CANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Software Requirements Specification

AGRICROWD

Advisor: Assoc. Prof. Dr. Gül TOKDEMİR

Enes Ramazan AKDAMAR 201911002

Mustafa Eren BURUK 201911017

Emirhan GÖKOĞLU 201911027

Adil Ayberk TÜKENMEZ 201911063

Mustafa YILMAZ 201911075

Contents

1. INTRODUCTION 3
   1.1. Purpose 3
   1.2. Scope of Project 3
   1.3. Glossary 4
   1.4. References 4
   1.5. Overview of the Document 5
2. OVERALL DESCRIPTION 5
   2.1. Product Perspective 5
   2.1.1. Development Methodology 5
   2.2. User Characteristic 6
      2.2.1. Investee 6
      2.2.2. Investor 6
3. REQUIREMENTS SPECIFICATION 6
   3.1. External Interface Requirements 6
      3.1.1. User Interfaces 6
      3.1.2. Hardware Interfaces 7
      3.1.3. Software Interfaces 7
      3.1.4. Communications Interfaces 7
   3.2. Functional Requirements 8
      3.2.1. Profile Management Use Case 8
      3.2.2. Investment Use Case Diagram 11
   3.3. Performance Requirement 15
   3.4. Non-Functional Requirements 15
      3.4.1. Portability 15
      3.4.2. Performance 15
      3.4.3. Usability 15
      3.4.4. Adaptability 16
      3.4.5. Scability 16
   3.5. Safety Requirement 16

1. INTRODUCTION

1.1. Purpose

This Software Requirements Specification (SRS) document outlines the requirements and specifications for an innovative agricultural crowdfunding platform. The platform aims to revolutionize traditional funding models for agricultural projects by leveraging cutting-edge technologies such as blockchain, smart contracts, and modern web frameworks. This document serves as a blueprint to define the functionalities, constraints, and technical specifications of the proposed platform.

1.2. Scope of Project

The agricultural crowdfunding platform aims to revolutionize financial support for agricultural initiatives, acting as a catalyst for the advancement of farming practices and investment prospects. It is conceived to cater to a wide array of stakeholders, encompassing individual farmers, agricultural enterprises, and potential investors, establishing a synergistic ecosystem for sustainable agricultural progress. The core objectives of the platform encompass the empowerment of agricultural initiatives, the facilitation of investment opportunities, and the assurance of transparency among all participants. By providing a dynamic online hub, the platform enables farmers and agricultural project initiators to showcase their ventures, solicit financial backing, and establish collaborations within a global network of potential investors. Key functionalities include the creation of comprehensive user profiles, detailed project listings, secure and transparent payment processing, implementation of smart contracts, blockchain-based transaction records, and audit trails ensuring transparency and traceability. The platform targets a diverse audience, including farmers, agricultural entrepreneurs, investors, and donors keen on supporting innovative agricultural endeavors. On the technical side, the development involves utilizing React.js, Node.js, and Express.js for frontend and backend functionalities, employing MongoDB for robust database management, and enforcing stringent security measures to safeguard user data and uphold platform integrity.

1.3. Glossary

Term	Definition
Blockchain	Blockchain" is a distributed database or ledger used in many industries. This technology can be used for a variety of purposes, including cryptocurrencies, decentralized finance (DeFi) applications, non-unique tokens (NFTs), and smart contracts.[1]
Smart Contracts	A "Smart Contract" is a self-executing program that automates the necessary processes in an agreement or contract.[2]
Cryptocurrency	A digital or virtual currency secured by cryptography, allowing secure and anonymous transactions, commonly built on blockchain technology.
Decentralization	The distribution of control and authority across multiple nodes or individuals rather than being concentrated in a single central entity.
Transparency	Refers to the openness and accessibility of information within a system, ensuring that data and processes are visible, accountable, and easily understood by stakeholders.
Crowdfunding	Crowdfunding is the process of raising funds for a project or venture by obtaining small amounts of money from a large number of people, often facilitated through online platforms.
Transaction	A "transaction" in blockchain refers to the transfer or exchange of digital assets, utilizing a transparent, secure, and decentralized record-keeping system.
Entrepreneur	Entrepreneur is an individual with the ability to establish, manage their own business, and often implement innovative projects with new ideas or ventures.

1.4. References

[1]. https://decrypt.co/resources/blockchain-basics-what-is-blockchain

[2]. https://www.investopedia.com/terms/s/smart-contracts.asp

1.5. Overview of the Document

The SRS document encompasses various sections, outlining both functional and non-functional requirements, system features, technical architecture, and constraints. It provides a comprehensive understanding of the agricultural crowdfunding platform's specifications, guiding the development process and ensuring alignment with project goals.

2. OVERALL DESCRIPTION

2.1. Product Perspective

The agricultural crowdfunding platform embodies an innovative digital environment specifically designed to revolutionize the financial landscape within the agricultural sector. Functioning as an independent and comprehensive online solution, it serves as a pivotal point of connection between agricultural project initiators, encompassing individual farmers, cooperatives, and larger-scale agricultural enterprises, and potential investors, including individuals, organizations, or institutions interested in supporting these initiatives.

2.1.1. Development Methodology

The platform's development methodology is anchored in the agile framework, an iterative and flexible approach that accommodates evolving project requirements and ensures a responsive development lifecycle. Agile methodology, with its emphasis on adaptability and collaboration, allows for incremental development, continuous feedback, and adjustments throughout the project lifecycle.

The development process initiates with a comprehensive analysis of stakeholder requirements, establishing a foundational understanding of the platform's core functionalities and features. Following this, the development team engages in a series of iterative sprints, each focused on specific aspects of the platform's architecture, design, and functionality.

The agile methodology's hallmark of iterative cycles fosters constant communication and collaboration between development teams, stakeholders, and end-users. Regular sprint reviews and retrospectives enable the team to evaluate progress, identify potential bottlenecks, and adapt swiftly to changing requirements.

Furthermore, embracing an agile mindset encourages the integration of best practices and innovative technologies into the development process. This facilitates the seamless incorporation of emerging trends and technologies, ensuring the platform remains adaptable, scalable, and aligned with industry standards.

Throughout the development lifecycle, agile principles, such as flexibility, adaptability, and customer-centricity, guide the development team in delivering a robust, user-friendly, and feature-rich agricultural crowdfunding platform.

2.2. User Characteristic

2.2.1. Investee

- Investee must upload details of the own project.
- Investee must enter the progress of the project into the system in detail.
- The investee should distribute the profits generated from product sales back through the system, particularly in a reward-based system.

2.2.2. Investor

- Investors should be able to thoroughly examine the details of projects.
- Investors should have the option to either contribute donations or engage in reward-based funding for projects.
- Investors should have the option to either contribute donations or engage in reward-based funding for projects.

3. REQUIREMENTS SPECIFICATION

3.1. External Interface Requirements

3.1.1. User Interfaces

The platform will feature an intuitive and user-centric web interface developed using React.js. It will offer a user-friendly dashboard enabling farmers and project initiators to showcase their ventures and allowing investors to explore projects and participate in crowdfunding activities seamlessly. The interface will prioritize accessibility, ensuring ease of navigation and interaction across various devices and screen sizes.

3.1.2. Hardware Interfaces

The platform's hardware requirements are minimal and will support standard web browser functionalities, allowing users to access the platform across a broad spectrum of devices including desktops, laptops, tablets, and smartphones.

3.1.3. Software Interfaces

To ensure seamless interaction between various components, the platform will integrate Node.js and Express.js for the backend, MongoDB for data storage, and blockchain technologies for secure and transparent transactions.

3.1.4. Communications Interfaces

The platform will utilize HTTPS (Hypertext Transfer Protocol Secure) for secure communication between users' browsers and the platform's servers, ensuring data confidentiality and integrity.

3.2. Functional Requirements

3.2.1. Profile Management Use Case

Use Case:

- Register
- Login
- Modify User Information
- Logout

Diagram:

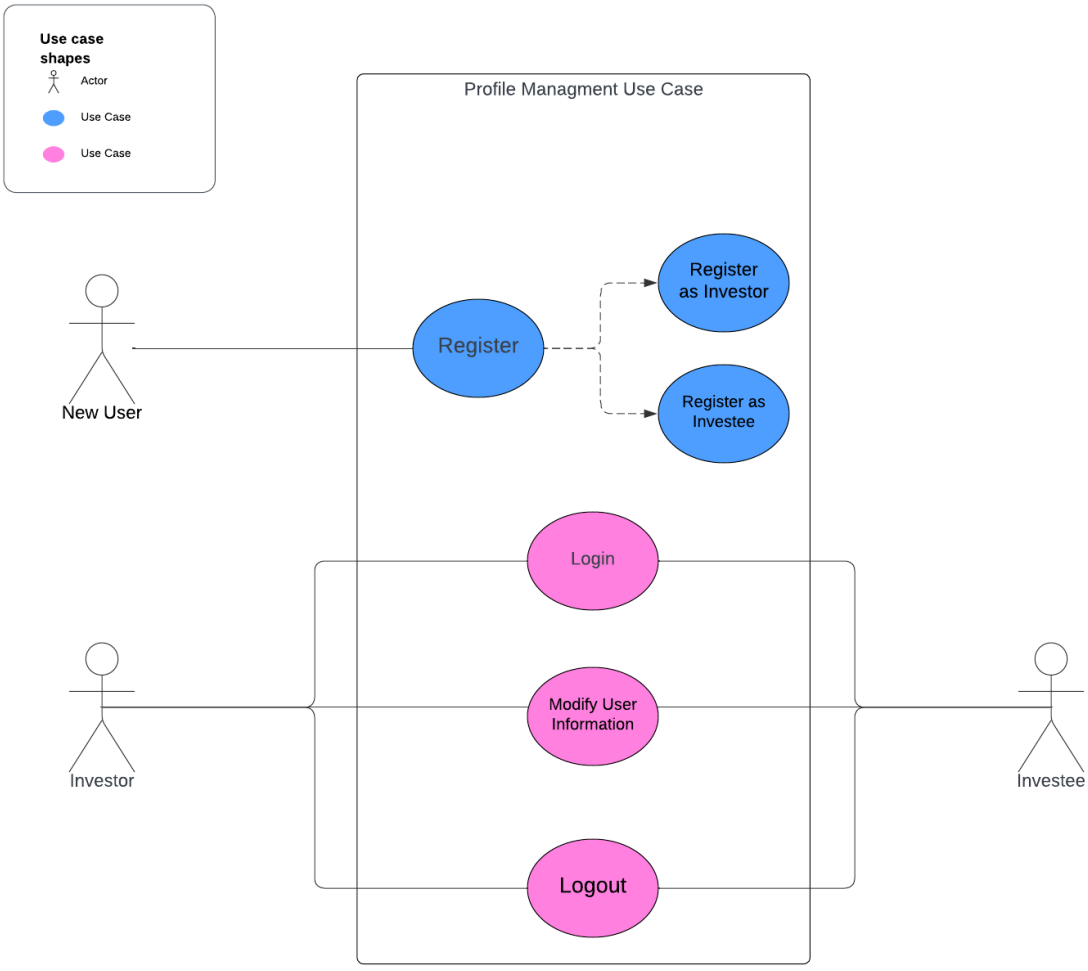

Figure 1: Profile Management Use Case

Brief Description: This use case involves users on the system registering with different roles (Investor, Investee), logging in, managing user information, and terminating their sessions.

Initial Step by Step Description:

- *New User Registration:*

Pre-Conditions:

- The user must intend to register in the system.
- The user should provide necessary information (name, email, password, etc.) during registration.

Scenario:

New users provide the necessary information to register in the system and choose their preferences among role options. These preferences determine whether they will register as an investor or a user seeking investment (Investee). After entering their personal information, users can register in the system.

Post-Conditions:

- The user must register by selecting one of the role options (Investor or Investee).
- The system successfully completes the registration process, allowing the user to log in to the system.

- *Login Process*:

Pre-Conditions:

- The user must have previously registered.
- The user must enter the correct username and password.

Scenario:

Registered users can log into the system with the accounts they created earlier. They log in by authenticating their identity with a username and password. This enables them to access their accounts and be authorized by the system.

Post-Conditions:

- The user safely logs out of the system.
- The system terminates the user's session and restricts the user's access.

- *Modify User Information:*

Pre-Conditions

- The user must have logged into the system.
- The user must have permission to edit their profile information.

Scenario

After logging in, users have the authority to edit their profile information. They can use the "Modify User Information" option to update their user details, correct any incomplete or inaccurate information. This step allows users to keep their personal information up to date.

Post-Conditions

- The user can successfully update their profile information.
- The system records the user's changes and identifies the user with updated information.

- *Logout Process*:

Pre-Conditions

- The user must want to log out while being logged into the system.

Scenario

The logout process allows users to sign out of the system. Users securely log out, terminating their session and exiting the services provided by the system.

Post-Conditions

- The user safely logs out of the system.
- The system terminates the user's session and restricts the user's access.

3.2.2. Investment Use Case Diagram

Use Case:

- Add a project
- Invest a project
- Receive Investment
- Give Reward
- Receive Reward

Diagram:

Figure 2: Investment Use Case Diagram

Brief Description: This use case involves transactions between Investee and Investor users. The first scenario covers listing projects and making investments. The second scenario includes the Investee seeking investment and the Investor making the investment by either purchasing company shares or donating to the project. The final scenario encompasses transactions such as acquiring products or company shares in return for the investments made.

Initial Step by Step Descriptions of Scenarios:

- Scenario 1: Add Project List, View Project List, Invest a Project
    - *Add Project List*

Pre-Condition

- Investee must have logged into the system.

Scenario

Investee adds new projects to the system.

Post-Condition

- A new project is successfully added to the system and listed.
    - *View Project List*

Pre-Condition

- Investor or Investee must have logged into the system.

Scenario

Investor views the list of existing projects.

Post-Condition

- The list of existing projects is displayed to the user.
    - *Invest a Project*

Pre-Condition

- Investor must have viewed the projects and made a selection.

Scenario:

Investor chooses to invest in a specific project and comp

Post-Condition

- Investment is made into a specific project.

- Scenario 2: Receive Investment, Buy Company Project Shares, Donation, Invest a Project

    - *Receive Investment*

Pre-Condition

- Investee must have added the projects to the system and they should have been selected by investors.

Scenario

Investee receives investment for their projects.

Post-Condition

- Investment is made into the project.

    - *Buy Company Project Shares*

Pre-Condition

- Investor must have identified suitable projects to purchase shares of the company, enabling them to receive a proportional share of profits corresponding to the invested amount.

Scenario

This scenario emphasizes that the "Buy Company Project Shares" step requires the investor to select appropriate projects for purchasing shares, allowing them to gain a proportionate share of profits based on their investment amount.

Post-Condition

- Shares of the company's projects are successfully acquired, entitling the investor to a proportional share of profits based on the invested amount.

- *Invest a Project*

Pre-Condition

- Investor must have decided to invest in a new project.

Scenario

Investor invests in a new project.

Donation: Investor makes the investment through a donation.

Invest a Project: Investor makes the investment by purchasing company shares.

Post-Condition

- Investment is made into the new project.

- Scenario 3: Give Reward, Reward(Product, Company Share, Receive Reward)

  - *Give Reward*

Pre-Condition

- Investee or Investor must have selected the project to reward participants.

Scenario

Investee provides the Investor with a reward in return for their investment.

Post-Condition

- Rewards are given to the project participants.

  - *Receive Reward*

Pre-Condition: Investee or Investor must have determined the project from which they will receive the reward.

Scenario

- Product: Investor receives the return in the form of a product.
- Company Share: Investor receives the return in the form of company shares.

Post-Condition: Users receive the reward.

3.3. Performance Requirement

The platform aims for optimal performance, with page load times optimized for efficiency. It should support a substantial number of simultaneous users without compromising performance, ensuring a responsive user experience even during peak traffic.

3.4. Non-Functional Requirements

3.4.1. Portability

The platform's design will prioritize cross-platform compatibility, ensuring seamless operation across various operating systems including Windows, macOS, and Linux, as well as compatibility with major web browsers such as Chrome, Firefox, Safari, and Edge. This approach ensures that users can access the platform from a wide range of devices and environments without compromising functionality or usability.

3.4.2. Performance

Performance optimization will be a key focus throughout the development process. The platform will undergo rigorous performance testing to ensure swift response times, minimal latency, and high throughput. The system will be designed to handle concurrent user interactions efficiently, maintaining stable performance levels even during periods of heavy traffic or increased user activity.

3.4.3. Usability

Usability is central to the platform's design ethos. The user interface will be intuitive, employing clear navigation, logical layout, and informative feedback mechanisms. Accessibility standards will be adhered to, ensuring compliance with guidelines for users with disabilities. Additionally, user feedback will be gathered and analyzed to continuously enhance the platform's usability.

3.4.4. Adaptability

The platform will be architected with a modular and flexible structure, allowing for easy integration of new features, upgrades, and enhancements. APIs (Application Programming Interfaces) will be developed to enable seamless interaction with external systems, facilitating future integrations and extensions. The platform's architecture will be designed to accommodate technological advancements and evolving industry standards.

3.4.5. Scability

Scalability is a core consideration in the platform's architecture. The system will be designed to scale horizontally and vertically, enabling it to handle an increasing volume of users, projects, and transactions. Load balancing mechanisms and cloud-based infrastructure will be leveraged to ensure optimal performance as the user base grows, without compromising on reliability or responsiveness.

3.5. Safety Requirement

The platform will adhere to stringent security measures, employing encryption standards, secure authentication protocols, and regular security audits to protect user data, financial transactions, and overall platform integrity.