# Software Requirements Specification

## 1. Introduction

- ### Purpose

  The purpose of this document is to provide a comprehensive overview of the requirements for the development of a decentralized Twitter application using Solidity.

- ### Scope

  The decentralized Twitter application aims to provide users with a censorship-resistant platform for microblogging and social interaction. Users can post, follow, and interact with others in a decentralized and secure environment.

## 2. System Overview

- ### System Description

  The decentralized Twitter application will be built on a blockchain using Solidity smart contracts. Users will have control over their data, and interactions will be recorded on the blockchain for transparency.

- ### System Architecture

  The system architecture will consist of smart contracts deployed on a blockchain, a front-end user interface, and communication protocols for decentralized networking.

## 3. Functional Requirements

- ### User Registration

  Users can create an account by generating a unique cryptographic key pair.

  Accounts will be pseudonymous to maintain user privacy.

- ### Posting Tweets

  Users can compose and post tweets.

  Tweets will be stored on the blockchain for immutability.

- ### Following and Unfollowing

  Users can follow and unfollow other users.

  Follower/following relationships will be recorded on the blockchain.

- ## Interactions

  Users can like and comment on tweets.

  Interactions will be recorded on the blockchain.

- ## User Profile

  Users can customize their profiles with a display name and avatar.

  Profile data will be stored on the blockchain.

- ## Decentralized Governance

  A decentralized governance mechanism will be implemented for decision-making processes, such as protocol upgrades.

# 4. Non-functional Requirements

- ## Performance

  The system should handle a high volume of transactions efficiently.

  Response times for interactions should be minimal.

- ## Security

  User data should be encrypted and stored securely on the blockchain.

  Smart contracts should be audited for vulnerabilities.

- ## Reliability

  The system should be available and operational 24/7.

  Backup and recovery mechanisms should be in place.

- ## Usability

  The user interface should be intuitive and user-friendly.

  Onboarding processes should be straightforward.

- ## Scalability

  The system should be designed to handle an increasing number of users and data.

# 5. Constraints

The application's performance may be affected by the underlying blockchain's limitations.

Users need to have access to a compatible blockchain wallet.

# 6. Future Enhancements

Integration with decentralized identity solutions.

Integration with decentralized file storage for media attachments.