Lovely Professional University

Software Requirements Specification (SRS)
Report on Online Farming Equipment and Material

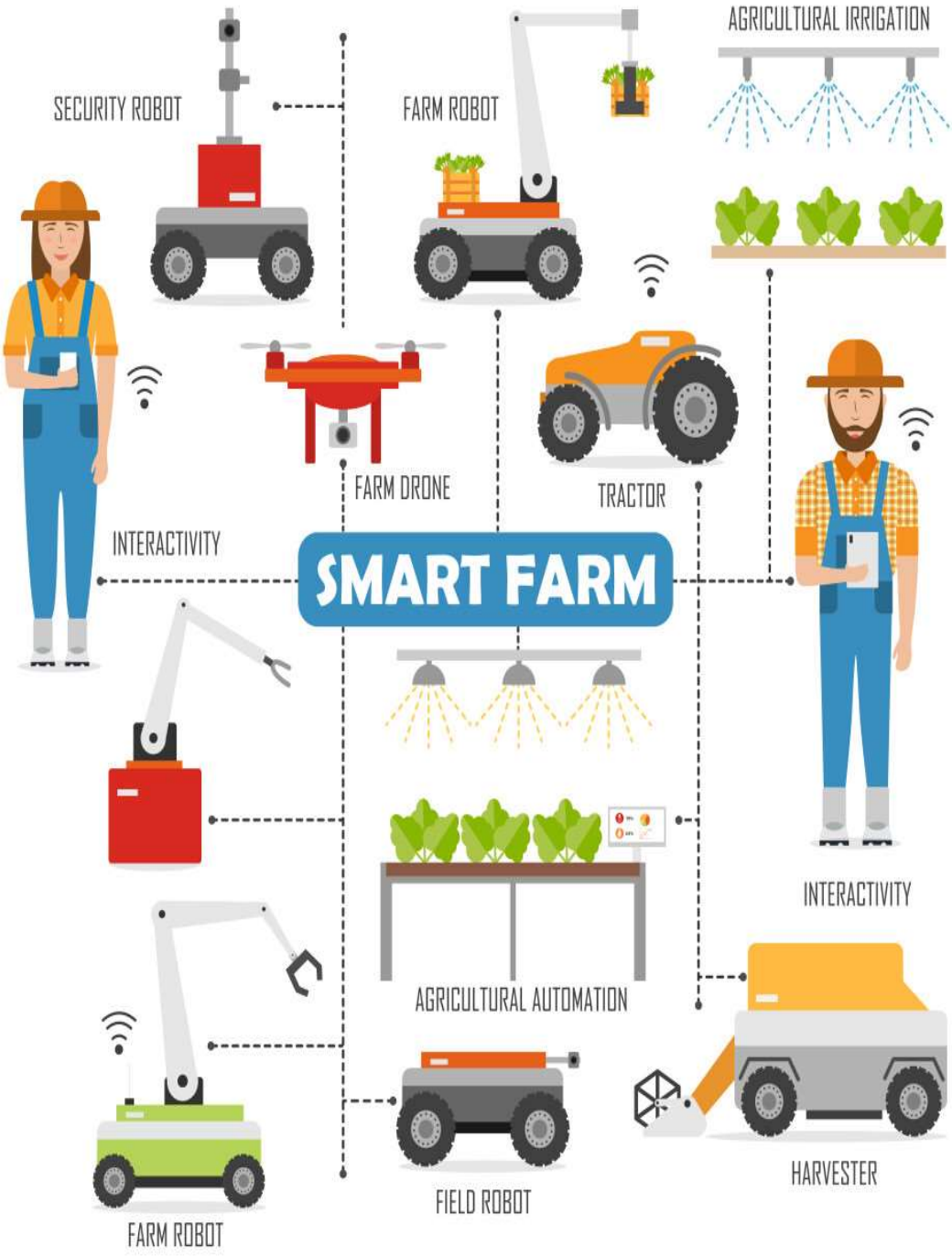
Submitted to:
Assistant Prof. Dr. Vishal Kumar

Submitted by:
Niketa, Aayushi Rajput,
SukhmanPreet *Singh*, Vikas

Index

1. **Introduction**
    - Purpose
    - Scope
    - Definitions, Acronyms, and Abbreviations
    - References
    - Overview
2. **Overall Description**
    - Product Perspective
    - Product Functions
    - User Classes and Characteristics
    - Operating Environment
    - Design and Implementation Constraints
    - Assumptions and Dependencies
3. **Specific Requirements**
    - Functional Requirements
        - User Registration and Authentication
        - Product Catalog Browsing

        - Shopping Cart Management
        - Payment Processing
        - Order Management
        - Customer Support
    - Non-Functional Requirements
        - Performance
        - Security
        - Usability
        - Reliability and Availability
        - Scalability
4. **Use Cases**
    - User Purchase of Farming Equipment
5. **Data Flow Diagrams (DFD)**
    - Level 0 DFD (Context Diagram)
    - Level 1 DFD
6. **Appendices**
    - Appendix A: Glossary of Terms
    - Appendix B: Use Case Scenarios
    - Appendix C: Data Models



Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the Online Farming Equipment and Material platform. This document serves as a comprehensive guide for stakeholders, including developers, project managers, and users, ensuring that all parties have a clear understanding of the system's objectives and functionalities. By detailing the necessary requirements, this SRS aims to minimize misunderstandings and misalignments during the development process, ultimately leading to a product that meets user needs and expectations.

1.2 Scope

The Online Farming Equipment and Material platform is designed to create a user-friendly online marketplace where farmers and agricultural professionals can efficiently access a diverse range of farming equipment and materials. The platform will facilitate various features, including:

- **User Registration**: Enabling users to create accounts and manage their profiles.
- **Product CatLog**: Providing an organized and searchable inventory of farming equipment and materials.

- **Shopping Cart**: Allowing users to add items for purchase and manage their selections.
- **Payment Processing**: Supporting secure online transactions through multiple payment options.
- **Order Management**: Enabling users to track their orders and view purchase history.
- **Customer Support**: Offering assistance through FAQs, chat support, and contact forms.

The target audience for this platform includes individual farmers, agricultural businesses, and suppliers, all aimed at improving the purchasing process and access to necessary farming resources. The project will encompass both web and mobile applications to ensure accessibility across various devices, thus enhancing user engagement and satisfaction.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS**: Software Requirements Specification – A document that describes the intended Behaviour of the software.
- **UI**: User Interface – The point of interaction between the user and the software.
- **UX**: User Experience – The overall experience a user has while interacting with the application.

- **API**: Application Programming Interface – A set of protocols for building and integrating software applications.
- **SSL**: Secure Sockets Layer – A standard security technology for establishing an encrypted link between a server and a client.
- **E-commerce**: Electronic commerce; the buying and selling of goods or services over the internet.
- **Backend**: The server-side of the application responsible for data processing, storage, and business logic.
- **Frontend**: The client-side of the application that users interact with directly, encompassing design and user experience.

1.4 References

Link to IEEE Std 830-1998

Link to IEEE Std 830-1998

Link to OWASP Top Ten

1.5 Overview

This SRS document is structured to first provide an overall description of the product, including its purpose, scope, and user characteristics. Following this, detailed specifications of both functional and non-functional requirements will be presented, offering clarity on what the system must achieve. Additionally, use cases will illustrate how users will interact with the platform, while system architecture will provide a visual representation of the components involved. This structured approach aims to facilitate effective communication among stakeholders and ensure a successful project outcome.

Overall Description

2.1 Product Perspective

The Online Farming Equipment and Material platform will function as an integrated e-commerce solution specifically designed for the agricultural sector. Similar to platforms like **Amazon** or **Alibaba**, which cater to a broad range of products, this platform will focus exclusively on farming equipment and materials. It will connect farmers, suppliers, and agricultural businesses, enabling efficient procurement. By integrating features such as inventory management and order tracking, it aims to streamline the purchasing process, just as **Farm & Fleet** does for rural supplies.

2.2 Product Functions

The primary functions of the platform will include:

- **User Registration and Authentication**: Users can create accounts securely, similar to how **Etsy** allows artisans to register and manage their shops.
- **Product catalogue Browsing**: Users can search and filter products based on categories such as equipment type, price, or brand, akin to the functionality found on **Home Depot**'s website for construction supplies.
- **Shopping Cart**: Users can add products, modify quantities, and review their selections before checkout,

similar to online grocery shopping on platforms like **Instacart**.

- **Payment Processing**: The platform will support secure payment options, including credit cards and digital wallets, mirroring the payment processes used by **Shopify**.
- **Order Management**: Users can track orders and view history, akin to the systems employed by **Zappos** for order tracking and returns.
- **Customer Support**: Features like FAQs and live chat will provide assistance, similar to the customer service options available on **Zillow**.

2.3 User Classes and Characteristics

The platform will cater to:

- **Farmers**: Individuals or businesses seeking equipment, similar to how small farmers use platforms like **Tractor House** to find used machinery.
- **Suppliers**: Businesses listing their products, akin to sellers on **Amazon** who need inventory management tools and sales analytics.
- **Administrators**: Users managing platform operations, similar to the backend management roles seen in companies like **eBay**.
- **Support Staff**: Customer service representatives assisting users, much like the support teams for **Apple** that help customers with product inquiries.

2.4 Operating Environment

The platform will operate in a web-based environment accessible through modern web browsers on desktops, laptops, and mobile devices. For example, similar to **Walmart's** website and mobile app, it will be designed responsively to ensure a seamless user experience across devices. The application will be hosted on a cloud service provider like **AWS** to ensure scalability and reliability, similar to how many e-commerce platforms manage their hosting.

2.5 Design and Implementation Constraints

- **Regulatory Compliance**: The platform must adhere to e-commerce regulations and agricultural product laws, just as **Etsy** complies with various local regulations for its sellers.
- **Performance Requirements**: The system should support multiple concurrent users without degradation, akin to how **eBay** manages high traffic during sales.
- **Budget Limitations**: Development costs must be controlled, similar to how startups use lean methodologies to develop MVPs (Minimum Viable Products).
- **Technology Stack**: The platform will utilize specific technologies (e.g., React, Node.js), which could limit integrations similar to the constraints faced by developers on platforms like **Shopify**.

2.6 Assumptions and Dependencies

- **User Adoption**: It is assumed there will be enough farmer and supplier interest, similar to how **Farmers only** attracted a niche audience through targeted marketing.
- **Internet Access**: Users are expected to have reliable internet access, as seen in the increasing digital engagement among rural communities, similar to how **Rural Connect** initiatives improve access.
- **Third-Party Services**: Successful integration with payment processors (e.g., **PayPal**) is crucial, similar to how e-commerce sites rely on these services for secure transactions.
- **Data Integrity**: Accurate user-generated data will be assumed, much like how platforms such as **TripAdvisor** rely on user reviews and listings to maintain quality.

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Registration and Authentication

- **Description**: Users must be able to create accounts and log in to access personalized features.
- **Requirements**:
    - Users should provide basic information (name, email, password) for registration.

    - Passwords must meet complexity criteria (e.g., at least 8 characters, including uppercase, lowercase, numbers, and special characters).
    - Users should be able to reset their passwords via email verification.
- **Example**: Similar to how **Amazon** requires users to create accounts to purchase items, the platform will ensure secure account management

.

3.1.2 Product CatLog Browsing

- **Description**: Users can browse an organized CatLog of farming equipment and materials.
- **Requirements**:
    - Users should be able to filter products by categories (e.g., machinery, seeds, fertilizers).
    - A search bar should allow users to search for specific items using keywords.
    - Each product listing must include images, descriptions, prices, and availability status.
- **Example**: This functionality is akin to **Home Depot**, where users can filter tools and materials by brand or price range.

3.1.3 Shopping Cart Management

- **Description**: Users can manage items they intend to purchase.
- **Requirements**:
  - Users should be able to add, remove, and update quantities of items in the shopping cart.
  - The shopping cart must persist items across sessions, requiring user login.
  - Users should see an estimated total price including taxes and shipping fees.
- **Example**: Similar to **Walmart's** online shopping experience, users will have an intuitive interface to manage their selected products.

3.1.4 Payment Processing

- **Description**: The platform must handle secure payment transactions.
- **Requirements**:
  - Support for multiple payment methods (credit/debit cards, PayPal, etc.).
  - Implement SSL encryption to protect sensitive payment information.

    - Users should receive confirmation of successful payments via email.
- **Example**: This mirrors the payment process of **Shopify**, where users can choose various payment methods securely.

3.1.5 Order Management

- **Description**: Users can track and manage their orders after purchase.
- **Requirements**:
    - Users should be able to view their order history and status.
    - Provide options for users to initiate returns or exchanges for purchased items.
    - Email notifications should be sent for order confirmations, shipping updates, and delivery confirmations.
- **Example**: Similar to how **Zappos** allows users to track their orders, this functionality will keep users informed throughout the order lifecycle.

3.1.6 Customer Support

- **Description**: The platform will provide support to users for any issues or inquiries.
- **Requirements**:
  - A FAQ section should address common user questions.
  - Live chat support must be available during business hours.
  - Users should have access to a contact form for email inquiries.
- **Example**: This approach is akin to **Etsy**, where customers can reach out for support and access helpful resources easily.

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- **Description**: The platform must perform efficiently under various load conditions.
- **Requirements**:
  - The system should support a minimum of 1000 concurrent users without significant performance degradation.

    - Page load times should not exceed 3 seconds under normal traffic conditions.
- **Example**: This is similar to **eBay**, which manages large volumes of concurrent users during sales events, ensuring a smooth experience.

3.2.2 Security Requirements

- **Description**: The platform must protect user data and transactions.
- **Requirements**:
    - Implement SSL encryption for all data transfers.
    - Regularly conduct security audits and vulnerability assessments.
    - Utilize two-factor authentication for added account security.
- **Example**: Just as **PayPal** ensures high-security standards for transactions, this platform will prioritize user data protection.

3.2.3 Usability Requirements

- **Description**: The platform must provide an intuitive and accessible user experience.
- **Requirements**:
  - The user interface must be consistent and easy to navigate.
  - Accessibility standards (e.g., WCAG 2.1) should be followed to accommodate users with disabilities.
  - User feedback mechanisms should be implemented to gather insights for continuous improvement.
- **Example**: This is similar to the user experience design employed by **Netflix**, which prioritizes ease of navigation.

3.2.4 Reliability and Availability

- **Description**: The platform should be reliable and available for users at all times.
- **Requirements**:
  - System uptime should be at least 99.9%.
  - Implement automated backup solutions to protect data integrity.
- **Example**: This is comparable to **Amazon Web Services** (AWS), which guarantees high availability and reliability for hosted applications.

3.2.5 Scalability Requirements

- **Description**: The platform should be designed to scale as user demand increases.
- **Requirements**:
  - The architecture must support the addition of new features without major disruptions.
  - Cloud-based hosting should allow for dynamic scaling based on traffic fluctuations.
- **Example**: This mirrors how **Spotify** scales its services to accommodate millions of new users without compromising performance.

Use Case: User Purchase of Farming Equipment

Use Case ID: UC-001

Use Case Name: User Purchase of Farming Equipment

Actors:

- **Primary Actor:** Farmer (User)
- **Secondary Actor:** Payment Gateway System

Description:
This use case outlines how a farmer selects and purchases

farming equipment from the online platform, ensuring a seamless experience.

Pre-conditions:

1. The user has created an account on the platform.
2. The user is logged in to the website.

Post-conditions:

1. The user successfully completes the purchase.
2. The system sends an order confirmation email to the user.

Main Flow:

1. **User Logs In:**
   The user visits the website and logs into their account.
2. **Browse Products:**
   The user explores the product catalog, using filters or the search bar to find specific farming equipment.
3. **View Product Details:**
   The user clicks on a product to see detailed information, including specifications, price, and availability.
4. **Add to Cart:**
   The user selects "Add to Cart" to place the product in their shopping cart.
5. **Review Cart:**
   The user reviews items in their cart, adjusting quantities or removing items as needed.

6. **Proceed to Checkout:**
   The user clicks "Checkout" to start the payment process.
7. **Enter Payment Information:**
   The user inputs payment details (credit card, PayPal, etc.) and confirms the shipping address.
8. **Confirm Order:**
   The user reviews the order summary and clicks "Confirm Order" to finalize the purchase.
9. **Payment Processing:**
   The system processes the payment via the payment gateway.
10. **Receive Confirmation:**
    Upon successful payment, the system shows an order confirmation message and sends an email receipt.

Alternative Flows:

- **AF-1: User Not Logged In:**
  If the user tries to access the cart or checkout without logging in, they are redirected to the login page.
- **AF-2: Payment Failure:**
  If the payment fails, the system notifies the user, prompting them to re-enter their payment information.
- **AF-3: Out of Stock:**
  If the user tries to add an out-of-stock item to the cart, the system displays a notification.

Exception Flows:

- **EF-1: Session Timeout:**
  If the user's session times out, they are prompted to log in again.
- **EF-2: Invalid Payment Information:**
  If the user enters invalid payment details, an error message is displayed, asking for corrections.

Real-Life Example:

- This use case is akin to the purchasing process on sites like **Tractor Supply Co.** or **Amazon**, where users can log in, search for products, manage their carts, and complete transactions securely. These steps ensure that farmers have a straightforward and efficient way to purchase essential farming equipment online.

Level 0 DFD (Context Diagram)

Entities:

1. **Farmer (User)**: Interacts with the system to browse and purchase equipment.
2. **Payment Gateway**: Processes payments made by users.
3. **Admin**: Manages products, orders, and user accounts.

Process:

- **Online Farming Equipment System**

Data Flows:

- **From Farmer to System**: User registration details, product search queries, order information.

- **From System to Farmer**: Order confirmations, product listings, status updates.
- **From System to Payment Gateway**: Payment information for processing.
- **From Payment Gateway to System**: Payment confirmation or denial.
- **From Admin to System**: Product updates, user management actions.
- **From System to Admin**: Reports and notifications.

Level 1 DFD

This level breaks down the main process into more detailed sub-processes.

Processes:

1. **User Registration and Login**
2. **Product Browsing**
3. **Shopping Cart Management**
4. **Checkout Process**
5. **Order Management**
6. **Admin Management**

Data Stores:

- **User Database**: Stores user profiles and authentication details.

- **Product Database**: Contains product information and inventory.
- **Order Database**: Records user orders and payment statuses.

Data Flows:

1. **User Registration and Login**:
    - **From User to Process**: Registration info, login credentials.
    - **From Process to User Database**: New user data for storage.
    - **From User Database to Process**: User authentication status.
2. **Product Browsing**:
    - **From User to Process**: Search queries and filters.
    - **From Process to Product Database**: Request for product data.
    - **From Product Database to Process**: Product listings sent back to the user.
3. **Shopping Cart Management**:
    - **From User to Process**: Add/remove items.
    - **From Process to Shopping Cart Data Store**: Update cart status.
4. **Checkout Process**:

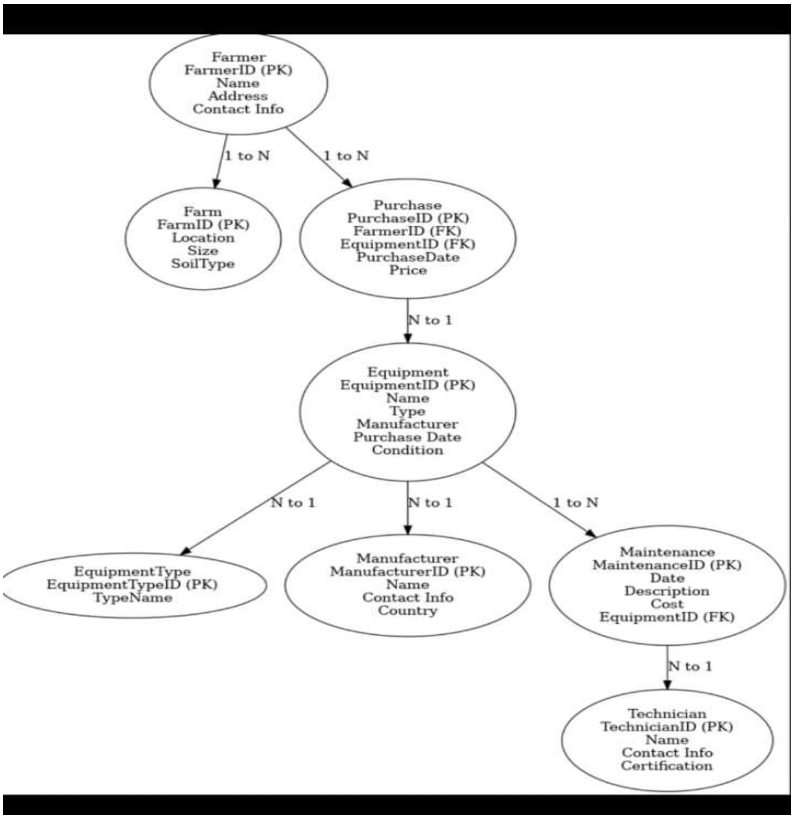
- **From User to Process**: Checkout request with cart details.
- **From Process to Order Database**: Create new order.
- **From Process to Payment Gateway**: Payment information for processing.
- **From Payment Gateway to Process**: Payment confirmation or failure.
- **From Process to User**: Order confirmation message.

5. **Order Management**:
   - **From User to Process**: Order status inquiries.
   - **From Process to Order Database**: Retrieve order status.
   - **From Order Database to Process**: Status information sent back to the user.
6. **Admin Management**:
   - **From Admin to Process**: Product and user management requests.
   - **From Process to Product Database**: Updates to product information.
   - **From Process to User Database**: Updates to user accounts.

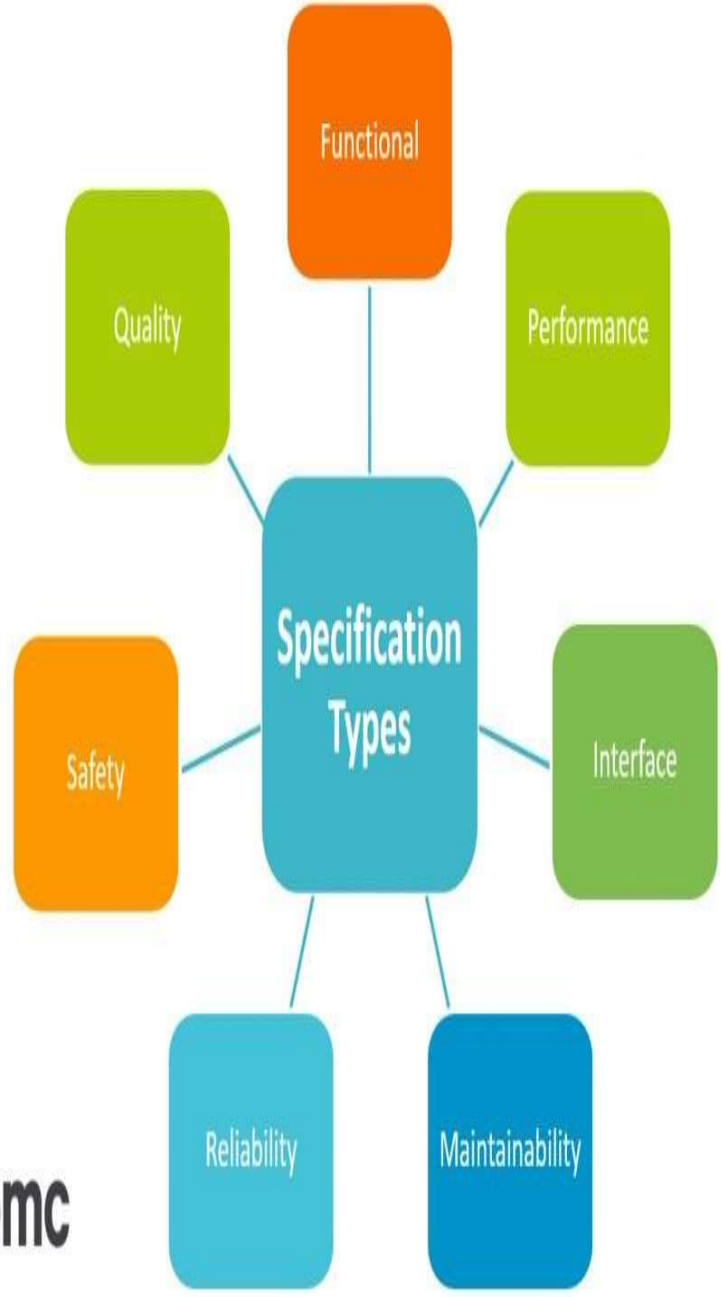
Visual Representation

here’s how we can visualize the DFD:

1. **Level 0 Diagram**:
    - Draw a central circle labeled "Online Farming Equipment System."
    - Connect arrows to and from external entities (Farmer, Payment Gateway, Admin) with labels indicating the data flows.
2. **Level 1 Diagram**:
    - Inside the central circle, create smaller circles for each sub-process.
    - Connect these with arrows to show the flow of data between processes, data stores, and external entities.



Software Requirement Specifications



bmc

FARMING TOOLS WITH PICTURES

Bolo

Sickle

Rake

Pickmattock

Hand fork

Pruning shears

Axe

Hand trowel

Wheelbarrow

Sprayer

Spade

Shovel

Tractor

Seed drill

Appendix for Online Farming Equipment Website

Appendix A: Glossary of Terms

- **User**: An individual who registers on the website to browse and purchase farming equipment.
- **Admin**: Responsible for managing the website's content, products, and user accounts.
- **Shopping Cart**: A feature that allows users to add and manage products before checkout.
- **Payment Gateway**: A service that securely processes online payments.
- **Product CatLog**: Displays available farming equipment for users.
- **Order Confirmation**: Notification sent to the user after a successful purchase.

Appendix B: Use Case Scenarios

- **User Registration**: A farmer creates an account, providing necessary details, and receives a confirmation email.
- **Product Browsing**: Users search for specific equipment, view details, and add items to their cart.
- **Checkout Process**: Users enter payment information and submit orders, receiving a confirmation.

Appendix C: Data Models

- **User Database**:

    - **User ID** (PK), **Name**, **Email**, **Password**, **Address**
- **Product Database**:
    - **Productid** (PK), **ProductName**, **Description**, **Price**, **Stock Quantity**
- **Order Database**:
    - **Order ID** (PK), **User ID** (FK), **Product ID** (FK), **Order Date**, **Payment Status**

Conclusion

In conclusion, this Software Requirements Specification (SRS) document outlines the essential requirements for the Online Farming Equipment website. By detailing both functional and non-functional requirements, this document aims to provide a comprehensive framework that will guide the development process, ensuring the final product meets the needs of its users.

The online farming equipment platform is designed to facilitate seamless interactions between farmers and suppliers, offering a user-friendly interface for browsing, purchasing, and managing farming equipment. The specified requirements will ensure the system is secure, reliable, and scalable, accommodating the evolving needs of its users.

We appreciate the contributions of all stakeholders who provided insights and feedback during the development of this SRS. Their expertise has been invaluable in shaping a clear and effective set of requirements.

Acknowledgments

We would like to thank:

- **Prof. Dr Vishal Kumar** for guidance and support throughout the project.
- **All stakeholders and users** who participated in the requirements gathering process.

References

1. IEEE Std 830-1998, “IEEE Recommended Practice for Software Requirements Specifications.”