**ENS Contract Events**

**System Requirements Specification**

**Version: 0.1**

**Document Serial Number: 001**

**Prepared by Nasr Eldin Mohamed**

Table of Contents

Table of Contents…………….…………………………………….……….……………………………………………………………………….2

Revision History……………………………………………………………………………………………………………………………………….3

[1. Introduction 4](#_Toc22347535)

[1.1 Purpose 4](#_Toc22347536)

[1.2 Definitions & Acronyms 4](#_Toc22347537)

[1.3 Use Cases and Stories Index 4](#_Toc22347538)

[2. Project Overview 5](#_Toc22347539)

[2.1 Business Objectives 5](#_Toc22347540)

[2.2 Scope of Work 5](#_Toc22347541)

[2.3 Product Perspective 5](#_Toc22347542)

[2.4 Out of Scope 5](#_Toc22347543)

[3. Actor Catalogue 6](#_Toc22347544)

[4. Application Non-Functional Requirements 6](#_Toc22347545)

[5. Application Functional Requirements 7](#_Toc22347546)

[5.1 Use Case Diagram 7](#_Toc22347547)

[5.2 Use Cases 8](#_Toc22347548)

[5.3 Application Messages 9](#_Toc22347549)

**Revision History Log**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Creator | Issue Date | Changes |
| 0.1 | **Nasr Eldin** | 18 October 2019 | Initial draft |
|  |  |  |  |

# Introduction

## Purpose

The Purpose of this document is to present Software requirements specifications for ENS Contract Events Service. This document is intended to be used as a reference during the development, implementation and testing of this Test case.

## Definitions & Acronyms

|  |  |
| --- | --- |
| Term\Acronym | Definition |
| ENS contract events | New test app allows to get list of contract events |
| Ethereum | Ethereum is a global, decentralized platform for money and new kinds of applications. |
| Smart Contract | A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. |
| Ethereum address | Is simply a 64-character hex string generated subject to various rules defined in the Ethereum yellow paper. |
| ENS | Ethereum Name Service (ENS) offers a secure and decentralized way to address resources both on and off the blockchain using simple, human-readable names. |
| MVP | A minimum viable product (MVP) is a development technique in which a new product or website is developed with enough features to satisfy early adopters. |
| DApp | Decentralized application |
|  |  |

## Use Cases and Stories Index

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Module | Use Case | Stories |
| Release 1 | | | |
| Sprint 1 | | | |
|  |  | [*[UC-001] Event List*](#_[UC-001]_Event_List) |  |
|  |  |  |  |

# Project Overview

## Business Objectives

We Need to investigate or list the ENS contract events of the past days. So, we write a MVP test application to get a list of the ENS contract events of the past (e.g. the last 1 or 2 days) focusing on the event for registered names.

## Scope of Work

Display contract events in the table UI and can search, filter, sort events with pagination.

## Product Perspective

|  |
| --- |
| A picture containing screenshot  Description automatically generated |

## Out of Scope

* Any services not mentioned within this document.

# Actor Catalogue

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Actor Name | Brief Description | Characteristics |
| 1 | DApp | Client app connect to eth network throw web3 to call smart contract and subscribe to events | System |
| 2 | Ethereum network | Public, blockchain-based distributed computing platform and operating system featuring smart contract functionality. | System |

# Application Non-Functional Requirements

|  |  |
| --- | --- |
| ID | Dependency |
|  | App should offer bullet-proof security mechanisms for access. |
|  | App should have great user-experience. |
|  | Asynchronous manner should be used for long-time processes. |
|  | Great user-experience   * Ease of access * Simple & short steps to view events * Excellent performance * Totally secured and bullet-proof against security threats |
|  | Performance:   * Load app settings and contract config within initialize app. * Lazy load for requested huge list of data based on demand approach. |
|  | Usability:   * DApp should support by Chrome. * DApp should support auto responsive |

# Application Functional Requirements

## Use Case Diagram

|  |
| --- |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | UCD-01 | **Name** | ENS Contract Events |
| Client app interact with smart contract and get past events from contract. | | | |
| **Remarks**: | | | |

## Use Cases

#### [UC-001] Event List

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID** | UC-001 | | |
| **Use Case Title** | Event List | | |
| **Description** | This use case to explain how DApp interact with eth smart contract to allowed get contract data and events from return values | | |
| **Primary Actor** | Client app | | |
| **Secondary Actors** | Eth network | | |
| **Assumptions** | * Service categorized into contract return values include events | | |
| **Triggering Event** | Load contract events initialize app | | |
| **Pre-Condition(s)** | * DApp Authorized to access smart contract * Provide eth address and provider network | | |
| **Post-Condition(s)** | * View of services details * map return values properties to events model | | |
| **Included Use Cases** | N/A | | |
| **Extended Use Cases** | N/A | | |
| **Related Business Use Case (BRS/BC)** | N/A | | |
| **Related Non-Functional Requirements** | - [Application Non-Functional Requirements](#_Application_Non-Functional_Requirem) | | |
| **Logging Requirements** | N/A | | |
| **Priority** | High | | |
| **Basic Flow** | **#** | **Step** | **Other Flows** |
| 1 | User open DApp and click on events list (DApp load events in initialize app automatic) |  |
| 2 | DApp display list of contract events |  |
|  |  |  |
| **Exception Flows** |  |  |  |
| **Activity Diagram** | **A close up of a map  Description automatically generated** | | |
| **Wireframe** | **A screenshot of a cell phone  Description automatically generated** | | |

## Application Messages

|  |  |  |
| --- | --- | --- |
| MSG ID | Error Type | English |
| MSG001 | Error | Could not load file ‘json file name’ |
| MSG002 | Error | error occurred |
| MSG003 | Success | Detect MetaMask has successfully connected. |
| MSG004 | Alert | No MetaMask detected. Install MetaMask in your browser https://metamask.io |