bank (if bank is already on the network).*

*Any party can revoke the contract after the first payment is completed.*

*An employee account can query all the details about its contract details like CTC, working hours about a particular employer or all its employers and its bank private details.*

*An employer account can query all the details about its contract details about a particular employee or list of all the employees working for it and its bank private details.*

*A bank can query all the details about its customer linked with the bank.*

### 3.1.2 Hardware Interfaces

*The app requires the following hardware:*

- *PC / Mobile (to access the web-app through a browser)*
- *Internet connection*

### 3.13 Software Interfaces

*The software interface of this app involves Hyperledger Fabric wrapped in APIs. React app utilizes these APIs to make calls, enabling changes in the blockchain state or querying blockchain data.*

## 3.2 Functional Requirements

- *User Registration: New users will be able to register through a valid email Id, they will then be needed to provide some confidential details, and a password for their account.*
- *Login/ Sign In: Once a user has registered successfully, he/ she will be able to login to the app using their email Id and password.*
- *Dashboard: After successful login the users will be redirected to the Dashboard, where they will be able to create contracts and manage existing contracts.*
- *Contract Creation: Employer account can create a contract mentioning the CTC, duration of work, nature of work and frequency of the payment.*
- *Contract Agreement: Employee account can agree the terms of the Employer and the currency will be decided based on the location of the Employee. Employee can reject the contract if any discrepancies are found. If Employee agrees, then deal is signed and approved by Employer and contract will be active.*
- *Advance payment:*
- *Local settlement: Local settlement refers to payments made within the country. There is no currency conversion or crossborder transaction fees involved.*
- *Cross Border Settlement: These are typically international transactions [Border 2023] similar to wire transfer [wire transfer 2024] or swift [Swift 2023].*

# 4 Other Non-functional Requirements

## 4.1 Performance Requirements

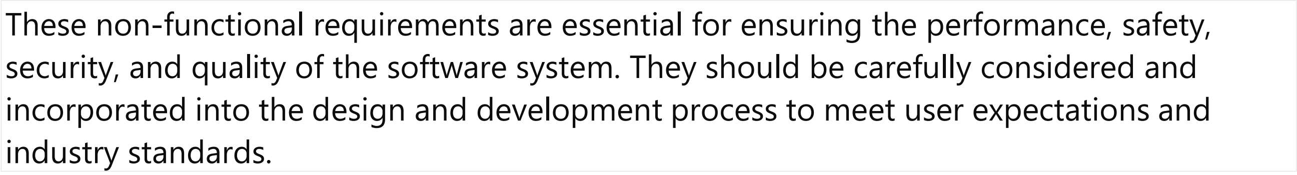
1. **Transaction Throughput**: The system should support a high transaction throughput to accommodate a large number of domestic and cross-border payments efficiently.
2. **Response Time**: The response time for executing transactions and generating reports should be minimal to ensure a smooth user experience.
3. **Scalability**: The software must be scalable to handle increasing transaction volumes and user interactions without compromising performance.
4. **Resource Utilization**: Efficient utilization of computational resources should be ensured to optimize system performance and reduce operational costs.

## i. 4.2 Safety and Security Requirements

1. **Data Encryption**: All sensitive data, including personal and financial information, should be encrypted to prevent unauthorized access.
2. **Access Control**: Role-based access control mechanisms should be implemented to restrict system access based on user roles and permissions.
3. **Audit Trails**: Comprehensive audit trails must be maintained to track user activities and detect any unauthorized or suspicious behavior.
4. **Regular Security Updates**: The system should undergo regular security updates and patches to address potential vulnerabilities and protect against cyber threats.

## ii. 4.3 Software Quality Attributes

1. **Reliability**: The software should consistently perform its intended functions accurately and reliably under various conditions.
2. **Maintainability**: The system should be designed with modular components and well-documented code to facilitate easy maintenance and future enhancements.
3. **Usability**: The user interface should be intuitive and user-friendly, providing seamless navigation and efficient interaction with the application.
4. **Compatibility**: The software should be compatible with different operating systems, web browsers, and devices to ensure broad accessibility for users.
5. **Scalability**: The architecture should support horizontal and vertical scalability to accommodate future growth and increasing user demands.
6. **Resilience**: The system should have built-in mechanisms to recover gracefully from failures and disruptions, ensuring continuous availability and minimal downtime.



These non-functional requirements are essential for ensuring the performance, safety, security, and quality of the software system. They should be carefully considered and incorporated into the design and development process to meet user expectations and industry standards.