RWANDA POLYTECHNIC

IPRC MUSANZE
Integrated Polytechnic Regional College

P.O.Box 226 Musanze-Rwanda
Tel: +250 785 189 494
Email: info@iprcmusanze.rp.ac.rw
www.iprcmusanze.rp.ac.rw

Module Title: BLOCKCHAIN TECHNOLOGY

Module Code: ITLBT801

Date: 23 June 2024

GROUP II MEMBERS

FULL NAME	REGISTRATION NUMBER
KWIZERA CYIZA Christopher	23RP00559
UWIMANA Xaverine	23RP01154
MANIRIHO Eduard	23RP00159
MUKANDAYISENGA Chantal	23RP01132

Project Title: Blockchain-Based Insurance Claim Optimization Application

Initial Requirement Specification (IRS) blockchain-based insurance claim optimization application

1. Introduction

This document outlines the initial requirements for a blockchain-based insurance claim optimization application built on the blockchain. This application aims to provide a secure and transparent way for insurance companies and policyholders to manage and process insurance claims efficiently.

2. System Overview

The blockchain-based insurance claim optimization application is a smart contract-based system deployed on a blockchain platform. It allows users to:

Submit insurance claims

- Provide required documentation and information
- Verify claim details through automated processes
- Trigger automatic claim approval or rejection based on predefined rules
- Ensure transparent and secure claim processing

3. Functional Requirements

User Management:

- **Account Creation:** Users (policyholders and insurance company representatives) can create an account and log in to the application.

- **Profile Management:** Users can view and update their profile information.
- **Claim History:** Users can view their past and current claims.

Claim Submission:

- **Claim Form:** Users can submit a new insurance claim by filling out a standardized digital form.
- **Document Upload:** Users can upload necessary documents such as medical reports, police reports, and photos to support their claim.

Claim Review: Users can review and edit the claim before submitting it.

Claim Processing:

- **Automated Verification:** The system integrates with third-party data sources to automatically verify claim details.
- **Rule-Based Approval:** Smart contracts define rules for automatic claim approval or rejection based on predefined criteria.
- **Fraud Detection:** The system uses machine learning algorithms to detect and flag potentially fraudulent claims.

Claim Management:

- **Edit Claims:** Users can edit submitted claims before processing begins.
- **Status Tracking:** Users can track the status of their claims in real-time.

Payment Processing:

- **Automatic Payouts:** Upon claim approval, the smart contract automatically triggers the payment to the policyholder's account.
- **Transaction History:** The application displays a clear transaction history for each claim.

4. Non-Functional Requirements

Security:

- **Data Encryption:** All user data and claim information must be securely stored on the blockchain.
- **Access Control:** Access to sensitive functions (e.g., claim approval) should be restricted to authorized users.

Code Audit: The smart contract code should be thoroughly audited to prevent vulnerabilities.

Performance:

- **Responsiveness:** The application should be responsive and provide a smooth user experience.
- **Minimal Transaction Fees:** Transaction fees for interacting with the smart contract should be kept minimal.

Scalability:

- **User Accommodation:** The application should be designed to accommodate an increasing number of users and claims.

Usability:

- **User-Friendly Interface:** The application interface should be user-friendly and intuitive.
- **Ease of Use:** Users should be able to easily understand and manage their claims.

5. Components:

Smart Contract:

- **Core Logic:** The core logic resides in a smart contract deployed on the blockchain. It stores claim data and manages claim processing and payouts.

User Interface (UI):

- **Interaction:** A web-based or mobile application provides a user interface for interacting with the smart contract.

Oracle Service:

- **Verification:** An external service verifies claim details (e.g., medical records, police reports) and triggers the smart contract for claim processing.

6. Data Flow:

- User creates an account and logs in to the application.
- User submits a new claim via the UI.
- The UI interacts with the smart contract, storing claim data (claim details, documentation) on the blockchain.
- The system verifies claim details through automated processes and third-party data sources.
- The smart contract processes the claim based on predefined rules.

- Upon approval, the smart contract triggers the payout to the policyholder’s account.
- The UI displays transaction history and relevant information for users.

7. Security Considerations

- **Authentication:** Secure user authentication and authorization mechanisms are needed to prevent unauthorized access.
- **Code Audit:** The smart contract code should be rigorously audited to identify and address potential vulnerabilities.
- **Data Protection:** Secure storage solutions should be employed for private user data (e.g., off-chain storage with encryption).

8. Integration Requirements

- **Blockchain Platform:** The application needs to integrate with a blockchain platform that supports smart contracts.
- **Oracle Service:** Integration with an external oracle service is necessary for claim detail verification.

9. Success Criteria

- The application allows users to submit and manage insurance claims securely on the blockchain.
- The application facilitates secure and transparent claim processing upon approval.
- The application is user-friendly and provides a smooth user experience.

10. References

Solidity Programming Language Documentation (https://docs.soliditylang.org/)

Blockchain Platforms with Smart Contract Support (e.g., Ethereum, Binance Smart Chain)

This initial requirement specification outlines the foundation for developing a blockchain-based insurance claim optimization application. It emphasizes security, transparency, and user experience to ensure effective and efficient claim management.