**Agri Shop**



**Project ID: Fall-2020-24**

**Session: BSCS Fall 2020 to 2024**

**Project Advisor: Namra Tahir**

**Submitted By**

|  |  |
| --- | --- |
| Zain ul Abideen | 70110969 |
| Khizar Ahmad | 70112282 |
|  |  |

Department of Computer Science & IT

The University of Lahore

Lahore, Pakistan

## *Declaration*

We have read the project guidelines and we understand the meaning of academic dishonesty, in particular plagiarism and collusion. We hereby declare that the work we submitted for our final year project, entitled **AGRI SHOP** is original work and has not been printed, published or submitted before as final year project, research work, publication or any other documentation.

## 

**Group Member 1 Name:**

**SAP No:**

**Signature: …………………………**

**Group Member 2 Name:**

**SAP No:**

**Signature: …………………………**

## *Statement of Submission*

This is to certify that **Zain Ul Abideen** Roll No. **70110969** and **Khizar Ahmad** Roll No. **70112282** have successfully submitted the final project named as: **Agri Shop**, at Computer Science & IT Department, The University of Lahore, Lahore Pakistan, to fulfill the partial requirement of the degree of **BS in Computer Science**.

**Supervisor Name: ………………………**

**Signature…………………………**

**Date: ………………………**

## *Dedication*

This project is dedicated to Allah, whose guidance and blessings have made everything possible; to our beloved father and mother, whose unwavering support and love have been our foundation; to my wife, whose constant support and encouragement have been my strength; and to our esteemed supervisor, whose mentorship has driven us to achieve our best.

## *Acknowledgement*

## 

We truly acknowledge the cooperation and help make by Name of **Acknowledger**, **Designation** of **Address of Organization**. He has been a constant source of guidance throughout the course of this project. We would also like to thank **Acknowledger** from **Designation**, **Address of Organization** for his help and guidance throughout this project. We are also thankful to our friends and families whose silent support led us to complete our project.

Date:

Jan 1, 2020

## *Abstract*

The Agri Shop Web Application is designed to tackle challenges within the agricultural supply chain by offering farmers using this platform to sell their products to consumers. By integrating Text to Speech Real Time (TTS) technologies, the application enhances accessibility for farmers who may be unable to read or write, thereby broadening their market reach and enabling fairer pricing. The platform aims to streamline the process from farm to consumer, reduce delays, and improve product quality. Additionally, it supports secure online payments, offers a user-friendly interface, and includes features like text-to-speech conversion and bidding systems. The application’s revenue model includes transaction fees, premium memberships, and targeted advertisements. It fosters community trust through user reviews, comments, and ratings and plans to generate further engagement through strategic partnerships with agricultural influencers. The Software Requirement Specification outlines the purpose, audience, scope, and functionalities such as user authentication, inventory management, order processing, and customer support. Key considerations include efficient memory management, compliance with regulatory policies, and technology assumptions like scalability and data encryption. Core features are prioritized for the initial release, with future iterations planned based on stakeholder feedback and market demands.

***Area of the Project***

Web Application, Text To Speech Real Time (TTS) Engine

***Technologies used***

HTML, CSS, TAILWAND CSS, REACT, MONGOS DB, PYTHON

## List of Figures

[Figure 1 Overall use case diagram 38](#_Toc187214601)

[Figure 2 Create profile 39](#_Toc187214602)

[Figure 3 view profile 40](#_Toc187214603)

[Figure 4 update profile 41](#_Toc187214604)

[Figure 5 Delete profile 42](#_Toc187214605)

[Figure 6 Account Login 43](#_Toc187214606)

[Figure 7 Create Product listing 45](#_Toc187214607)

[Figure 8 Update Product Listing 47](#_Toc187214608)

[Figure 9 Delete Product listing 49](#_Toc187214609)

[Figure 10 bidding function 51](#_Toc187214610)

[Figure 11 Add to Cart 53](#_Toc187214611)

[Figure 12 voice narrator 54](#_Toc187214612)

[Figure 13 Online Payment 56](#_Toc187214613)

[Figure 14 User Registration management 58](#_Toc187214614)

[Figure 15 inventory management 60](#_Toc187214615)

[Figure 16User authentication: 62](#_Toc187214616)

[Figure 17 Discounts and Coupons: 64](#_Toc187214617)

[Figure 18 Customer Support 66](#_Toc187214618)

[Figure 19 Architecture Diagram 68](#_Toc187214619)

[Figure 20 ERD 69](#_Toc187214620)

[Figure 21 Level 0 DFD 70](#_Toc187214621)

[Figure 22 Level 1 DFD 70](#_Toc187214622)

[Figure 23 Class Diagram 71](#_Toc187214623)

[Figure 24 Activity Diagram Create Account 72](#_Toc187214624)

[Figure 25 Activity Diagram Create Account 72](#_Toc187214625)

[Figure 26 Activity diagram for edit profile 73](#_Toc187214626)

[Figure 27 Activity diagram for customer details 73](#_Toc187214627)

[Figure 28 Activity diagram for search product 74](#_Toc187214628)

[Figure 29 Activity diagram for Text to Speech 74](#_Toc187214629)

[Figure 30 Activity diagram for Bidding system 75](#_Toc187214630)

[Figure 31Activity diagram for inventory 76](#_Toc187214631)

[Figure 32 Activity diagram for payment gateway 77](#_Toc187214632)

[Figure 33 Sequence Diagram Create Account 77](#_Toc187214633)

[Figure 34 Sequence diagram for login Account 78](#_Toc187214634)

[Figure 35 Sequence diagram for search product 78](#_Toc187214635)

[Figure 36 Sequence diagram for Inventory management 79](#_Toc187214636)

[Figure 37 Sequence diagram for customer details 79](#_Toc187214637)

[Figure 38 Sequence diagram for payment gateway 80](#_Toc187214638)

[Figure 39 Sequence diagram for Bidding System 80](#_Toc187214639)

[Figure 40 Sequence diagram for Text to speech 81](#_Toc187214640)

[Figure 41 Aggregated Sequence Diagram 82](#_Toc187214641)

[Figure 42 Collaboration Diagram 83](#_Toc187214642)

[Figure 43 State Transition Diagram 84](#_Toc187214643)

[Figure 44 Component Diagram **Error! Bookmark not defined.**](#_Toc187214644)

[Figure 45 Deployment Diagram 86](#_Toc187214645)

[Figure 46 State transition testing diagram of URM 126](#_Toc187214646)

[Figure 47state transition testing for Add to cart 127](#_Toc187214647)

[Figure 48 State transition testing for Product List 128](#_Toc187214648)

[Figure 49 State transition testing for voice narration 129](#_Toc187214649)

[Figure 50 State transition testing for Bidding system 130](#_Toc187214650)

[Figure 51Use Case Testing for Product listing 131](#_Toc187214651)

[Figure 52 Use Case Testing Bidding system 131](#_Toc187214652)

[Figure 53 Use Case Testing Add to Cart 132](#_Toc187214653)

[Figure 54 Use Case Testing for Voice narrator 132](#_Toc187214654)

[Figure 55 Use Case Testing for Payment process 133](#_Toc187214655)

## List of Tables

[Table 1 Create Account 12](#_Toc187217888)

[Table 2 View Profile 13](#_Toc187217889)

[Table 3 Update Profile 14](#_Toc187217890)

[Table 4 Update password 15](#_Toc187217891)

[Table 5 Delete Profile 16](#_Toc187217892)

[Table 6 Account Login 17](#_Toc187217893)

[Table 7 Create Product Listing 18](#_Toc187217894)

[Table 8 Update Product Listing 19](#_Toc187217895)

[Table 9 Delete Product Listing 20](#_Toc187217896)

[Table 10 Bidding Function 21](#_Toc187217897)

[Table 11 Add to Cart 22](#_Toc187217898)

[Table 12 Voice Narrator 23](#_Toc187217899)

[Table 13 Online Payment 24](#_Toc187217900)

[Table 14 User Registration Management 25](#_Toc187217901)

[Table 15 Inventory Management 26](#_Toc187217902)

[Table 16 Order Processing 27](#_Toc187217903)

[Table 17 Refund Processing 28](#_Toc187217904)

[Table 18 User Authentication 29](#_Toc187217905)

[Table 19 Discounts and Coupons 30](#_Toc187217906)

[Table 20 Customer Support 31](#_Toc187217907)

[Table 21 Positive Test Case for creating profile 88](#_Toc187217908)

[Table 22 Negative Test Case 89](#_Toc187217909)

[Table 23 Positive Test case for view profile 90](#_Toc187217910)

[Table 24 Negative Test case 91](#_Toc187217911)

[Table 25 Positive Test case for inventory management 92](#_Toc187217912)

[Table 26 Negative Test case 93](#_Toc187217913)

[Table 27 Positive Test case for order processing 94](#_Toc187217914)

[Table 28 negative test case 95](#_Toc187217915)

[Table 29 Positive Test case for Bidding function 96](#_Toc187217916)

[Table 30 Negative test case 97](#_Toc187217917)

[Table 31 Positive Test case for Add to cart 98](#_Toc187217918)

[Table 32 Negative test case 99](#_Toc187217919)

[Table 33 Positive Test case for voice narrator 100](#_Toc187217920)

[Table 34 Negative Test case 101](#_Toc187217921)

[Table 35Positive Test case for online payment 102](#_Toc187217922)

[Table 36 Negative Test case 103](#_Toc187217923)

[Table 37 Positive Test case for User Registration management 104](#_Toc187217924)

[Table 38 Negative Test case 105](#_Toc187217925)

[Table 39 BB for Create Profile 106](#_Toc187217926)

[Table 40 BB for view profile 107](#_Toc187217927)

[Table 41BB for inventory management 108](#_Toc187217928)

[Table 42 BB for Bidding Function 109](#_Toc187217929)

[Table 43 BB for Add to Cart 110](#_Toc187217930)

[Table 44 BB for Voice narrator 111](#_Toc187217931)

[Table 45 BB for online payment 112](#_Toc187217932)

[Table 46 BB for urm 113](#_Toc187217933)

[Table 47 Create profile Valid Classes and Invalid Classes 114](#_Toc187217934)

[Table 48 view profile Valid Classes and Invalid Classes 114](#_Toc187217935)

[Table 49 Create Product Listing Valid Classes and Invalid Classes 115](#_Toc187217936)

[Table 50 Bidding Function Valid Classes and Invalid Classes 115](#_Toc187217937)

[Table 51 Online Payment Valid Classes and Invalid Classes 116](#_Toc187217938)

[Table 52 Add to cart Valid Classes and Invalid Classes 116](#_Toc187217939)

[Table 53 URM Valid Classes and Invalid Classes 117](#_Toc187217940)

**Table of Content**

[*Declaration* i](#_Toc187217805)

[*Statement of Submission* ii](#_Toc187217806)

[*Dedication* iii](#_Toc187217807)

[*Acknowledgement* iv](#_Toc187217808)

[*Abstract* v](#_Toc187217809)

[List of Figures viii](#_Toc187217810)

[List of Tables x](#_Toc187217811)

[Chapter 1: Introduction to the Problem 1](#_Toc187217812)

[1.1 Introduction 2](#_Toc187217813)

[1.2 Purpose 2](#_Toc187217814)

[1.3 Objective 2](#_Toc187217815)

[1.4 Existing Solution 3](#_Toc187217816)

[1.5 Proposed Solution 4](#_Toc187217817)

[Chapter 2: Software Requirement Specification 6](#_Toc187217818)

[2.1 Introduction 7](#_Toc187217819)

[2.1.1 Purpose 7](#_Toc187217820)

[2.1.2 Scope 7](#_Toc187217821)

[2.1.3 Definitions, acronyms, and abbreviations 8](#_Toc187217822)

[2.2 Overall description 9](#_Toc187217823)

[2.2.1 Product perspective 9](#_Toc187217824)

[2.2.2 Product functions 12](#_Toc187217825)

[2.2.3 User characteristics 32](#_Toc187217826)

[2.2.4 Constraints 32](#_Toc187217827)

[2.2.5 Assumptions and dependencies 34](#_Toc187217828)

[2.2.6 Apportioning of requirements 35](#_Toc187217829)

[2.3 Specific requirements 36](#_Toc187217830)

[2.3.1 Functional Requirement 36](#_Toc187217831)

[2.3.2 Non-functional Requirements 36](#_Toc187217832)

[Chapter 3: Use Case Analysis 37](#_Toc187217833)

[3.1 Overall Use cases diagram 38](#_Toc187217834)

[3.2 Create Profile: 39](#_Toc187217835)

[3.3 View Profile: 40](#_Toc187217836)

[3.3 Update Profile: 41](#_Toc187217837)

[3.5 Delete profile: 42](#_Toc187217838)

[3.6 Account Login: 43](#_Toc187217839)

[3.7 Create Product linsting: 44](#_Toc187217840)

[3.8 Update product list: 46](#_Toc187217841)

[3.9 Delete Product Listing: 48](#_Toc187217842)

[3.10 Bidding Function: 50](#_Toc187217843)

[3.12 Voice narrator: 54](#_Toc187217844)

[3.13 Online payment: 55](#_Toc187217845)

[3.14 User Registration Management: 57](#_Toc187217846)

[3.15 Inventory management: 59](#_Toc187217847)

[3.18 User authentication: 61](#_Toc187217848)

[3.19 Discounts and Coupons: 63](#_Toc187217849)

[3.18 Discounts and Coupons: 65](#_Toc187217850)

[Chapter 4: Design 67](#_Toc187217851)

[4.1 Architecture Diagram 68](#_Toc187217852)

[4.2 ERD with data dictionary 69](#_Toc187217853)

[4.3 Data Flow diagram 70](#_Toc187217854)

[4.3.1 The level 0 70](#_Toc187217855)

[4.3.2 The level 1 70](#_Toc187217856)

[4.4 Class Diagram 71](#_Toc187217857)

[4.5 Activity Diagram 72](#_Toc187217858)

[4.6 Sequence Diagram 77](#_Toc187217859)

[4.7 Collaboration Diagram 83](#_Toc187217860)

[4.8 State Transition Diagram 84](#_Toc187217861)

[4.9 Component Diagram 85](#_Toc187217862)

[4.10 Deployment Diagram 86](#_Toc187217863)

[Chapter 5: Testing 87](#_Toc187217864)

[5.1 Test Case Specifications 88](#_Toc187217865)

[5.2 Black Box Test Cases 106](#_Toc187217866)

[5.2.1 Equivalence Partitions (EP) 114](#_Toc187217867)

[5.2.2 Boundary Value Analysis 118](#_Toc187217868)

[5.2.3 Decision Table Testing 122](#_Toc187217869)

[2. Product Listing Creation (FR\_06) 123](#_Toc187217870)

[3. Bidding Function (FR\_09) 123](#_Toc187217871)

[4. Online Payment (FR\_12) 124](#_Toc187217872)

[5. Add to Cart (FR\_10) 124](#_Toc187217873)

[6. Voice Narrator (FR\_11) 125](#_Toc187217874)

[5.2.4 State transition Testing 126](#_Toc187217875)

[5.2.5 Use Case Testing 131](#_Toc187217876)

[5.3 White Box Test Cases 134](#_Toc187217877)

[5.3.1 Cyclometric complexity 138](#_Toc187217878)

[5.4 Performance testing 144](#_Toc187217879)

[5.5 Stress Testing 144](#_Toc187217880)

[5.6 System Testing 145](#_Toc187217881)

[5.7 Regression Testing 146](#_Toc187217882)

[Chapter 6: Tools and Techniques 151](#_Toc187217883)

[Chapter 7: Summary and Conclusion 152](#_Toc187217884)

[Chapter 8: User Manual 154](#_Toc187217885)

[Chapter 9: Lessons Learnt and Future Work 162](#_Toc187217886)

[References 164](#_Toc187217887)

### Chapter 1: Introduction to the Problem

##### Introduction

In Pakistan farmers face hurdles such as limited market access, reliance on intermediaries, and a lack of a centralized marketplace. The Agri Shop Web Application aims to address the existing challenges in the agricultural supply chain by providing a platform for farmers to sell their products to consumers. This project seeks to bridge these gaps, empowering farmers to reach a wider consumer base, ensuring fairer pricing, and encouraging a more transparent and efficient agricultural trade ecosystem. Additionally, this web application aims to enhance the overall connectivity and accessibility within the agricultural community, thereby promoting sustainable and farmer-to-consumer transactions in an interactive way through this platform

##### Purpose

The purpose of the Agri Shop Web Application is to address the existing challenges within the agricultural supply chain by providing a efficient platform for farmers to sell their products to consumers. This application is intended to help farmers overcome obstacles such as limited market access, reliance on intermediaries, and the absence of a centralized marketplace. By doing so, it aims to empower farmers to reach a wider consumer base, enable fairer pricing, and foster a more transparent and efficient agricultural trade ecosystem, thus promoting sustainable and farmer-to-consumer transactions. Additionally, the application employs Text To Speech Real Time (TTS) to assist farmers.

##### Objective

* Provide farmers with a platform to sell their products to consumers.
* Expand the market reach for farmers, enabling access to urban.
* Establish a more transparent pricing mechanism where farmers can get fair prices.
* Streamline the process from farm to consumer to reduce delays, cut costs, and enhance the quality of the end product reaching the consumer.
* Utilize Text to Speech Real Time (TTS) to assist the user
* Develop a user-friendly interface that supports secure online payment systems
* Generate revenue through transaction fees, premium memberships, and targeted advertisements within the app.
* Incorporate features like user reviews, comments, and ratings to encourage community trust and encourage informed purchasing decisions.
* Engage in strategic partnerships with agricultural influencers and organizations to broaden the customer base and enhance credibility.

##### Existing Solution

**Khushaal Watan:**

Khushaal Watan is a free Android app designed for farmers to connect with other stakeholders in the farming community. It offers a plethora of localized agricultural content, accurate weather updates, and the ability to connect with agricultural and livestock experts. The app also provides health insurance for farmers and their families, localized mandi rates, and contact details of nearby agricultural shops and retailers.

**Strengths:**

It offers localized agricultural content and weather updates, which are tailored for specific regions, potentially more beneficial for local farmers.

Farmers can connect with agricultural and livestock experts, which is valuable for gaining tailored advice and support. It provides health insurance and contact details for nearby agricultural shops and retailers, enhancing the overall welfare of farmers and their access to resources.

**Weaknesses:**

It does not mention interactive features such as user input forms or real-time communications, which may limit user engagement compared to your web application.

Its Lack a bidding system and online payment options which are crucial for modern e-commerce and ensuring optimal prices. It does not provide language translation or text-to-speech features for broader accessibility.

**Kissan Karobar:**

Kissan Karobar is an online market place where a farmer can buy and sell agriculture items. The app includes separate sections for fruit, vegetable, poultry, and livestock. It provides an e-commerce facility to the farmer making it easy for the rural farmer to get access to the market directly through his smartphone.

**Strengths:**

It directly targets the need for an online marketplace which is excellent for rural farmers to access broader markets. It includes separate sections for various agricultural categories, potentially making it easier for users to navigate and find specific items.

**Weaknesses:**

In this no mention of interactive web views or data visualizations that might enhance user interaction.

It lacks advanced features like bidding systems, language translation, text-to-speech, and comprehensive user reviews which could limit user experience and accessibility. It does not specify if there are flexible payment options like installment plans, which can be a limitation for buyers with financial constraints.

**Bakhabar Kissan:**

Jazz Bakhabar Kissan is a dedicated app that helps farmers to increase their crop yield with updated information and technology. The information provided by this app covers everything from soil preparation to post-harvest, along with livestock and weather updates with audio, video and pictorial presentations. This app indicates how private organizations such as Jazz can contribute to revolutionizing the concept of agriculture in Pakistan.

**Strengths:**

It provides extensive agricultural information from soil preparation to post-harvest, which can significantly help in increasing crop yields.

It utilizes audio, video, and pictorial presentations which can enhance understanding and user engagement.

**Weaknesses:**

It does not provide a platform for buying or selling goods directly or a bidding system which may limit its functionality as a commercial tool. It lacks online payment facilities and user review systems, which are important for building trust and ease of transactions in e-commerce environments. No mention of multiple language support or text-to-speech functionalities for accessibility purposes.

##### Proposed Solution

The Agri Shop app aims to revolutionize user engagement by employing targeted digital marketing, influencer collaborations, and partnerships with agricultural organizations. These strategies are geared towards fostering active participation and increasing platform awareness. Additionally, it seeks to reduce reliance on third-party services for critical functions like payment processing and delivery, enhancing control and minimizing disruptions. Data privacy is a top priority, addressing concerns about breaches and earning the trust of the agricultural community. By streamlining the supply chain and implementing stringent fraud prevention measures, it promises a secure trading environment. Setting itself apart from competitors, the Agri Shop Web Application facilitates between farmers and consumers, offering a comprehensive solution. It integrates Text To Speech Real Time (TTS), providing accessibility features like text-to-speech. This benefits farmers. Moreover, the app presents a structured revenue generation plan through transaction fees, premium memberships, and advertisement opportunities, ensuring sustainable income streams while adding value to users and advertisers alike.

### Chapter 2: Software Requirement Specification

##### Introduction

###### Purpose

The Software Requirements Specifications (SRS) document serves to provide a comprehensive understanding of the requirements for developing the Agri Shop Web Application. Its primary purpose is to offer guidance to stakeholders involved in the project, including the development team, project managers, quality assurance personnel, and other relevant parties. The SRS delineates the functional and non-functional requirements necessary for the successful development and implementation of the software.

The intended audience for this SRS includes various stakeholders with a vested interest in the project's success. This audience comprises the development team, including software engineers, developers, designers, and testers, who will directly contribute to the creation of the platform. Additionally, project managers responsible for overseeing the project's progress, resource allocation, and timeline management, as well as stakeholders such as farmers, consumers, investors, and regulatory bodies, will benefit from the insights provided in the document. Quality assurance personnel tasked with testing and ensuring the quality and reliability of the software are also part of the intended audience.

###### Scope

The software product to be produced by this project is the Agri Shop Web Application. This application aims to revolutionize the agricultural supply chain by providing a platform for farmers to list and sell their products to consumers. It will enable consumers to search for and purchase agricultural products from farmers, facilitating secure online payment processing for transactions. Additionally, the application will support language translation and accessibility features to cater to users with literacy challenges. It will incorporate features such as user reviews, comments, and ratings to foster trust and informed purchasing decisions. However, it will not engage in physical delivery or logistics management of agricultural products, provide financial or legal advice to users, or guarantee the success of individual transactions between farmers and consumers. The application's scope aligns with the objectives and goals defined in higher-level specifications, ensuring consistency across the project's documentation and objectives.

###### Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| Term | Definition |
| Agricultural  Supply Chain | The series of interconnected stages involved in producing, processing,  And distributing agricultural products, from farm to consumer. |
| Intermediaries | Middlemen or third-party entities involved in the distribution and sale of agricultural products, often acting as brokers between farmers and  Consumers. |
| Centralized  Marketplace | A single platform or location where buyers and sellers can interact and  conduct transactions without the need for intermediaries. |
| Text-to-Speech (TTS) Conversion | The capability of converting written text into spoken language by synthesizing human-like speech output, facilitating accessibility for  users who may have difficulty reading written content. |
| User Interface  (UI) | The visual elements and interactive features of a software application  through which users interact with the system. |
| Transaction Fees | Charges imposed on users for each transaction conducted through the Agri Shop Web Application, contributing to the platform's revenue  generation. |

|  |  |
| --- | --- |
| Acronyms | Abbreviations |
| API | |  |  | | --- | --- | | Application Programming Interface |  |  |  | | --- | |  | |
| TTS | Text To Speech |
| UI | User Interface |

##### Overall description

###### Product perspective

* **Product perspective**

The product perspective of the Agri Shop Web Application is with other related products and systems. It is crucial to understand how our product interacts with its environment and interfaces with various components. If the Agri Shop Web Application is independent and self-contained, it will be explicitly stated here. Additionally, a block diagram illustrating the major components of the system, their interconnections, and external interfaces can provide valuable insights.

* **System interfaces**

The Agri Shop Web Application's interface is user-friendly for farmers and buyers alike, allowing for secure sign-ups and quick access to updates via a simple dashboard.

Farmers can effortlessly list their products, while buyers can swiftly find what they need through intuitive search features. The platform facilitates smooth transaction management and fosters easy communication between users. Moreover, its compatibility across various devices ensures seamless accessibility for all users. Overall, the interface is designed with simplicity in mind, prioritizing ease of use and efficiency for enhancing the agricultural trading experience.

* **User interfaces**

The user interface of the Agri Shop Web Application is like a friendly guide for farmers and buyers. It helps them easily navigate the app and find what they're looking for. Farmers can add their products with pictures and descriptions, while buyers can search for items they want to purchase. The app also makes it simple for users to manage their transactions and communicate with each other. It works well on phones, tablets, and computers, so everyone can use it easily. Overall, the interface is designed to be straightforward and helpful, making trading in agriculture more accessible for everyone involved.

* **Hardware interfaces**

The hardware interface of the Agri Shop Web Application connects the software to devices like computers, smartphones, and tablets. It ensures that the app runs smoothly and efficiently on different devices. This interface allows users to interact with the app using touchscreens, keyboards, and other input methods. It also enables features like taking pictures of products using device cameras. The hardware interface ensures that the app's functions, such as processing payments and displaying images, work correctly. It's like a bridge that lets the app communicate with the physical devices it's used on. Overall, the hardware interface makes sure the app is compatible with various devices and easy to use for everyone.

* **Software interfaces**

The software interface of the Agri Shop Web Application is like a translator between the user and the app's functions. It's what allows users to interact with the app and access its features. This interface includes buttons, menus, and screens that users see and interact with. It makes it easy for farmers to list their products and for buyers to search for items they want to purchase. The software interface also manages tasks like processing payments and sending notifications to users. It's designed to be user- friendly, so everyone can use the app without any confusion. Overall, the software interface ensures a smooth and enjoyable experience for users interacting with the app.

* **Communications interfaces**

The communication interface of the Agri Shop Web Application facilitates interactions between the different components of the system. It ensures seamless data exchange between modules, such as between the user interface (UI) and backend systems. This interface includes communication protocols for user authentication, product listings, order management, payment processing, and more. For instance, when a user login, the UI communicates with the authentication service in the backend, sending user credentials and receiving a success or failure response. Similarly, when a user makes a purchase, the product listing module communicates with the inventory management system to update stock levels, and the payment gateway communicates with the transaction service to process payments. These interactions are essential to ensure data consistency and system reliability across all functions, from product creation to order fulfillment and payment processing. The communication interface is designed to handle these exchanges efficiently, ensuring a smooth flow of information between parts of the system, such as between the user interface, backend services, and external systems like payment gateways.

* **Memory**

The Agri Shop Web Application utilizes primary memory (RAM) to temporarily store data required during active processes. RAM ensures fast access to frequently used data for smooth app operation. Secondary memory, such as hard drives or SSDs, stores data long-term, such as user profiles and product listings. While secondary memory offers more storage space, it has slower data access compared to RAM. Efficient management of both memory types is essential to optimize the app’s performance and prevent slowdowns.

* **Operations**

In the Agri Shop Web Application, daily operations involve tasks such as user management, product listing, and transaction processing, all of which ensure the platform runs efficiently and provides a seamless user experience. To ensure the application remains reliable and secure, we plan to implement regular backup operations to safeguard critical data, such as user profiles and product listings, by storing it in secure, remote storage. In case of system failure or emergencies, recovery operations will be in place to restore the system quickly, minimizing downtime and disruptions for users. This approach will ensure that the Agri Shop Web Application remains operational, even during unexpected issues, by balancing normal and special operations effectively. Additionally, continuous monitoring will be implemented to track the system's health and prevent any potential operational failures.

* **Site adaptation requirements**

Site adaptation requirements for the Agri Shop Web Application involve customizing data and initialization sequences to suit specific locations, missions, or operational modes. This includes tailoring the application to accommodate regional preferences, languages, and currencies. Additionally, site-specific requirements may involve integrating local regulations or market dynamics into the application's functionality. Initialization sequences need to be adjusted to ensure seamless deployment and operation of the application in different environments. These adaptations are essential for optimizing user experience and ensuring the application's effectiveness in diverse contexts. Overall, site adaptation requirements aim to make the Agri Shop Web Application relevant, accessible, and functional across various sites, missions, and operational modes

###### Product functions

**Create Profile:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID: | FR\_01 | | | | |
| Name: | Create Profile | | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Enter details to create account | Name, Email, Password etc. | Profile created. | Internet Connectivity required. | Enter correct information and click submit button  System save the record in database. |

Table 1 Create Account

**View Profile:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_02 | | | |
| Name: | View Profile | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Enter details to view profile | User ID (retrieved from session after created) | Display of account details (Name, Email, and Phone Number. | Internet Connectivity required , user must be logged in. | User navigates to the "View Account" page. System retrieves the user's details from the database using the User ID. System displays the user's account details  on the screen. |

Table 2 View Profile

**Update Profile:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_03 | | | |
| Name: | Update Profile | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Enter details to update profile | Name, Email,  Phone Number, Password | Profile update confirmation message. | Internet Connectivity required. | User navigates to the "Update Account" page. User enters new details. User clicks the "Submit" button. System validates the new input. System updates the user record in the database. User receives an "Account updated" confirmation  Message. |

Table 3 Update Profile

**Update password:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_04 | | | |
| Name: | Update password | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Enter details to update password | |  | | --- | | Current Password,  New Password |  |  | | --- | |  | | |  | | --- | | Password  update confirmation message |  |  | | --- | |  | | Internet Connectivity required. | User navigates to the "Login" page. User enters their Email and Password. User clicks the "Login" button. System validates the email and password. If valid, system starts a user session and redirects to the  dashboard. User receives a "Login successful" message or  an error message. |

Table 4 Update password

**Delete Profile:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_05 | | | |
| Name: | Delete Profile | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Enter details to delete profile | Name, Email,  Phone Number, Password | Profile deletion confirmation message. | Internet Connectivity required, user must be logged in. | User navigates to the "Delete Account" page. User clicks the "Delete Account" button. System asks for confirmation. User confirms account  deletion. System deletes the user record from the database. User receives an "Account deleted" confirmation  message. |

Table 5 Delete Profile

**Account Login:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_06 | | | |
| Name: | Account login | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Enter details to account login | Email, Password | Login success or failure message. | Internet Connectivity required. | User navigates to the "Login" page. User enters their Email and Password. User clicks the "Login" button. System validates the email and password. If valid, system starts a user session and redirects to the  Dashboard.The User receives a "Login successful" message or  an error message. |

Table 6 Account Login

**Create Product Listing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_07 | | | |
| Name: | Create Product Listing | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Create Product Listing | Product Name, Category, Price, Quantity, Description, Images. | Product listing created confirmation message. | User must be logged in, Internet connectivity. | Farmer navigates to the "Create Product Listing" page. Farmer enters product details and uploads images. Farmer clicks the "Submit" button. System validates the input. System saves the product listing in the database. Farmer receives a "Product listing created" confirmation message. |

Table 7 Create Product Listing

**Update Product Listing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_08 | | | |
| Name: | Update Product Listing | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Update Product Listing | Product ID, updated Product Name, Category, Price, Quantity, Description, Images. | Product listing updated confirmation message. | User must be logged in, Internet connectivity. | Farmer navigates to the "Update Product Listing" page. Farmer selects the product to update. Farmer enters updated product details and uploads new images. Farmer clicks the "Submit" button. System validates the new input. System updates the product listing in the database. Farmer receives a "Product listing updated" confirmation message. |

Table 8 Update Product Listing

**Delete Product Listing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_09 | | | |
| Name: | Delete Product Listing | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Delete Product Listing | Product ID | Product listing deletion confirmation message. | User must be logged in, Internet connectivity. | Farmer navigates to the "Delete Product Listing" page. Farmer selects the product to delete. Farmer clicks the "Delete" button. System asks for confirmation. Farmer confirms the deletion. System deletes the product listing from the database. Farmer receives a "Product listing deleted"  confirmation message. |

Table 9 Delete Product Listing

**Bidding Function:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_10 | | | |
| Name: | Bidding function | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Allows users to | Product ID, Bid | Confirmation | User must be | User navigates to the product listing page that supports bidding. User views the current highest bid and decides on a bid amount.  User enters the bid amount and clicks the 'Place Bid' button.  The system checks if the bid amount is higher than the current highest bid.  If valid, the system records the bid and updates the highest bid information.  Display confirmation message to the user and  update bid history for the product. |
| place bids on | Amount | message | logged in, |
| agricultural |  | indicating | Internet |
| products listed |  | the bid has | connectivity. |
| for auction, |  | been placed, |  |
| enabling |  | updated bid |  |
| competitive |  | history. |  |
| pricing and fair |  |  |  |
| market value. |  |  |  |

Table 10 Bidding Function

**Add to Cart:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID: | FR\_11 | | | | |
| Name: | Add to Cart | | | | |
| Description | Input | | Output | Requirements | Basic Work Flow |
| Allows users to | Product | ID, | Confirmation | User must be | User navigates to the product listing page.  User selects a product and specifies the quantity.  User clicks the 'Add to Cart' button.  The system checks the availability of the product. Adds it to the user's shopping cart.  Display confirmation message to the user. |
| add products to | Quantity. |  | message | logged in, |
| their virtual |  |  | indicating | Internet |
| shopping cart |  |  | the product | connectivity. |
| for later |  |  | has been |  |
| purchase. |  |  | added to the |  |
|  |  |  | cart. |  |

Table 11 Add to Cart

**Voice Narrator:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_12 | | | |
| Name: | Voice Narrator | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Provides voice | Text displayed on | Audio | User must be | Users navigate to a page with text (e.g., registration, product listing).  The system detects the text to be narrated.  Users click the 'Voice Narrator' button.  The system converts the text to speech.  Audio narration is played, reading out the  text on the screen. |
| narration for | the screen. | narration of | logged in, |
| key |  | the text. | Internet |
| functionalities |  |  | connectivity, |
| to assist users |  |  | Device with |
| who are unable |  |  | audio output |
| to read or write. |  |  | capability. |

Table 12 Voice Narrator

**Online Payment:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_13 | | | |
| Name: | Online Payment | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Allows users to | Payment Method, | Confirmation | Internet | User proceeds to checkout after adding products to the cart.  User selects a preferred payment method (e.g., credit card, debit card, digital wallet).  User enters required payment details (e.g., card number, expiration date, CVV). User clicks the 'Pay Now' button.  The system processes the payment through a secure payment gateway.  Display confirmation message to the user and update the order status. |
| make secure | Card Details or | message | connectivity, |
| online | Digital Wallet | indicating | Secure |
| payments for | Information. | successful | payment |
| products |  | payment, | gateway |
| purchased on |  | updated | integration. |
| the Agri Shop |  | order status. |  |
| Web |  |  |  |
| Application. |  |  |  |

Table 13 Online Payment

**User Registration Management:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_14 | | | |
| Name: | User Registration Management | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Allows admins to approve or reject user registrations. | User registration details. | Confirmation of registration approval or rejection. | Admin privileges, Internet connectivity. | Admin accesses the user registration management section.  Admin reviews  pending user registration details.  Admin approves or rejects the user registration.  System updates user status and notifies user via email.  Display confirmation message to the admin. |

Table 14 User Registration Management

**Inventory Management:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_15 | | | |
| Name: | Inventory Management | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Enables admins to monitor stock levels, update inventory, and manage low stock alerts. | Product ID, Stock quantity. | Updated inventory levels, Low stock alerts. | Admin privileges, Internet connectivity. | Admin navigates to the inventory management section.  Admin enters or updates stock quantities for products. System updates the inventory levels in the database.  System generates low stock alerts for products below threshold levels.  Display confirmation message to the admin. |

Table 15 Inventory Management

**Order Processing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_16 | | | |
| Name: | Order Processing | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Allows admins to view, process, update order statuses (pending, shipped, delivered, etc.) | Order ID, Updated Status | Updated order status, Tracking information, Confirmation message. | Admin privileges, Internet connectivity. | Admin navigates to the order management section.  Admin selects an order to view its details.  Admin updates the order status (e.g., from pending to shipped).  System saves the updated status in the database.  Display confirmation message to the admin.  Customer views their updated order details through their account. |

Table 16 Order Processing

**Refund Processing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_17 | | | |
| Name: | Refund Processing | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Allows admins to handle and process refund requests submitted by customers. | Refund Request ID, Refund Amount. | Confirmation of refund processing. | Admin privileges, Internet connectivity. | Customer submits a refund request through their account.  Admin reviews the refund request in the refund management section.  Admin approves or rejects the refund request.  If approved, admin processes the refund through the appropriate payment gateway.  System updates the transaction and order status to reflect the refund.  Display confirmation message to the customer and admin. |

Table 17 Refund Processing

**User Authentication:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_18 | | | |
| Name: | User Authentication | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Implements strong authentication mechanisms, including two- factor authentication (2FA), to  Ensure secure access to the Agri Shop Web Application. | User credentials (username, password), 2FA code. | Access granted or denied message. | Internet connectivity, Integration with 2FA service provider. | User navigates to the login page.  User enters their username and password.  System verifies the entered credentials.  If credentials are correct, system prompts user to enter a 2FA code sent to their registered device (e.g., phone, email).  User enters the 2FA code.  System verifies the 2FA code.  If the 2FA code is correct, system grants access to the user and redirects them to their dashboard.  If credentials or 2FA code are incorrect, system displays an error message and denies access. |

Table 18 User Authentication

**Discounts and Coupons:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_19 | | | |
| Name: | Discounts and Coupons | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Allows admins | Discount code | Confirmation | Admin | Admin navigates to the |
| to create and | details (code, | of discount | privileges, | marketing and |
| manage | percentage or | code creation | Internet | promotions section. |
| discount codes and promotions for customers. | amount off,  expiration date, usage limits). | or update. | connectivity. | Admin selects the  option to add or edit a discount code.  Admin enters discount |
|  |  |  |  | code details including |
|  |  |  |  | code, percentage or |
|  |  |  |  | amount off, expiration |
|  |  |  |  | date, and usage limits. |
|  |  |  |  | Admin saves the |
|  |  |  |  | discount code |
|  |  |  |  | configuration. |
|  |  |  |  | System updates the |
|  |  |  |  | discount codes in the |
|  |  |  |  | database. |
|  |  |  |  | Display confirmation |
|  |  |  |  | message to the admin. |
|  |  |  |  | Customers can enter |
|  |  |  |  | discount codes during |
|  |  |  |  | checkout to receive the |
|  |  |  |  | specified discount. |

Table 19 Discounts and Coupons

**Customer Support:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID: | FR\_20 | | | |
| Name: | Customer Support | | | |
| Description | Input | Output | Requirements | Basic Work Flow |
| Provides tools | Support ticket | Confirmation | Admin | Customer submits a |
| for managing | details (customer | of ticket | privileges, | support ticket or |
| customer | name, issue | resolution or | Internet | inquiry through the |
| support tickets and inquiries, as well as creating and managing a knowledge base and FAQs to assist customers. | description), FAQ details (question, answer). | FAQ  creation/update. | connectivity. | help desk section.  Admin accesses the customer support management section to view and manage support tickets.  Admin reviews and responds to support tickets, resolving issues as needed. |
|  |  |  |  | System updates the |
|  |  |  |  | status of the support |
|  |  |  |  | ticket and notifies the |
|  |  |  |  | customer of the |
|  |  |  |  | resolution. |
|  |  |  |  | Admin navigates to |
|  |  |  |  | the knowledge base |
|  |  |  |  | management section |
|  |  |  |  | to add or update |
|  |  |  |  | FAQs. |
|  |  |  |  | Admin enters FAQ |
|  |  |  |  | details including |
|  |  |  |  | question and answer. |
|  |  |  |  | Admin saves the |
|  |  |  |  | FAQ. |
|  |  |  |  | System updates the |
|  |  |  |  | knowledge base with |
|  |  |  |  | the new or updated |
|  |  |  |  | FAQ. |
|  |  |  |  | Customers can access |
|  |  |  |  | the FAQs and |
|  |  |  |  | knowledge base for |
|  |  |  |  | self-service support. |

Table 20 Customer Support

###### User characteristics

The intended users of the Agri Shop Web Application vary widely in terms of educational level, experience, and technical expertise. Farmers, who form a significant portion of the user base, may have varying levels of education from basic literacy to advanced degrees in agriculture. Their experience with technology ranges from minimal to moderate, primarily through basic smartphone use. To accommodate their needs, the application should be intuitive and provide ample guidance.

Consumers, on the other hand, generally have higher educational backgrounds and are familiar with e-commerce platforms. They are comfortable with online transactions and navigating digital interfaces. Therefore, the application should offer advanced search options, secure payment gateways, and a seamless user experience.

Agricultural experts and advisors typically possess specialized degrees and extensive experience in the agricultural sector. They have high technical expertise and use digital tools for research and communication. The application should include features like data analytics and expert forums to meet their needs.

Other stakeholders, such as investors and regulatory bodies, also interact with the application. These users often have higher education qualifications in business, finance, or law and are proficient in using complex software systems. The application should provide detailed reports and secure data access to cater to their requirements. Understanding these diverse user characteristics ensures the Agri Shop Web Application is accessible, user- friendly, and meets the varied needs of its users.

###### Constraints

* **Regulatory Policies**

The application must comply with regulatory frameworks governing agricultural trade, consumer protection, and data privacy. This includes adhering to laws and standards related to online transactions, data handling, and user privacy to ensure legal compliance and user trust.

* **Hardware Limitations**

The performance and capabilities of the hardware used by farmers and consumers, such as smartphones and computers, impose constraints on the application. It must be optimized to function effectively across various devices with varying processing power, memory, and screen sizes.

* **Interfaces to Other Application**

Integration with external systems or applications, such as payment gateways, weather APIs, or government databases, imposes constraints on data exchange protocols and compatibility requirements. The application must seamlessly interact with these interfaces to provide accurate information and smooth transactions.

* **Parallel Operation**

The application's ability to handle multiple concurrent users and transactions, known as parallel operation, imposes constraints on scalability, performance, and resource management. It must efficiently manage server resources and database access to prevent bottlenecks during peak usage periods.

* **Audit and Control Functions**

To ensure accountability and transparency, the application may require audit trails and control functions to track user actions, manage permissions, and enforce security policies. These functions impose constraints on data logging, access control mechanisms, and system monitoring.

* **Reliability and Safety Considerations**

The criticality of agricultural transactions and user data necessitates reliability and safety measures. The application must ensure data integrity, fault tolerance, and disaster recovery capabilities to mitigate risks of system failures or data breaches.

* **Higher-Order Language Requirements**

Depending on the target user base and operational environment, the application may have language requirements for localization and accessibility. Multilingual support and cultural adaptation impose constraints on user interfaces, content translation, and communication protocols.

* **Signal Handshake Protocols**

For seamless communication between different components of the application, signal handshake protocols impose constraints on data synchronization, error handling, and message integrity. Proper protocol implementation ensures reliable data exchange and system interoperability.

* **Criticality of the Application**

Given the application's role in facilitating agricultural transactions and connecting farmers with consumers, its criticality imposes constraints on uptime, response times, and service availability. The application must prioritize reliability and performance to support continuous operations.

* **Safety and Security Considerations**

To protect user data, financial transactions, and sensitive information, the application must adhere to stringent security measures. Constraints include encryption standards, access controls, authentication mechanisms, and vulnerability management to safeguard against cyber threats and unauthorized access.

###### Assumptions and dependencies

* **Operational environment Assumptions**

Assumptions regarding the operational environment include factors such as the availability of stable internet connectivity for users accessing the Agri Shop Web Application. It assumes that users will have access to devices like smartphones or computers capable of running the application smoothly. Additionally, it assumes basic familiarity with using digital platforms among both farmers and consumers, ensuring they can effectively engage with the application's features.

* **External Dependencies**

Dependencies on external systems or services, such as payment gateways, weather APIs for agricultural updates, or government databases for regulatory compliance, are critical. The functionality and performance of the Agri Shop Web Application depend on seamless integration with these external interfaces. Changes or disruptions in these dependencies could impact the application's ability to provide accurate information and conduct transactions securely.

* **Regulatory and Legal Assumptions**

Assumptions related to regulatory and legal compliance involve adherence to laws governing e-commerce, data privacy, and agricultural trade. It assumes that the application will comply with relevant regulations, ensuring user trust and legal operation. Changes in regulatory requirements could necessitate updates to the application's functionalities and data handling practices.

* **Business and Organizational Assumptions**

Business assumptions may include expectations about revenue models, user adoption rates, and partnerships with agricultural stakeholders. For instance, assumptions about generating revenue through transaction fees or advertising within the application influence its design and feature prioritization. Changes in business strategies or market conditions may require adjustments to the application's business logic and operational framework.

* **Technology Assumptions**

Assumptions about technological capabilities, such as the scalability of server infrastructure, the effectiveness of data encryption methods, and the compatibility of software across different devices and operating systems, are also critical. The application assumes that technological solutions implemented during development will effectively support its operational requirements and user demands.

###### Apportioning of requirements

* **Prioritization of Core Features**

During initial development phases, core features essential for the application's basic functionality and usability are prioritized. These include functionalities such as user registration, product listing and searching, secure online transactions, and basic communication tools between farmers and consumers. By focusing on these foundational elements, the development team ensures the application's viability and immediate utility to users.

* **Deferred Features**

Certain features that are deemed non-critical or require more extensive development effort may be postponed to future releases. These could include advanced analytics and reporting capabilities, integration with complex third-party systems, additional language support, or enhanced user customization options. Deferring these features allows the team to concentrate on delivering a robust initial version of the application while planning for incremental improvements and enhancements over time.

* **Stakeholder Feedback and Iterative Development**

The decision to apportion requirements is influenced by ongoing stakeholder feedback and iterative development practices. Input from farmers, consumers, agricultural experts, and other stakeholders helps prioritize features based on user needs and market demands. Agile methodologies, such as Scrum or Kanban, facilitate adaptive planning and continuous refinement of requirements, enabling the team to respond effectively to changing priorities and emerging opportunities.

* **Roadmap for Future Releases**

The Apportioning of Requirements section serves as a roadmap for future releases of the Agri Shop Web Application. It outlines a strategic approach to phased feature implementation, balancing immediate user requirements with long-term project goals. By outlining clear timelines and milestones for deferred features, the development team can maintain transparency with stakeholders and manage expectations regarding the evolution of the application.

##### Specific requirements

###### Functional Requirement

Functional Requirements outline the specific modules and functionalities of the Agri Shop Web Application. These include:

* **User Registration:** Farmers and consumers should be able to create and manage their accounts securely.
* **Product Listing and Search:** Farmers can list agricultural products with descriptions and images, while consumers can search and browse products.
* **Transaction Management:** Facilitate secure online transactions between farmers and consumers, ensuring reliability and data integrity.
* **Communication Tools:** Provide messaging or commenting features to foster interaction and trust between users.
* **Accessibility Features:** Implement Natural Language Processing (NLP) for text-to-speech and language translation to support users with literacy challenges.
* **Revenue Generation:** Include mechanisms for transaction fees, premium memberships, and targeted advertisements to sustain the platform financially.
* **User Feedback:** Enable user reviews, ratings, and comments to enhance transparency and informed decision-making.

###### Non-functional Requirements

Non-functional Requirements specify attributes of the Agri Shop Web Application beyond its specific functionalities. These include:

* **Usability:** Ensure the application is intuitive and easy to navigate for users with varying technical expertise.
* **Reliability:** Maintain high availability and minimize downtime to ensure continuous access to the platform.
* **Performance:** Support concurrent users and handle peak loads efficiently to provide responsive user experiences.
* **Design Constraints:** Adhere to regulatory, legal, and technical standards governing e- commerce and data privacy.
* **Portability:** Ensure compatibility across devices and platforms (e.g., smartphones, tablets, computers) to maximize accessibility

# Chapter 3: Use Case Analysis

## 3.1 Overall Use cases diagram

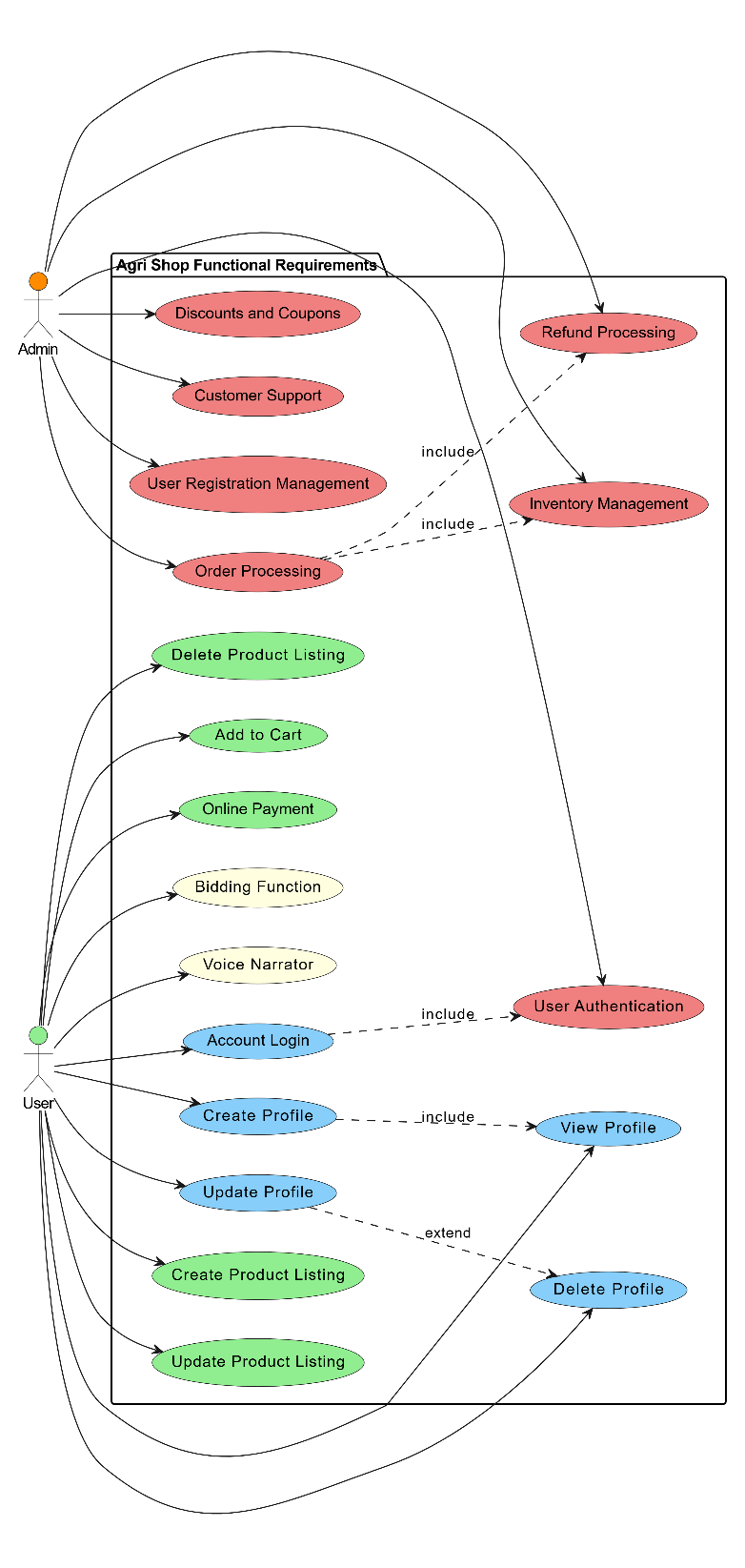


Figure 1 Overall use case diagram

## 3.2 Create Profile:

|  |  |
| --- | --- |
| Use Case ID | FR\_01 |
| Use Case Name | Create Profile |
| Description | Allows a user to create a new profile on the Agri Shop web application. |
| Primary Actor | User |
| Secondary Actor | None |
| Pre-Condition | The user must have internet connectivity. |
| Post-Condition | A new profile is created and saved in the database. |
| Basic Flow | The user begins by entering their name, email, password, and other necessary details into the designated fields. After ensuring all information is correct, the user clicks the submit button. The system then validates the provided input and, upon successful validation, saves the user's information in the database. Finally, the system confirms that the profile has been created successfully. |

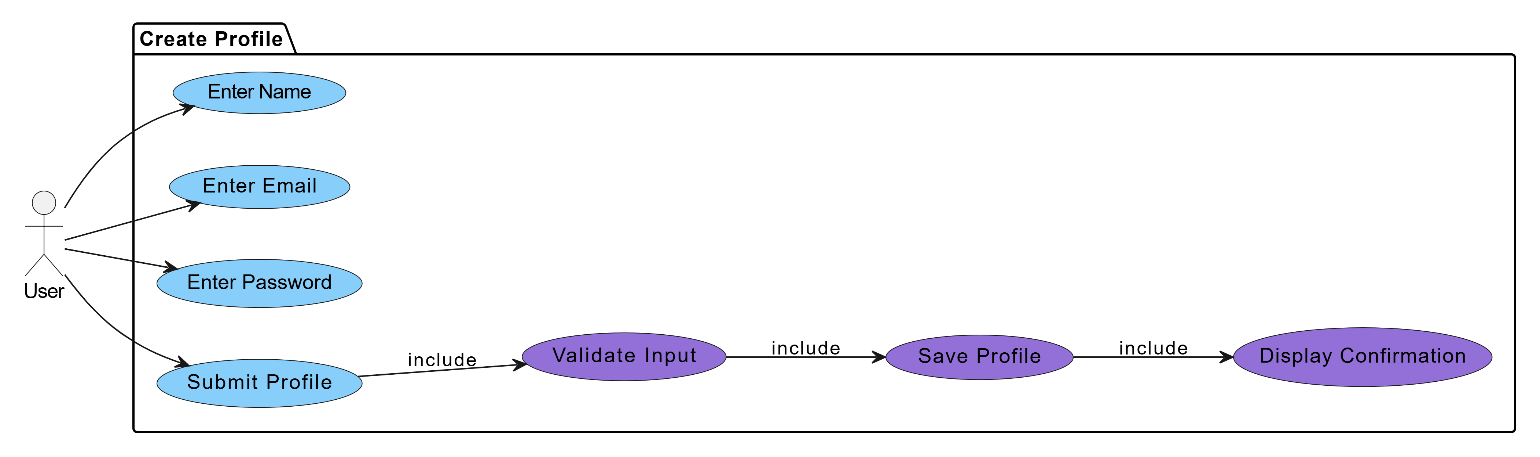


Figure 2 Create profile

## 3.3 View Profile:

|  |  |
| --- | --- |
| Use Case ID | FR\_02 |
| Use Case Name | View Profile |
| Description | Allows a user to view their profile details on the Agri Shop web application. |
| Primary Actor | User |
| Secondary Actor | None |
| Pre-Condition | The user must be logged in and have an internet connection. |
| Post-Condition | The user's profile details are displayed on the screen. |
| Basic Flow | User navigates to the "View Account" page. System retrieves the user's details from the database using the User ID. System displays the user's account details on the screen. |

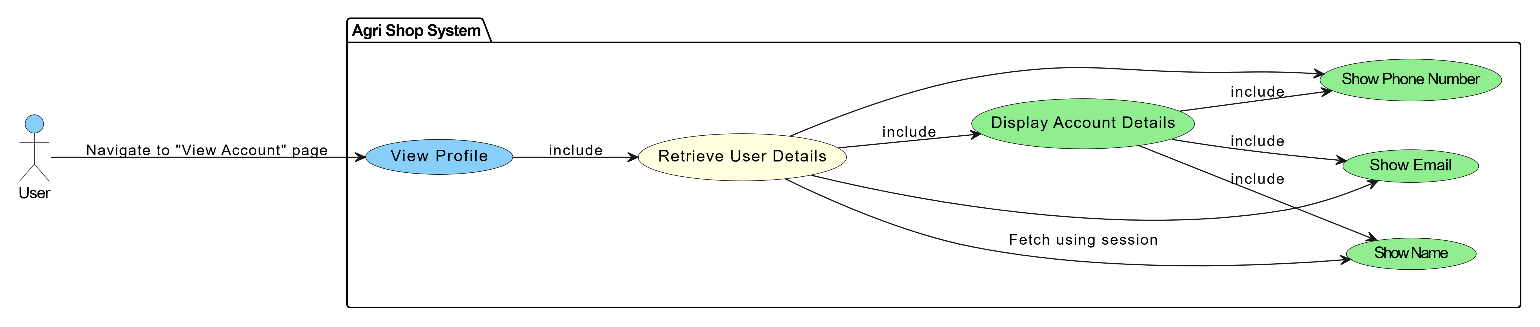


Figure 3 view profile

## 3.3 Update Profile:

|  |  |
| --- | --- |
| Use Case ID | FR\_03 |
| Use Case Name | Update Profile |
| Description | Allows a user to update their profile details on the Agri Shop mobile application. |
| Primary Actor | User |
| Secondary Actor | None |
| Pre-Condition | The user must be logged in and have an internet connection. |
| Post-Condition | The user's profile is updated, and a confirmation message is displayed. |
| Basic Flow | User navigates to the "Update Account" page. User enters new details. User clicks the "Submit" button. System validates the new input. System updates the user record in the database. User receives an "Account updated" confirmation message. |



Figure 4 update profile

## 3.5 Delete profile:

|  |  |
| --- | --- |
| Use Case ID | FR\_04 |
| Use Case Name | Delete profile |
| Description | Allows a user to delete their profile details on the Agri Shop mobile application. |
| Primary Actor | User |
| Secondary Actor | None |
| Pre-Condition | The user must be logged in and have an internet connection. |
| Post-Condition | The user's profile is deleted, and a confirmation message is displayed. |
| Basic Flow | User navigates to the "Delete Account" page. User clicks the "Delete Account" button. System asks for confirmation. User confirms account deletion. System deletes the user record from the database. User receives an "Account deleted" confirmation message. |

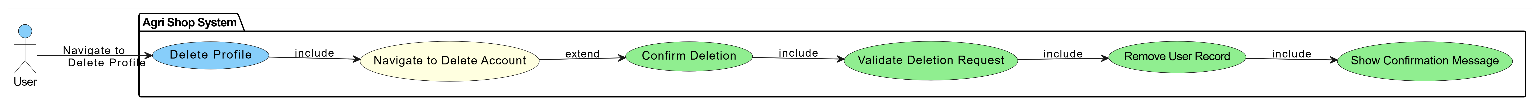


Figure 5 Delete profile

## 3.6 Account Login:

|  |  |
| --- | --- |
| Use Case ID | FR\_05 |
| Use Case Name | Account Login |
| Description | Allows a user to log in to the Agri Shop mobile application using their email and password. |
| Primary Actor | User |
| Secondary Actor | None |
| Pre-Condition | The user must have a registered account and an internet connection. |
| Post-Condition | User is logged in, and a success or error message is displayed. |
| Basic Flow | User navigates to the "Login" page. User enters their Email and Password. User clicks the "Login" button. System validates the email and password. If valid, the system starts a user session and redirects to the dashboard. User receives a "Login successful" message or an error message. |

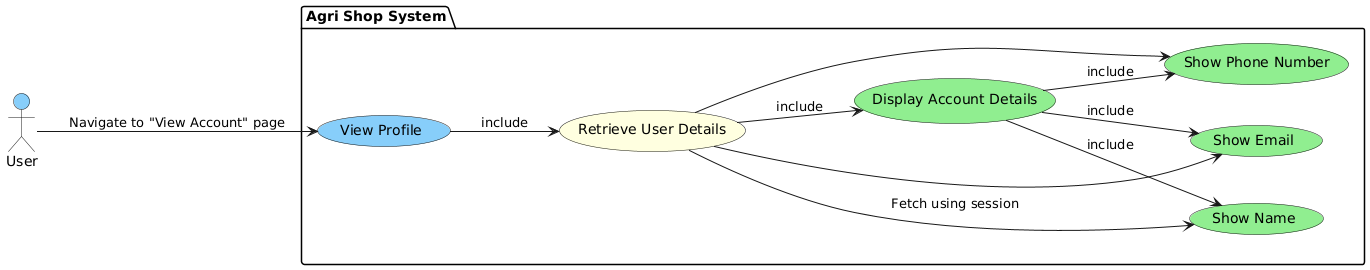


Figure 6 Account Login

## 3.7 Create Product linsting:

|  |  |
| --- | --- |
| Use Case ID | FR\_06 |
| Use Case Name | Create Product Listing |
| Description | Allows a farmer to create a new product listing by providing details an  Uploading images. |
| Primary Actor | Farmer |
| Secondary Actor | None |
| Pre-Condition | The farmer must have a registered account and an internet connection. |
| Post-Condition | The product listing is created, and a confirmation message is displayed. |
| Basic Flow | Farmer navigates to the "Create Product Listing" page. Farmer enters product details and uploads images. Farmer clicks the "Submit" button. System validates the input. System saves the product listing in the database. Farmer receives a "Product listing created" confirmation message. |

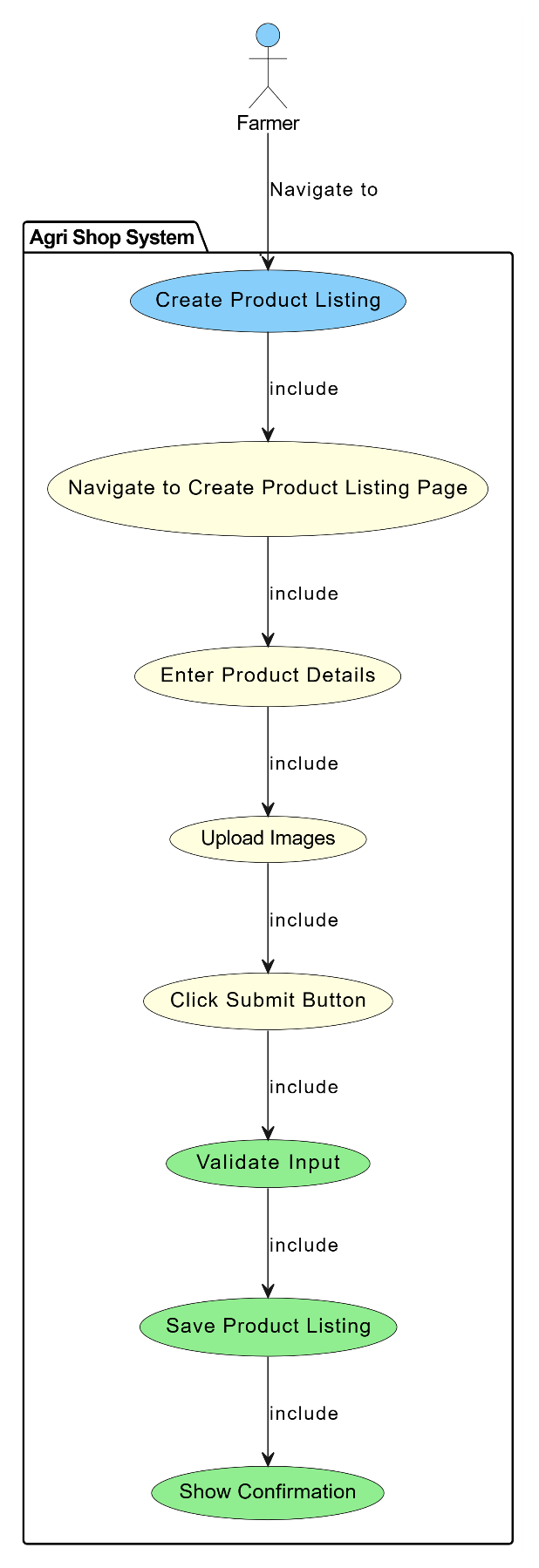


Figure 7 Create Product listing

## 3.8 Update product list:

|  |  |
| --- | --- |
| Use Case ID | FR\_07 |
| Use Case Name | Update product listing |
| Description | Allows a farmer to update an existing product listing by providing updated details and uploading new images. |
| Primary Actor | Farmer |
| Secondary Actor | None |
| Pre-Condition | The farmer must have a registered account and an internet connection. |
| Post-Condition | The product listing is updated, and a confirmation message is displayed. |
| Basic Flow | Farmer navigates to the "Update Product Listing" page. Farmer selects the product to update. Farmer enters updated product details and uploads new images. Farmer clicks the "Submit" button. System validates the new input. System updates the product listing in the database. Farmer receives a "Product listing updated" confirmation message. |

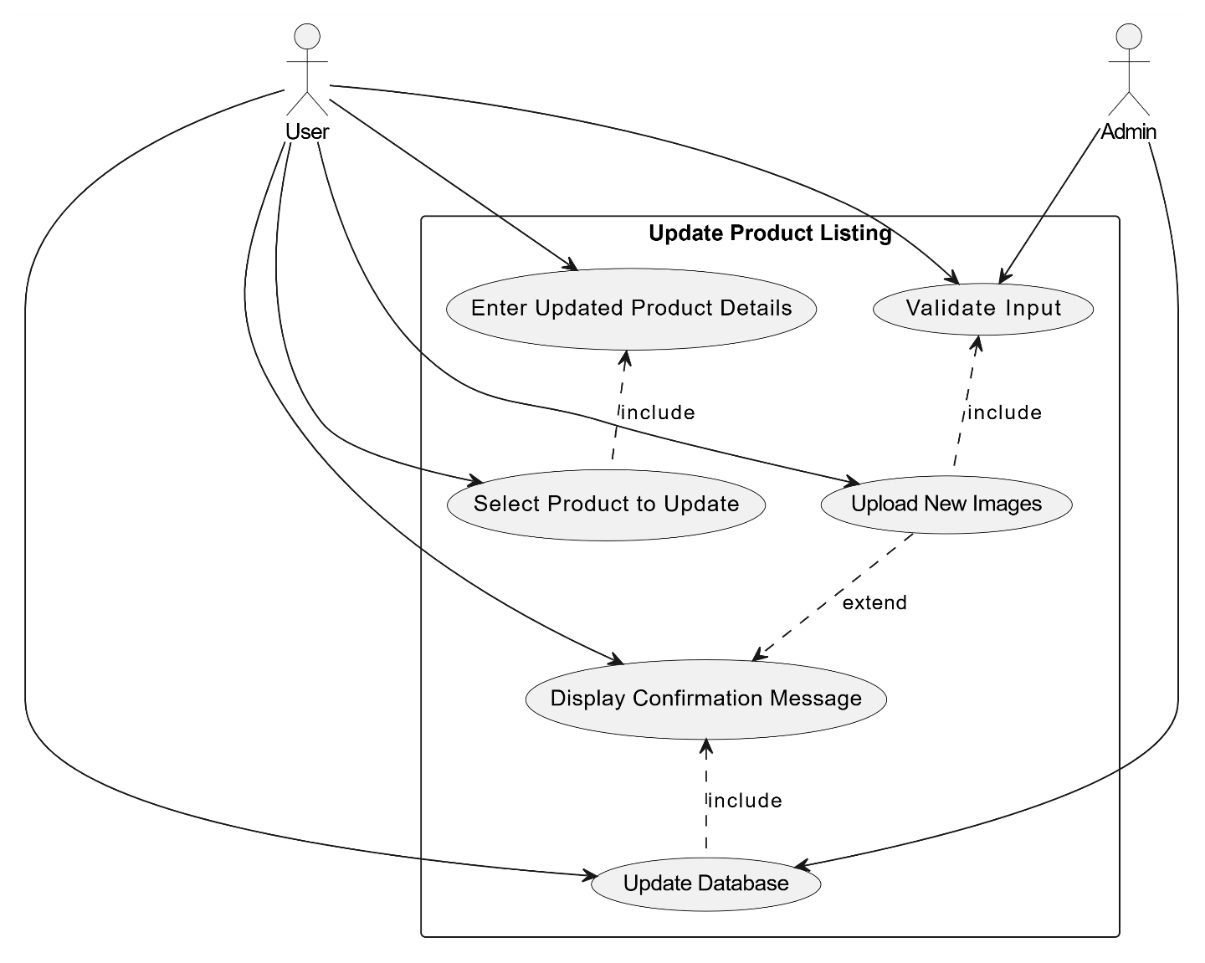


Figure 8 Update Product Listing

## 3.9 Delete Product Listing:

|  |  |
| --- | --- |
| Use Case ID | FR\_08 |
| Use Case Name | Delete Product Listing |
| Description | Allows a farmer to delete an existing product listing from the system. |
| Primary Actor | Farmer |
| Secondary Actor | None |
| Pre-Condition | The farmer must have a registered account and an internet connection. |
| Post-Condition | The product listing is deleted, and a confirmation message is displayed. |
| Basic Flow | Farmer navigates to the "Delete Product Listing" page. Farmer selects the product to delete. Farmer clicks the "Delete" button. System asks for confirmation. Farmer confirms the deletion. System deletes the product listing from the database. Farmer receives a "Product listing deleted" confirmation message. |



Figure 9 Delete Product listing

## 3.10 Bidding Function:

|  |  |
| --- | --- |
| Use Case ID | FR\_09 |
| Use Case Name | Bidding Function |
| Description | Allows users to place bids on agricultural products listed for auction, facilitating competitive pricing and fair market value. |
| Primary Actor | User |
| Secondary Actor | None |
| Pre-Condition | The user must be logged in and have an internet connection. |
| Post-Condition | The bid is placed, and the bid history is updated. |
| Basic Flow | User navigates to the product listing page that supports bidding. User views the current highest bid and decides on a bid amount. User enters the bid amount and clicks the 'Place Bid' button. The system checks if the bid amount is higher than the current highest bid. If valid, the system records the bid and updates the highest bid information. Display confirmation message to the user and update bid history for the product. |

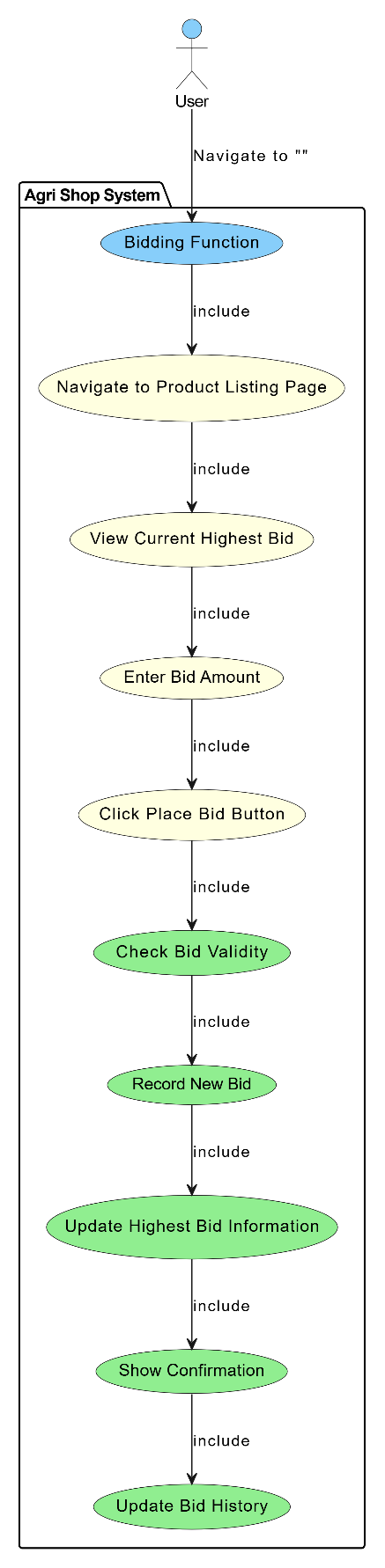


Figure 10 bidding function

**3.11 Add to cart:**

|  |  |
| --- | --- |
| Use Case ID | FR\_10 |
| Use Case Name | Add to cart |
| Description | Allows users to add products to their virtual shopping cart for later purchase |
| Primary Actor | User |
| Secondary Actor | None |
| Input | Product ID, Quantity |
| Output | Confirmation message indicating the product has been added to the cart |
| Basic Flow | User navigates to the product listing page. User selects a product and specifies the quantity. User clicks the 'Add to Cart' button. The system checks the availability of the product. Adds it to the user's shopping cart. Display confirmation message to the user. |

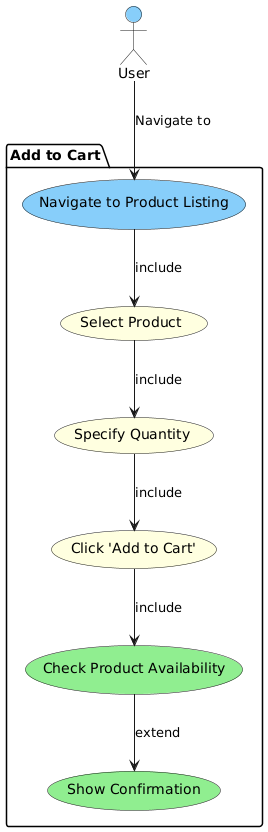
****

Figure 11 Add to Cart

## 3.12 Voice narrator:

|  |  |
| --- | --- |
| Use Case ID | FR\_11 |
| Use Case Name | Voice narrator |
| Description | Provides voice narration for key functionalities to assist users who are unable to read or write. |
| Primary Actor | User |
| Secondary Actor | None |
| Input | Text displayed on the screen. |
| Output | Audio narration of the text. |
| Basic Flow | Users navigate to a page with text (e.g., registration, product listing). The system detects the text to be narrated. Users click the 'Voice Narrator' button. The system converts the text to speech. Audio narration is played, reading out the text on the screen. |

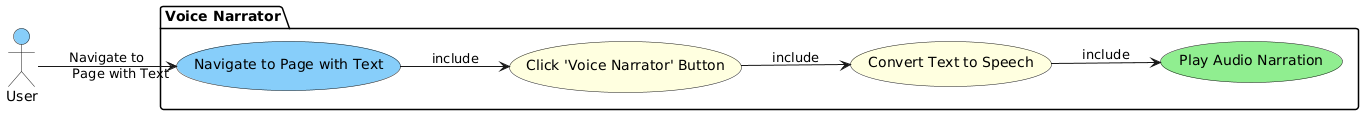


Figure 12 voice narrator

## 3.13 Online payment:

|  |  |
| --- | --- |
| Use Case ID | FR\_12 |
| Use Case Name | Online Payment |
| Description | Allows users to make secure online payments for products purchased on the Agri Shop Web Application |
| Primary Actor | User |
| Secondary Actor | None |
| Input | Payment Method, Card Details or Digital Wallet Information. |
| Output | Confirmation message indicating successful payment, updated order status. |
| Basic Flow | User proceeds to checkout after adding products to the cart. User selects a preferred payment method (e.g., credit card, debit card, digital wallet). User enters required payment details (e.g., card number, expiration date, CVV). User clicks the 'Pay Now' button. The system processes the payment through a secure payment gateway. Display confirmation message to the user and update the order status. |

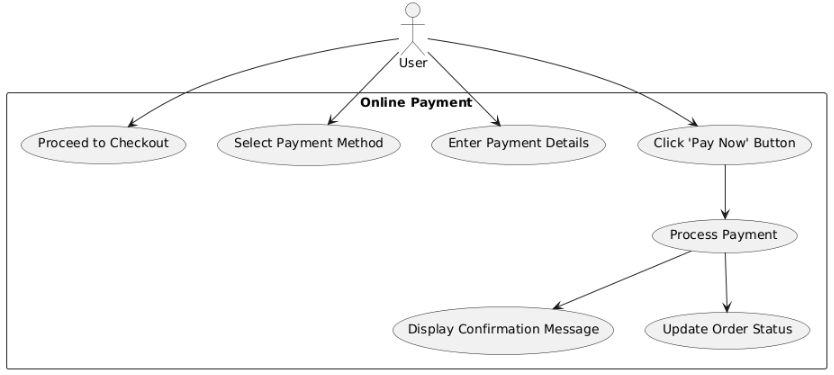


Figure 13 Online Payment

## 3.14 User Registration Management:

|  |  |
| --- | --- |
| Use Case ID | FR\_13 |
| Use Case Name | User Registration Management |
| Description | Allows admins to approve or reject user registrations. |
| Primary Actor | Admin |
| Secondary Actor | None |
| Input | User registration details. |
| Output | Confirmation of registration approval or rejection. |
| Basic Flow | Admin accesses the user registration management section. Admin reviews pending user registration details. Admin approves or rejects the user registration. System updates user status and notifies users via email. Display confirmation message to the admin. |

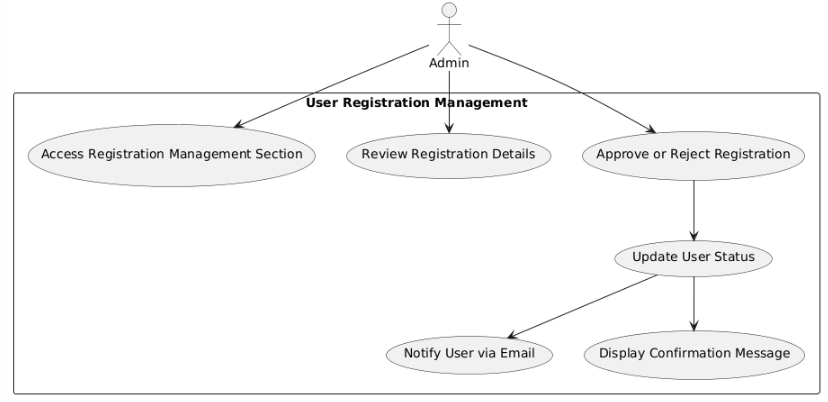


Figure 14 User Registration management

## 3.15 Inventory management:

|  |  |
| --- | --- |
| Use Case ID | FR\_14 |
| Use Case Name | Inventory management |
| Description | Enables admins to monitor stock levels, update inventory, and manage low stock alerts. |
| Primary Actor | Admin |
| Secondary Actor | None |
| Input | Product ID, Stock quantity |
| Output | Updated inventory levels, Low stock alerts |
| Basic Flow | Admin navigates to the inventory management section.Admin enters or updates stock quantities for products.System updates the inventory levels in the database.System generates low stock alerts for products below threshold levels.Display confirmation message to the admin |

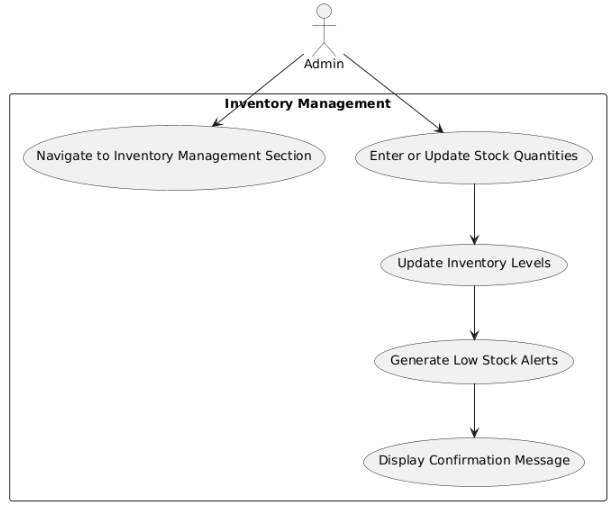


Figure 15 inventory management

## 3.18 User authentication:

|  |  |
| --- | --- |
| Use Case ID | FR\_17 |
| Use Case Name | User authentication |
| Description | Implements strong authentication mechanisms, including two-factor authentication (2FA), to ensure secure access to the Agri Shop web application. |
| Input | User credentials (username, password), 2FA code |
| Output | Access granted or denied messageg |
| Basic Flow | Users log in with their username and password. If correct, they enter a 2FA code sent to their device. The system verifies the code and grants access if it's correct, otherwise, it shows an error message. |

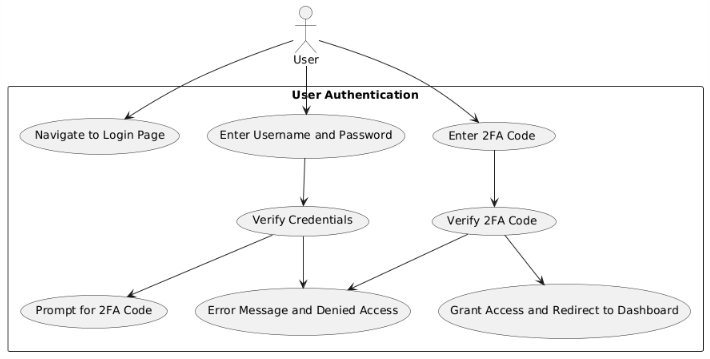


Figure 16User authentication:

## 3.19 Discounts and Coupons:

|  |  |
| --- | --- |
| Use Case ID | FR\_18 |
| Use Case Name | Discounts and Coupons |
| Description | Allows admins to handle and process refund requests submitted by customers. |
| Primary Actor | Admin |
| Secondary Actor | Customer |
| Input | Discount code details (code, percentage or amount off, expiration date, usage limits) |
| Output | Confirmation of discount code creation or update |
| Basic Flow | Admin navigates to the marketing and promotions section. Admin selects the option to add or edit a discount code. Admin enters discount code details including code, percentage or amount off, expiration date, and usage limits. Admin saves the discount code configuration. System updates the discount codes in the database. Display confirmation message to the admin. Customers can enter discount codes during checkout to receive the specified discount |

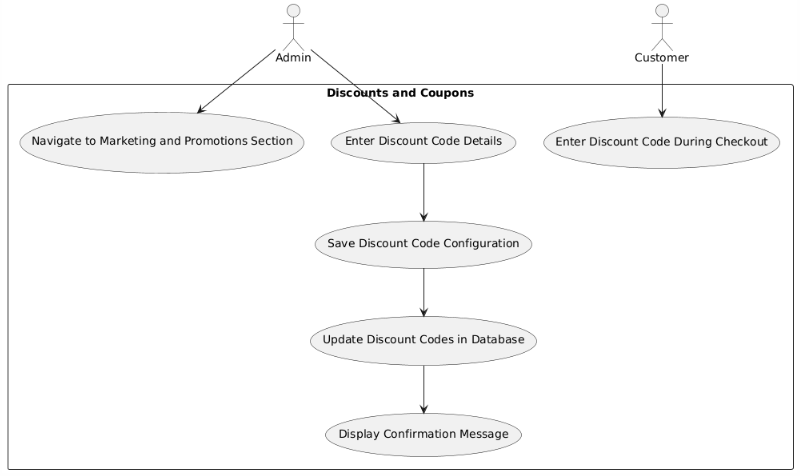


Figure 17 Discounts and Coupons:

## 3.18 Discounts and Coupons:

|  |  |
| --- | --- |
| Use Case ID | FR\_19 |
| Use Case Name | Customer support |
| Description | Provides tools for managing customer support tickets and inquiries, as well as creating and managing a knowledge base and FAQs to assist customers. |
| Primary Actor | Admin |
| Secondary Actor | Customer |
| Input | Support ticket details (customer name, issue description), FAQ details (question, answer) |
| Output | Confirmation of ticket resolution or FAQ creation/update |
| Basic Flow | Customers submit support tickets or inquiries. Admins manage and respond to these tickets, updating their status and notifying customers of resolutions. Admins also add or update FAQs in the knowledge base. The system updates the knowledge base, and customers can access FAQs for self-service support. |

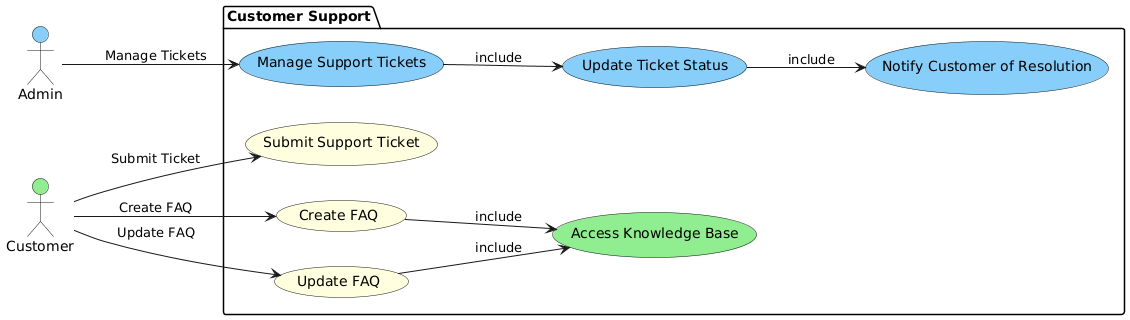


Figure 18 Customer Support

# Chapter 4: Design

#### Architecture Diagram

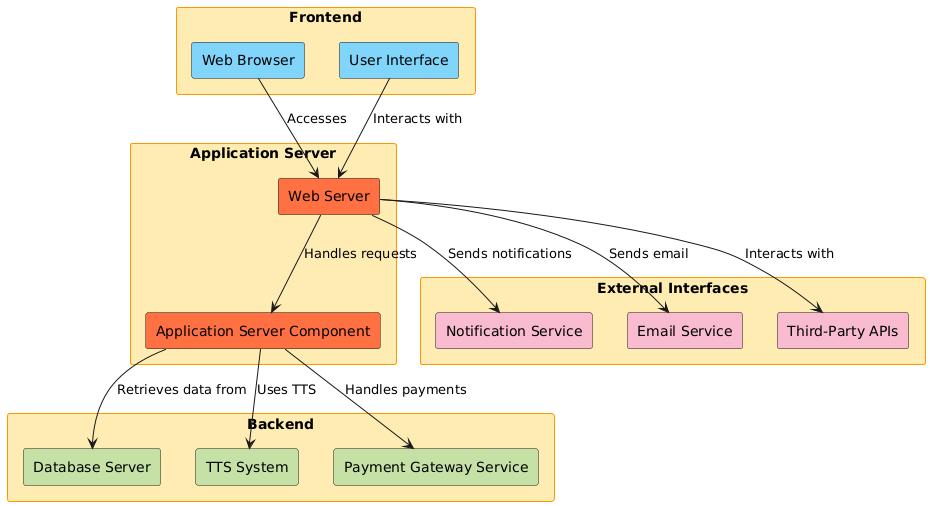


Figure 19 Architecture Diagram

#### ERD with data dictionary

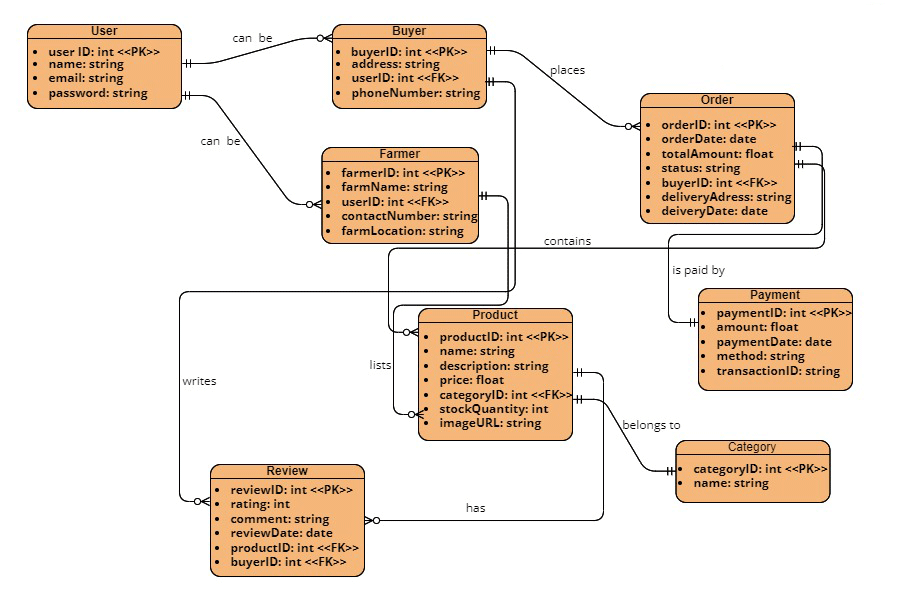


Figure 20 ERD

#### Data Flow diagram

###### The level 0

The flow of information inside the system is defined in this level

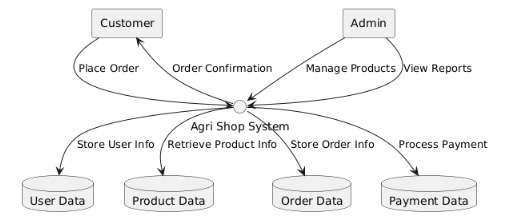


Figure 21 Level 0 DFD

###### The level 1

The flow of information outside the system is defined in this level

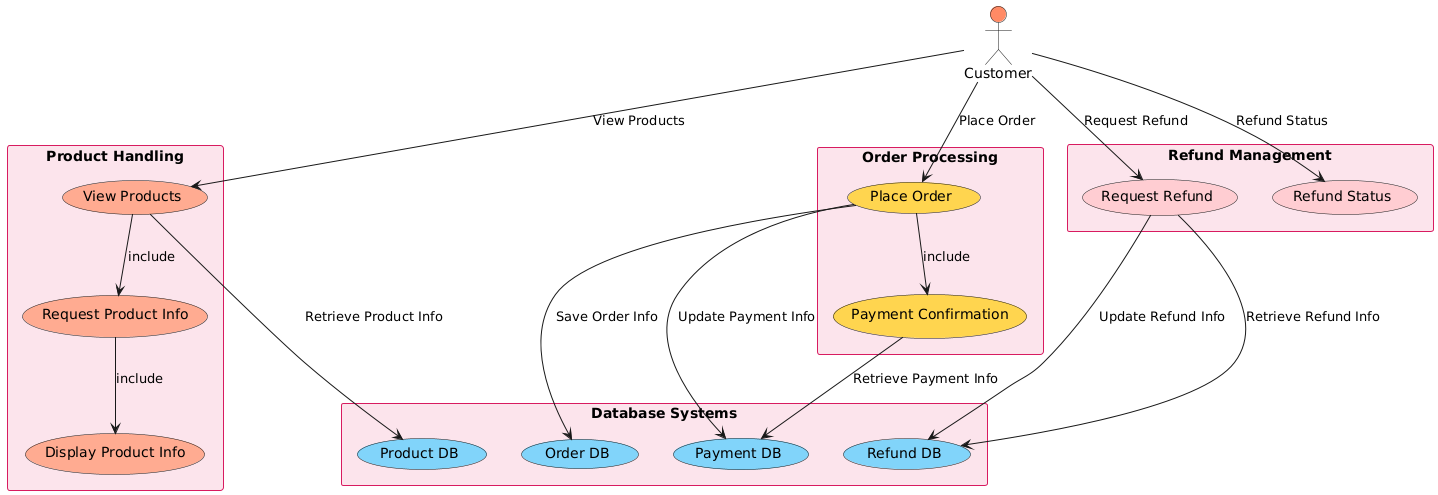


Figure 22 Level 1 DFD

#### Class Diagram

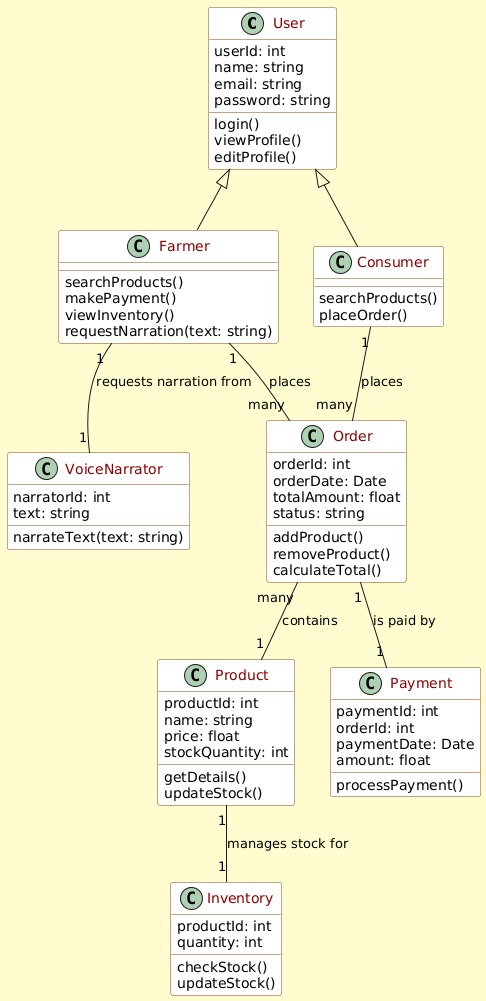


Figure 23 Class Diagram

#### Activity Diagram

**Activity diagram for create account:**

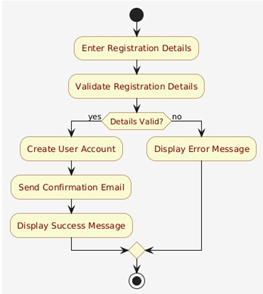


Figure 24 Activity Diagram Create Account

**Activity diagram for login account:**

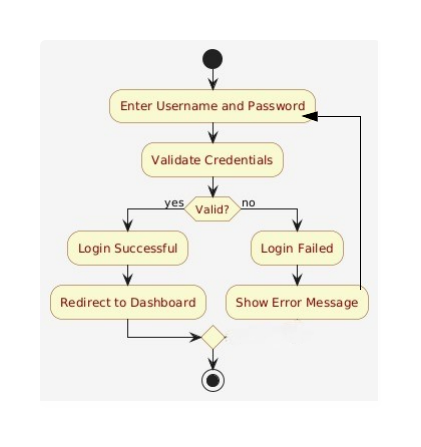
****

Figure 25 Activity Diagram Create Account

**Activity diagram for edit profile:**

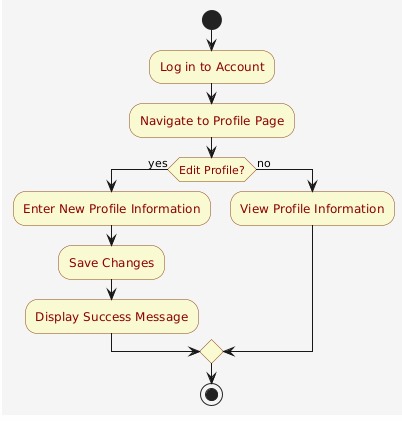


Figure 26 Activity diagram for edit profile

**Activity diagram for customer details:**

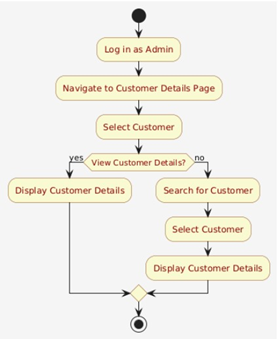


Figure 27 Activity diagram for customer details

**Activity diagram for search product:**

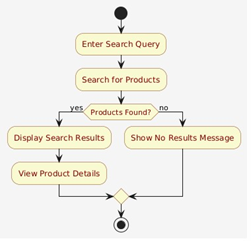


Figure 28 Activity diagram for search product

**Activity diagram for Text to Speech:**

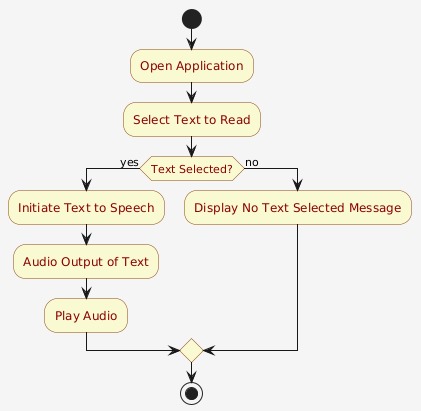


Figure 29 Activity diagram for Text to Speech

**Activity diagram for Bidding system:**

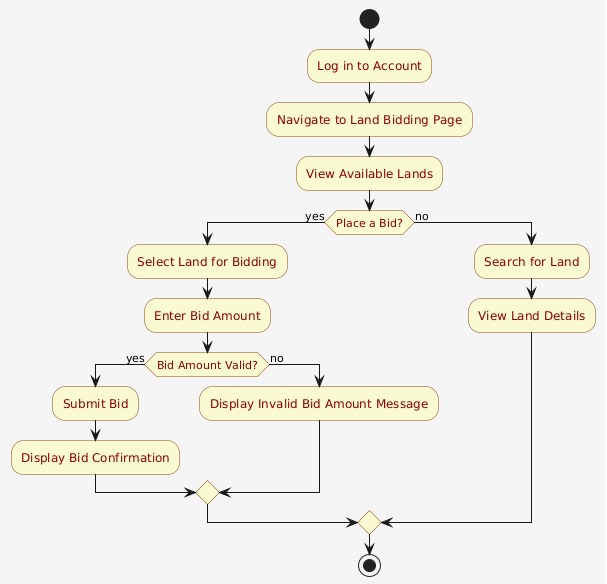


Figure 30 Activity diagram for Bidding system

**Activity diagram for inventory:**

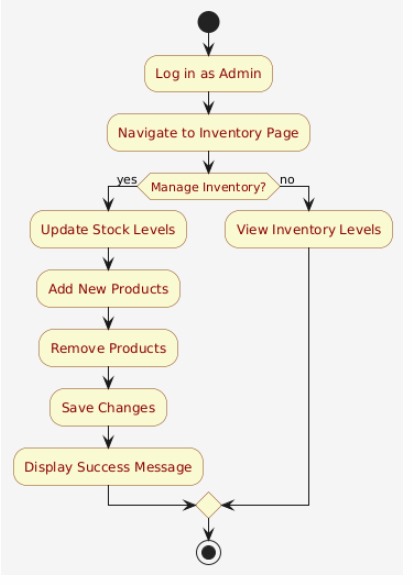


Figure 31Activity diagram for inventory

**Activity diagram for payment gateway:**

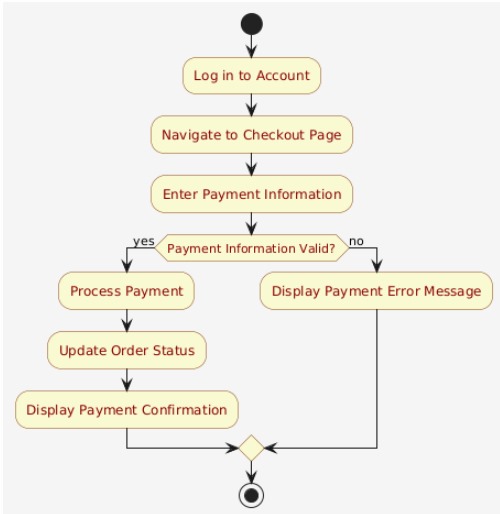


Figure 32 Activity diagram for payment gateway

#### Sequence Diagram

**Sequence diagram for create account**

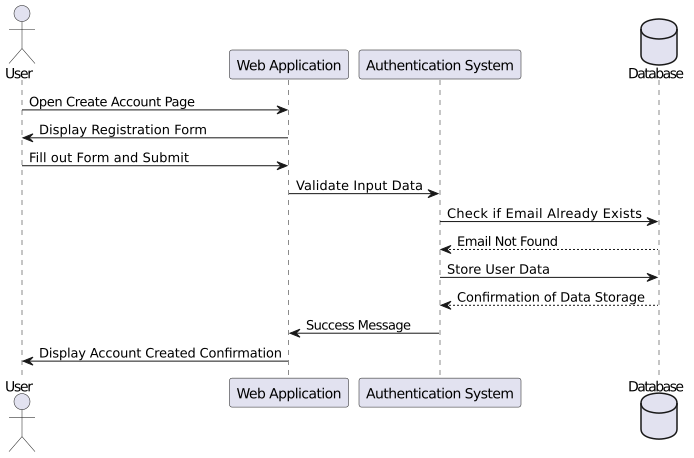


Figure 33 Sequence Diagram Create Account

**Sequence diagram for login Account:**

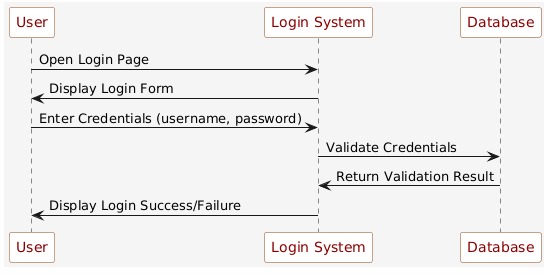


Figure 34 Sequence diagram for login Account

**Sequence diagram for search product:**

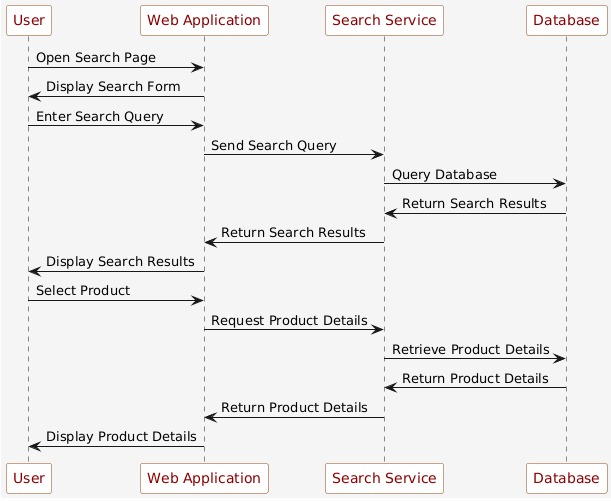


Figure 35 Sequence diagram for search product

**Sequence diagram for Inventory management:**

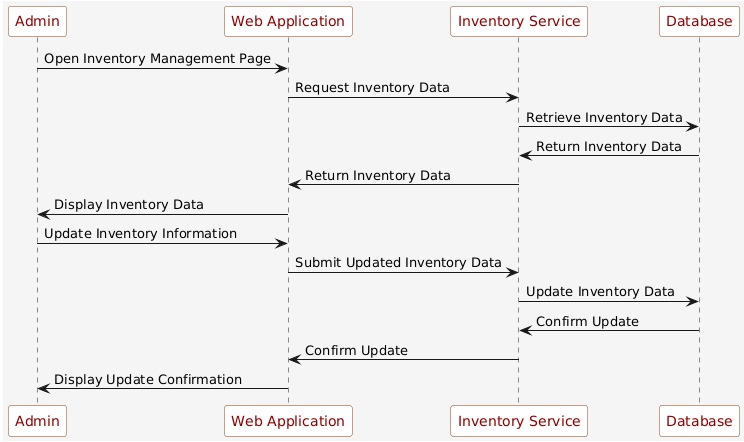


Figure 36 Sequence diagram for Inventory management

**Sequence diagram for customer details:**

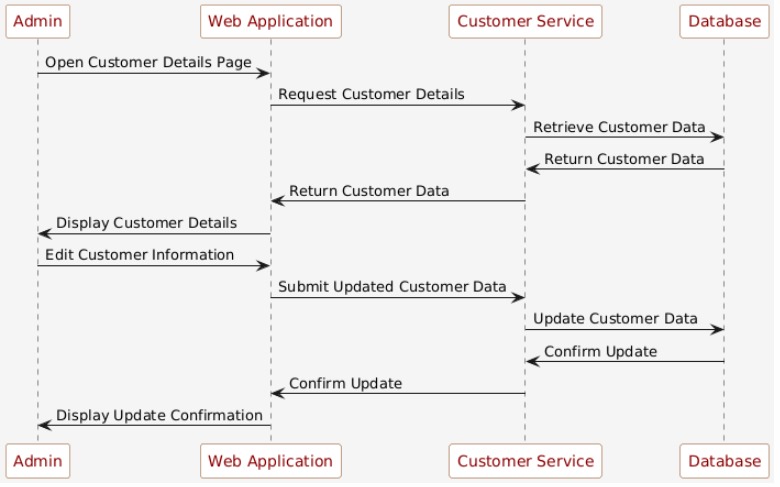


Figure 37 Sequence diagram for customer details

**Sequence diagram for payment gateway:**



Figure 38 Sequence diagram for payment gateway

**Sequence diagram for Bidding System:**

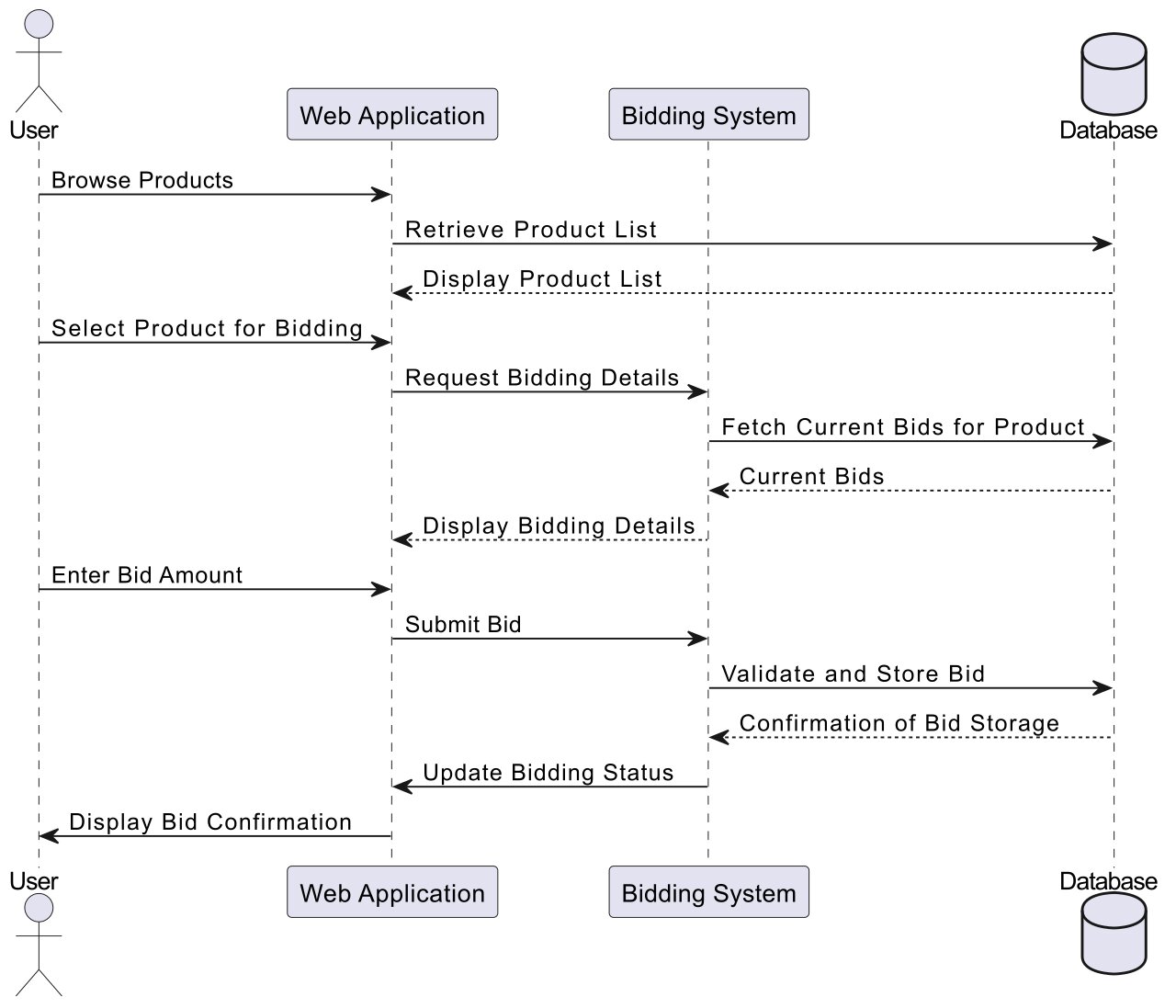


Figure 39 Sequence diagram for Bidding System

**Sequence diagram for Text to speech:**

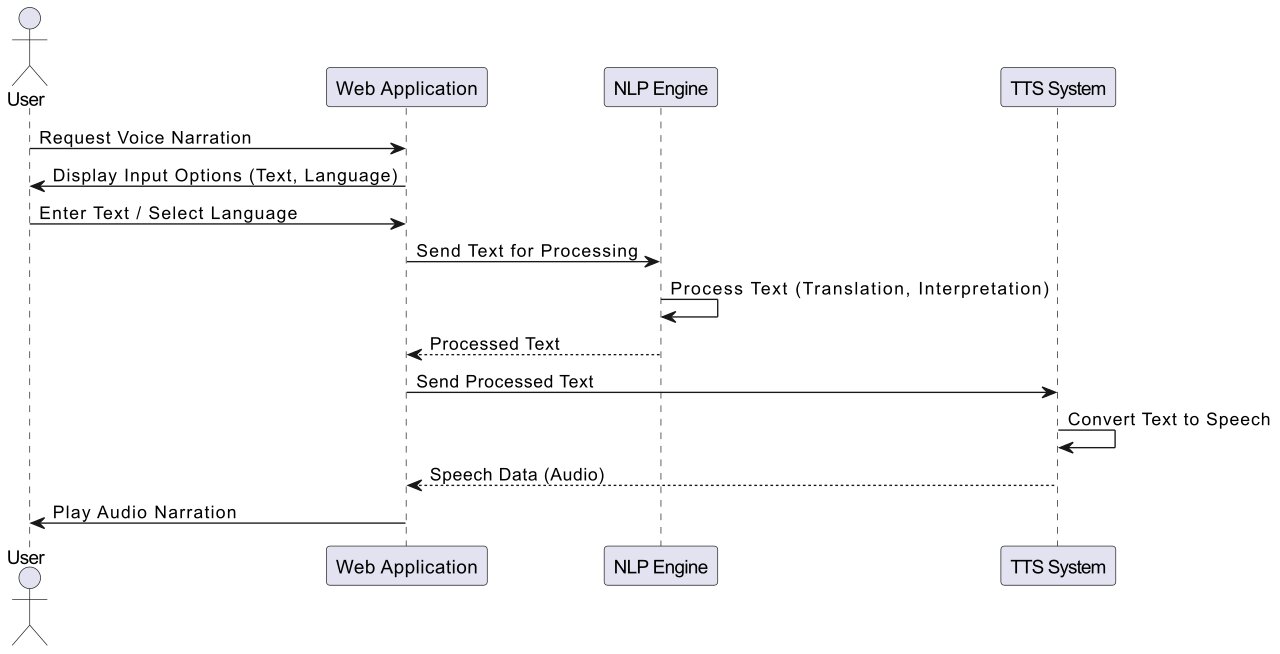


Figure 40 Sequence diagram for Text to speech

**Aggregated Sequence Diagram:**

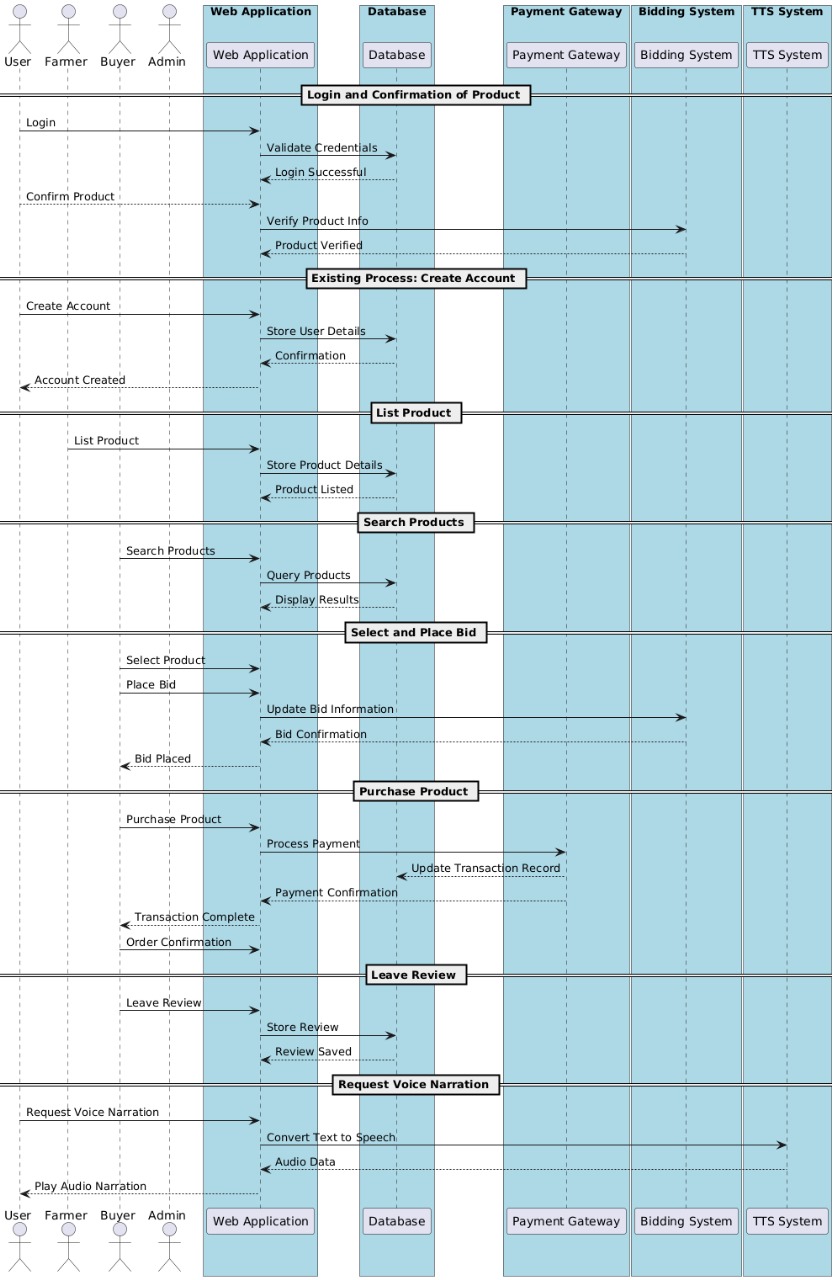
****

Figure 41 Aggregated Sequence Diagram

#### Collaboration Diagram

#### 

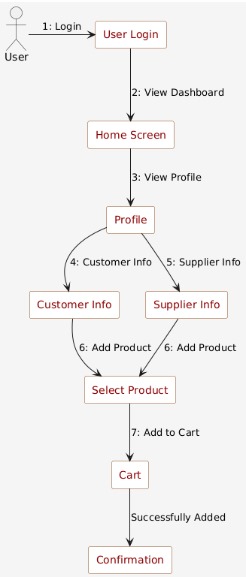


Figure 42 Collaboration Diagram

#### State Transition Diagram

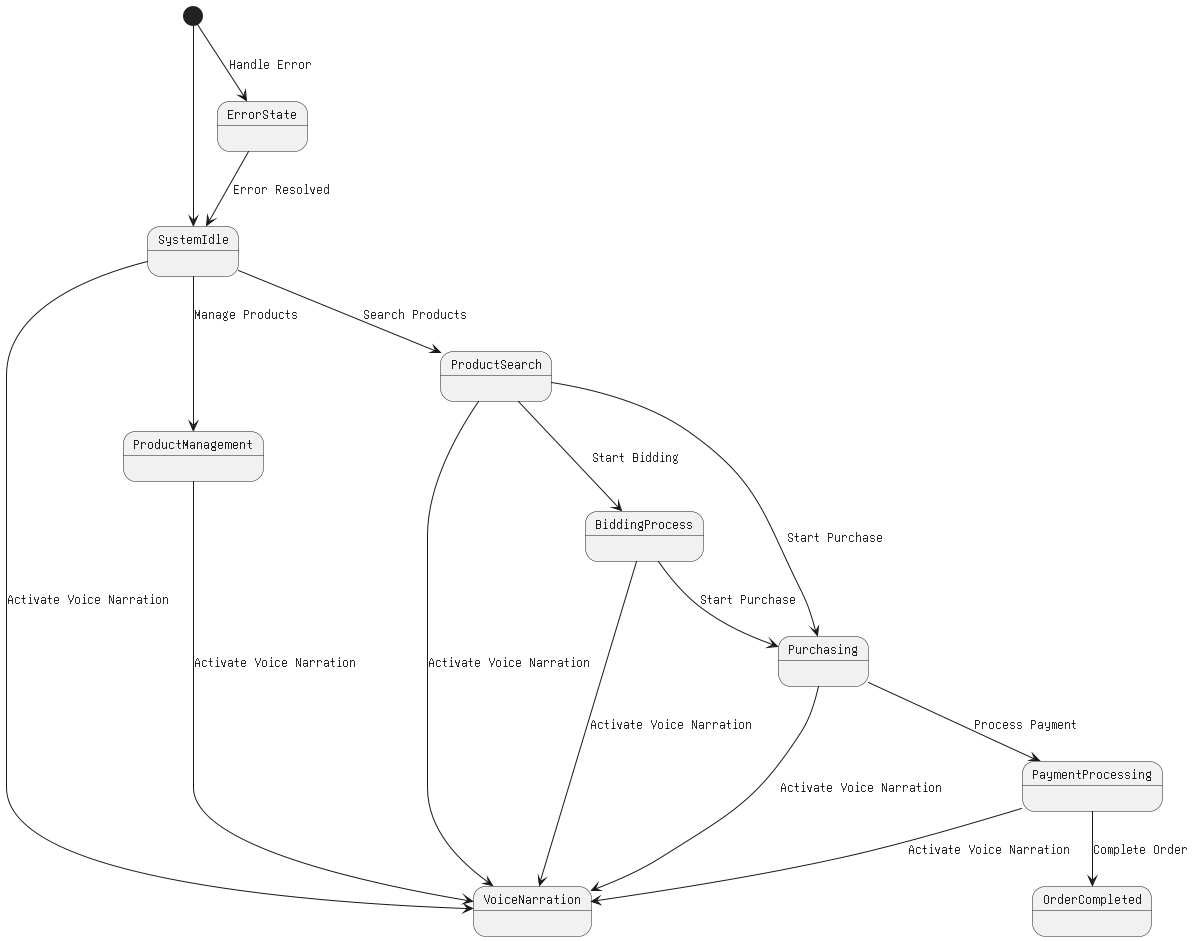


Figure 43 State Transition Diagram

#### Component Diagram

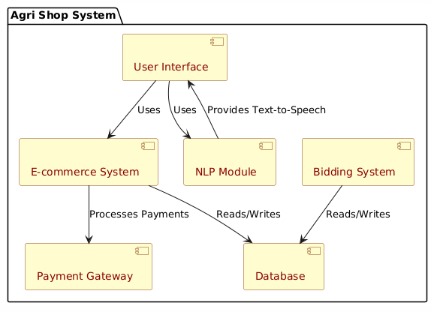


Figure 44 Component Diagram

#### Deployment Diagram

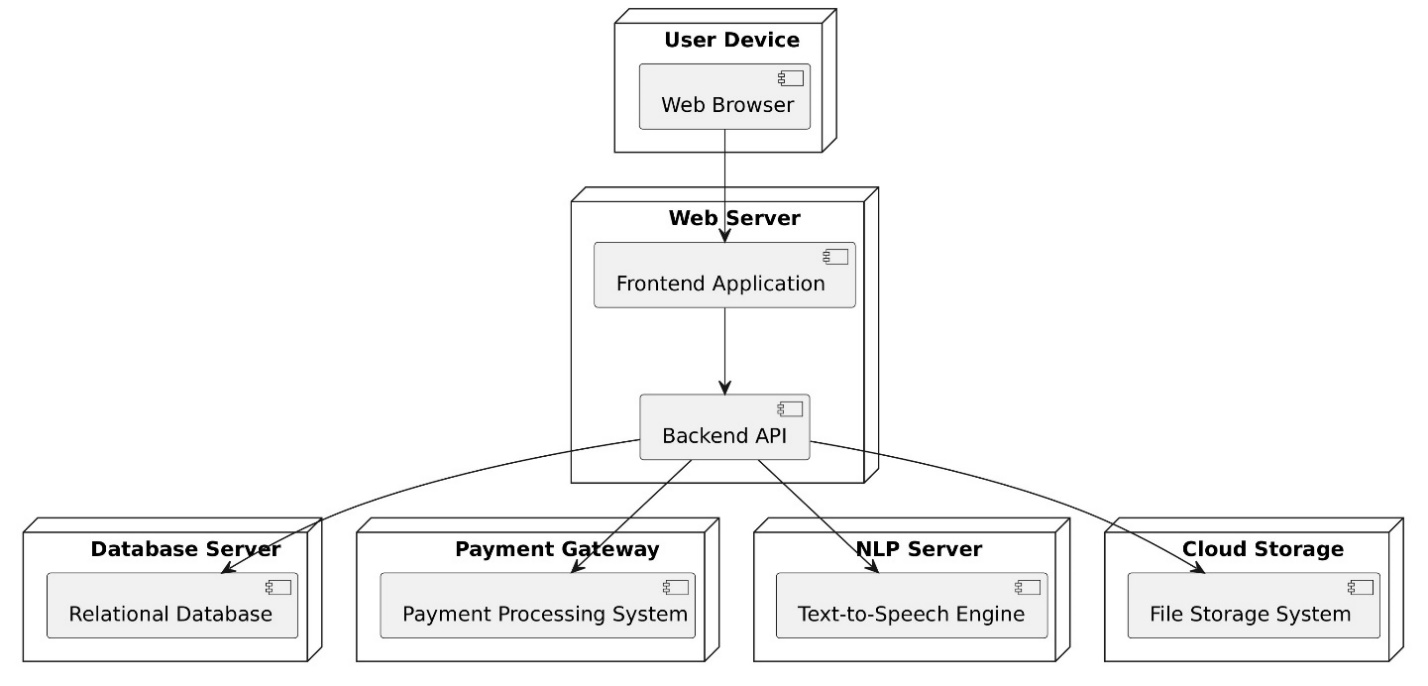


Figure 45 Deployment Diagram

# Chapter 5: Testing

#### Test Case Specifications

**Test Case for Create Profile**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Positive Test Case** | |
| ID |  | TC\_CREATE\_PROFILE\_SUCCESS | |
| Priority |  | High | |
| Description |  | To verify successful creation of a user profile with valid inputs. | |
| Reference |  | FR\_01 (Create Profile) | |
| Users |  | Administrator, Farmers, Consumers. | |
| Pre-requisites |  | A | System is online. |
|  |  | B | User must have active login credentials provided by system administrator. |
|  |  | C | User has internet access. |
| Steps |  | A | Open the "Create Profile" page. |
|  |  | B | Enter valid details (e.g., Name, Email, Password, etc.). |
|  |  | C | Click the "Submit" button. |
|  |  |  |  |
| Input |  | User details: name, email, phone number, password. | |
| Expected result |  | The system successfully creates the profile and displays a confirmation message. | |
| Status |  | Tested, passed. | |

Table 21 Positive Test Case for creating profile

|  |  |  |
| --- | --- | --- |
|  | **Negative Test Case** | |
| ID | TC\_CREATE\_PROFILE\_FAILURE | |
| Priority | High | |
| Description | To verify the profile creation process with missing or invalid inputs. | |
| Reference | FR\_01 (Create Profile) | |
| Users | Administrator, Farmers, Consumers | |
| Pre-requisites | A | System is online. |
|  | B | User has internet access. |
|  |  |  |
| Steps | A | Open the "Create Profile" page. |
|  | B | Enter invalid or incomplete details (e.g., missing Email, Password less than 6 characters). |
|  | C | Click the "Submit" button. |
|  |  |  |
| Input | Incorrect Phone number or password or deactivated credentials. | |
| Expected result | The system prevents profile creation and displays an error message. | |
| Status | Tested, passed. | |

Table 22 Negative Test Case

**Test Case for view profile**

|  |  |  |
| --- | --- | --- |
|  | **Positive Test Case** | |
| ID | TC\_VIEW\_PROFILE\_SUCCESS | |
| Priority | High | |
| Description | To verify that the user can view their profile details after logging in. | |
| Reference | FR\_02 (view profile) | |
| Users | Administrator, Farmers, Consumers | |
| Pre-requisites | A | User must be logged in and have internet access. |
| Steps | A | Log into the system. |
|  | B | Navigate to the "View Profile" section. |
|  | C | Click the "View Profile" button. |
| Input | User ID retrieved from session after successful login. | |
| Expected result | The system successfully displays the user's profile details | |
| Status | Tested, passed. | |

Table 23 Positive Test case for view profile

|  |  |  |
| --- | --- | --- |
|  | **Negative Test Case** | |
| ID | TC\_VIEW\_PROFILE\_FAILURE | |
| Priority | High | |
| Description | To verify that a user cannot view their profile if they are not logged in. | |
| Reference | FR\_02 (view profile) | |
| Users | Administrator, Farmers, Consumers | |
| Pre-requisites | A | System is online. |
|  | B | User must be logged in. |
|  |  |  |
| Steps | A | Open the "View Profile" page without logging in. |
|  |  |  |
| Input | Click on view profile | |
| Expected result | The system should redirect to the login page and show a message like "Please log in to view your profile." | |
| Status | Tested, passed. | |

Table 24 Negative Test case

**Test case for inventory management**

|  |  |  |
| --- | --- | --- |
|  | **Positive Test Case** | |
| ID | TC\_INVENTORY\_UPDATE\_SUCCESS | |
| Priority | High | |
| Description | To verify that the admin can successfully update the inventory of a product | |
| Reference | FR\_14 (Inventory Management) | |
| Users | Admin | |
| Pre-requisites | A | Admin must be logged in and have internet access. |
|  | B | Product must exist in the inventory. |
|  |  |  |
| Steps | A | Log into the system as an admin. |
|  | B | Navigate to the "Inventory Management" section. |
|  | C | Select a product to update. |
|  | D | Update the stock quantity (e.g., increase or decrease inventory). |
| Input | Product ID: 12345 New Stock Quantity: "-50"(invalid) | |
| Expected result | The system prevents the update and shows an error message | |
| Status | Tested, passed. | |

Table 25 Positive Test case for inventory management

|  |  |  |
| --- | --- | --- |
|  | **Negative Test Case** | |
| ID | TC\_INVENTORY\_UPDATE\_FAILURE | |
| Priority | High | |
| Description | To verify that the admin cannot update the inventory with invalid or incomplete details. | |
| Reference | FR\_14 (Inventory Management) | |
| Users | Admin | |
| Pre-requisites | A | Admin must be logged in and have internet access. |
|  | B | Product must exist in the inventory. |
|  |  |  |
| Steps | A | Log into the system as an admin. |
|  | B | Navigate to the "Inventory Management" section. |
|  | C | Select a product to update. |
|  | D | Update the stock quantity (e.g., increase or decrease inventory) |
| Input | Product ID: 12345 New Stock Quantity: "200" | |
| Expected result | The system successfully updates the inventory and displays a confirmation message. | |
| Status | Tested, passed. | |

Table 26 Negative Test case

**Test Case for order processing**

|  |  |  |
| --- | --- | --- |
|  | **Positive Test Case** | |
| ID | TC\_ORDER\_PROCESSING\_SUCCESS | |
| Priority | High | |
| Description | To verify that the admin can successfully process an order. | |
| Reference | FR\_15 (Order Processing) | |
| Users | Admin | |
| Pre-requisites | A | Order must be in a "Pending" state. |
|  | B | Admin must be logged in. |
|  |  |  |
| Steps | A | Log into the system as an admin. |
|  | B | Navigate to the "Order Management" section. |
|  | C | Select an order to process. |
|  | D | Change the order status to "Shipped". |
| Input | Order ID: 78901 New Status: "Shipped" Tracking Number: "XYZ123" | |
| Expected result | The system successfully updates the order status and provides tracking information. | |
| Status | Tested, passed. | |

Table 27 Positive Test case for order processing

|  |  |  |
| --- | --- | --- |
|  | **Negative Test Case** | |
| ID | TC\_ORDER\_PROCESSING\_FAILURE | |
| Priority | High | |
| Description | To verify that the admin cannot process an order with invalid details. | |
| Reference | FR\_15 (Order Processing) | |
| Users | Admin | |
| Pre-requisites | A | Order must be in a "Pending" state. |
|  | B | Admin must be logged in. |
|  |  |  |
| Steps | A | Log into the system as an admin. |
|  | B | Navigate to the "Order Management" section. |
|  | C | Select an order to process. |
|  | D | Change the order status to "Shipped".  Click Save |
| Input | Order ID: 78901 New Status: "Shipped" Tracking Number: "empty" | |
| Expected result | The system prevents the update and shows an error message | |
| Status | Tested, passed. | |

Table 28 negative test case

**Test case for bidding function:**

|  |  |  |
| --- | --- | --- |
|  | **Positive Test Case** | |
| ID | TC\_BIDDING\_SUCCESS | |
| Priority | High | |
| Description | Verify that the user can successfully place a bid. | |
| Reference | FR\_09 (Bidding system) | |
| Users | Administrator, Farmers, Consumers. | |
| Pre-requisites | A | User must be logged in. |
|  | B | User must have internet access. |
|  | C | Product must be listed for bidding. |
| Steps | A | Log into the system. |
|  | B | Navigate to a product listing with bidding enabled. |
|  | C  D | Enter a bid amount higher than the current bid.  Click "Place Bid". |
|  |  |  |
| Input | Product ID: 67890  Bid Amount: "10.00" | |
| Expected result | The system accepts the bid and updates the bid history. | |
| Status | Tested, passed. | |

Table 29 Positive Test case for Bidding function

|  |  |  |
| --- | --- | --- |
|  | **Negative Test Case** | |
| ID | TC\_BIDDING\_SUCCESS | |
| Priority | High | |
| Description | Verify that the user can successfully place a bid. | |
| Reference | FR\_09 | |
| Users | Farmers, Consumers. | |
| Pre-requisites | A | User must be logged in. |
|  | B | User must have internet access. |
|  | C | Product must be listed for bidding. |
| Steps | A | Log into the system. |
|  | B | Navigate to a product listing with bidding enabled. |
|  | C  D | Enter a bid amount higher than the current bid.  Click "Place Bid". |
|  |  |  |
| Input | Product ID: 67890  Bid Amount: "5.00" (current highest bid is "10.00") | |
| Expected result | The system rejects the bid and shows an error message (e.g., "Bid must be higher than the current bid."). | |
| Status | Tested, passed. | |

Table 30 Negative test case

**Test case for add to cart:**

|  |  |  |
| --- | --- | --- |
|  | **Positive Test Case** | |
| ID | TC\_ADD\_TO\_CART\_SUCCESS | |
| Priority | High | |
| Description | Verify that the user can successfully add a product to their shopping cart. | |
| Reference | FR\_010 (Add to cart) | |
| Users | Consumers. | |
| Pre-requisites | A | User must be logged in. |
|  | B | User must have internet access. |
|  | C | Product must be available. |
| Steps | A | Log into the system. |
|  | B | Navigate to a product listing |
|  | C  D | Select the desired quantity.  Click the "Add to Cart" button. |
|  |  |  |
| Input | Product ID: 23456  Quantity: "2" | |
| Expected result | The system successfully adds the product to the cart and displays a confirmation message. | |
| Status | Tested, passed. | |

Table 31 Positive Test case for Add to cart

|  |  |  |
| --- | --- | --- |
|  | **Negative Test Case** | |
| ID | TC\_ADD\_TO\_CART\_FAILUREW | |
| Priority | High | |
| Description | Verify that the system does not allow adding a product to the cart if the desired quantity exceeds available stock. | |
| Reference | FR\_010 (Add to cart) | |
| Users | Consumers. | |
| Pre-requisites | A | User must be logged in. |
|  | B | User must have internet access. |
|  | C | Product must be available but with limited stock (e.g., less than the desired quantity). |
| Steps | A | Log into the system. |
|  | B | Navigate to a product listing |
|  | C  D | Select a quantity greater than available stock.  Click the "Add to Cart" button. |
|  |  |  |
| Input | Product ID: 23456  Quantity: "10" (if available stock is "5") | |
| Expected result | The system rejects the request and displays an error message (e.g., "Requested quantity exceeds available stock."). | |
| Status | Tested, passed. | |

Table 32 Negative test case

**Test case for voice narrator:**

|  |  |  |
| --- | --- | --- |
|  | **Positive Test Case** | |
| ID | TC\_VOICE\_NARRATOR\_SUCCESS | |
| Priority | High | |
| Description | Verify that the voice narrator reads the displayed text on the screen. | |
| Reference | FR\_011 Voice narrator) | |
| Users | Consumers, Farmers | |
| Pre-requisites | A | User must be logged in. |
|  | B | User must have internet access. |
|  | C | Text should be available for narration. |
| Steps | A | Log into the system. |
|  | B | Navigate to a product listing |
|  | C | Navigate to a page with displayed text (e.g., registration page, product listing).  Click the "Add to Cart" button. |
|  | D | Click the "Voice Narrator" button. |
| Input | Text on the page (e.g., "Welcome to Agri Shop") | |
| Expected result | The system reads the text aloud via the voice narrator. | |
| Status | Tested, passed. | |

Table 33 Positive Test case for voice narrator

|  |  |  |
| --- | --- | --- |
|  | **Negative Test Case** | |
| ID | TC\_VOICE\_NARRATOR\_FAILURE | |
| Priority | High | |
| Description | Verify that the system displays an error message when the voice narrator is unable to process the text. | |
| Reference | FR\_011 (Voice narrator) | |
| Users | Consumers, Farmers | |
| Pre-requisites | A | User must be logged in. |
|  | B | User must have internet access. |
|  | C | Text must be unavailable or corrupted. |
| Steps | A | Log into the system. |
|  | B | Navigate to a page with no displayed text or corrupted text. |
|  | C | Click the "Voice Narrator" button. |
| Input | Text on the page: Empty or invalid text (e.g., special characters like "###$$$%%%") | |
| Expected result | The system does not read the text and displays an error message (e.g., "Unable to process text for narration."). | |
| Status | Tested, passed. | |

Table 34 Negative Test case

**Test case for online payment:**

|  |  |  |
| --- | --- | --- |
|  | **Positive Test Case** | |
| ID | TC\_ONLINE\_PAYMENT\_SUCCESS | |
| Priority | High | |
| Description | Verify that the user can successfully make an online payment for the products. | |
| Reference | FR\_012 (Online payment) | |
| Users | Consumers | |
| Pre-requisites | A | User must be logged in. |
|  | B | User must have internet access. |
|  | C  D | Product(s) must be added to the cart.  Invalid payment details (e.g., expired credit card). |
| Steps | A | Log into the system. |
|  | B | Add products to the cart. |
|  | C  D  E  F | Navigate to the checkout page.  Select a payment method  Enter invalid or expired payment details (e.g., expired credit card).  Click the "Pay Now" button. |
| Input | Payment Method: Credit card  Card Number: "4111111111111111"  CVV: "123"  Expiration Date: "01/2020" (expired) | |
| Expected result | The system rejects the payment and displays an error message (e.g., "Payment failed. Please check your card details."). | |
| Status | Tested, passed. | |

Table 35Positive Test case for online payment

|  |  |  |
| --- | --- | --- |
|  | **Negative Test Case** | |
| ID | TC\_ONLINE\_PAYMENT\_FAILURE | |
| Priority | High | |
| Description | Verify that the user cannot make an online payment with invalid or expired payment details. | |
| Reference | FR\_012 (Online payment) | |
| Users | Consumers | |
| Pre-requisites | A | User must be logged in. |
|  | B | User must have internet access. |
|  | C  D | Product(s) must be added to the cart.  A valid payment method (e.g., credit card, digital wallet) is available |
| Steps | A | Log into the system. |
|  | B | Add products to the cart. |
|  | C  D  E  F | Navigate to the checkout page.  Select a valid payment method  Enter payment details (e.g., card number, CVV).  Click the "Pay Now" button. |
| Input | Payment Method: Credit card  Card Number: "4111111111111111"  CVV: "123"  Expiration Date: "12/2025" | |
| Expected result | The system processes the payment successfully and updates the order status to "Paid". | |
| Status | Tested, passed. | |

Table 36 Negative Test case

**Test case for user Registration management:**

|  |  |  |
| --- | --- | --- |
|  | **Positive Test Case** | |
| ID | TC\_USER\_REGISTRATION\_APPROVE\_SUCCESS | |
| Priority | High | |
| Description | Verify that the admin can successfully approve a user registration request. | |
| Reference | FR\_013 (User registration management) | |
| Users | Admin | |
| Pre-requisites | A | User registration request must be pending. |
|  | B | Admin must be logged in. |
| Steps | A | Log into the system as an admin. |
|  | B | Navigate to the "User Registration Management" section. |
|  | C  D | Review the pending registration requests.  Approve a user's registration. |
| Input | User Registration Request: "Alies Doe, aliesdoe@example.com" | |
| Expected result | The system successfully approves the user registration and sends a confirmation email to the user. | |
| Status | Tested, passed. | |

Table 37 Positive Test case for User Registration management

|  |  |  |
| --- | --- | --- |
|  | **Negative Test Case** | |
| ID | TC\_USER\_REGISTRATION\_APPROVE\_FAILURE | |
| Priority | High | |
| Description | Verify that the admin can successfully approve a user registration request. | |
| Reference | FR\_013 (User registration management) | |
| Users | Admin | |
| Pre-requisites | A | User registration request must be pending. |
|  | B | Admin must be logged in. |
| Steps | A | Log into the system as an admin. |
|  | B | Navigate to the "User Registration Management" section. |
|  | C  D | Review the pending registration requests.  Reject a user's registration. |
| Input | User Registration Request: "Allies Doe, alliesdoe@example.com" | |
| Expected result | The system successfully approves the user registration and sends a confirmation email to the user. | |
| Status | Tested, passed. | |

Table 38 Negative Test case

#### Black Box Test Cases

**Create Profile**

|  |  |  |
| --- | --- | --- |
| ID | BB\_TC\_01 | |
| Description | Verify that a user (farmer or consumer) can successfully create a profile with valid information. | |
| Preconditions | User is logged in, and the system is operational. | |
| Input | Valid user profile details (e.g., name, email, password, address, etc.). | |
| Steps |  | 1. Log into the system as a user. 2. Navigate to the "Create Profile" section. 3. Enter all required profile details 4. Click the "Submit" button |
| Expected Output | The profile is successfully created, and the system shows a confirmation message. | |
| Success Criteria | The user profile is created with all entered details and stored correctly in the system. | |

Table 39 BB for Create Profile

**View Profile**

|  |  |  |
| --- | --- | --- |
| ID | BB\_TC\_02 | |
| Description | Verify that a user can view their profile details after logging into the system. | |
| Preconditions | User is logged in, and the profile exists. | |
| Input | Request to view the profile (e.g., by clicking "View Profile" button). | |
| Steps |  | 1. Log into the system as a user. 2. Navigate to the "View Profile" section. 3. Click the "View Profile" button. |
| Expected Output | The user’s profile is displayed with all stored details. | |
| Success Criteria | All details entered in the profile are visible and correct. | |

Table 40 BB for view profile

**Inventory Management**

|  |  |  |
| --- | --- | --- |
| ID | BB\_TC\_03 | |
| Description | Verify that the admin can successfully manage the inventory, including updating stock levels and product details. | |
| Preconditions | Admin is logged in, and products are available in the inventory. | |
| Input | Updated product details (e.g., quantity, price, description). | |
| Steps |  | 1. Log into the system as a user. 2. Navigate to the "Inventory Management" section 3. Select a product to update. 4. Update product details (e.g., stock quantity or price). 5. Click "Save". |
| Expected Output | The product is added to the cart successfully. | |
| Success Criteria | The cart is updated with the correct product and quantity. | |

Table 41BB for inventory management

**Bidding Function**

|  |  |  |
| --- | --- | --- |
| ID | BB\_TC\_04 | |
| Description | Verify that consumers can successfully place bids on available products. | |
| Preconditions | Consumer is logged in, and the product is listed for bidding. | |
| Input | Bid amount for the product. | |
| Steps |  | 1. Log into the system as a consumer. |
|  |  | 1. Browse the listed products to find one available for bidding 2. Place a bid on the selected product. |
| Expected Output | The product is added to the cart successfully. | |
| Success Criteria | The cart is updated with the correct product and quantity. | |

Table 42 BB for Bidding Function

**Add to cart**

|  |  |  |
| --- | --- | --- |
| ID | BB\_TC\_05 | |
| Description | Verify that the user can successfully add products to their shopping cart. | |
| Preconditions | User is logged in, and products are available for purchase. | |
| Input | Product details (e.g., name, quantity) and user action to add the product to the cart. | |
| Steps |  | 1. Log into the system as a consumer. 2. Browse products and select an item. 3. Select quantity and click "Add to Cart" |
| Expected Output | The product is added to the cart successfully. | |
| Success Criteria | The cart is updated with the correct product and quantity. | |

Table 43 BB for Add to Cart

**Voice Narrator**

|  |  |  |
| --- | --- | --- |
| ID | BB\_TC\_06 | |
| Description | Verify that the voice narrator reads the displayed text, such as product descriptions or user instructions. | |
| Preconditions | User is logged in, and the text is available for narration. | |
| Input | Text that is displayed on the screen for narration. | |
| Steps |  | 1. Log into the system. |
|  |  | 1. Navigate to a page with displayed text (e.g., product listing, instructions). 2. Click on the "Voice Narrator" button. |
| Expected Output | The system reads the text aloud through the voice narrator. | |
| Success Criteria | The voice narrator successfully reads the text aloud as specified. | |

Table 44 BB for Voice narrator

**Online Payment**

|  |  |  |
| --- | --- | --- |
| ID | BB\_TC\_07 | |
| Description | Verify that the consumer can successfully make an online payment for the products in the cart. | |
| Preconditions | User is logged in, products are in the cart, and valid payment details are available. | |
| Input | Payment details (e.g., credit card information or digital wallet). | |
| Steps |  | 1. Navigate to the checkout page. 2. Select a payment method. 3. Enter valid payment details (e.g., credit card number, expiration date). 4. Confirm the payment. |
| Expected Output | The payment is successfully processed, and the order status is updated to "Paid." | |
| Success Criteria | The payment is completed without errors, and the order reflects the payment status. | |

Table 45 BB for online payment

**User Registration Management**

|  |  |  |
| --- | --- | --- |
| ID | BB\_TC\_08 | |
| Description | Verify that the admin can manage user registrations, including approving or rejecting user accounts. | |
| Preconditions | Admin is logged in, and user registrations are pending. | |
| Input | User registration details (e.g., name, email, role).  Steps: | |
| Steps |  | Log into the system as an admin.  Navigate to the "User Registration Management" section.  Approve or reject pending registration requests. |
| Expected Output | The user registration status is updated to "Approved" or "Rejected" as per the admin's action. | |
| Success Criteria | The system updates the user registration status correctly and sends notifications (e.g., approval or rejection email) | |

Table 46 BB for urm

* + 1. Equivalence Partitions (EP)

**Create Profile**

|  |  |  |  |
| --- | --- | --- | --- |
| **Variables** | **Valid Classes** | | **Invalid Classes** |
| Username | 1. 2. | Alphanumeric characters only.  Minimum length 5 characters. | 1. Special characters. 2. Empty Field. |
| |  | | --- | | Email |  |  | | --- | |  | |  | valid email format (e.g., user@example.com). | Invalid email format (e.g., user@com,user@example). |
| Password | 1.  2. | Length should be greater than 5 characters.  Alphanumeric, symbols allowed. | 1. Length less than 5 characters. 2. Empty field. |

Table 47 Create profile Valid Classes and Invalid Classes

**View Profile**

|  |  |  |
| --- | --- | --- |
| **Variables** | **Valid Classes** | **Invalid Classes** |
| |  | | --- | | **Login Status** |  |  | | --- | |  | | |  | | --- | | 1. User is logged in and authenticated. |  |  | | --- | |  | | 1. User is not logged in. 2. Invalid login credentials. |
| |  | | --- | | **Profile Type** |  |  | | --- | |  | | 1. Consumer profile. 2. Farmer profile. | 1. Profile does not exist. 2. Profile not linked to the user. |

Table 48 view profile Valid Classes and Invalid Classes

**Create Product listing**

|  |  |  |
| --- | --- | --- |
| **Variables** | **Valid Classes** | **Invalid Classes** |
| |  | | --- | | Product Name |  |  | | --- | |  | | |  | | --- | | 1. Alphanumeric characters only. 2. Minimum 3 characters. |  |  | | --- | |  | | 1. Empty field. 2. Invalid characters (e.g., special symbols). |
| |  | | --- | | Profile Type |  |  | | --- | |  | | 1. Positive numeric value greater than 0. | 1. Negative value. 2. Empty field. 3. Non-numeric characters. |
| |  | | --- | | Quantity |  |  | | --- | |  | | |  | | --- | | 1. Positive integer greater than 0. |  |  | | --- | |  | | 1. Negative value. 2. Empty field. 3. Non-integer values. |

Table 49 Create Product Listing Valid Classes and Invalid Classes

**Bidding Function**

|  |  |  |
| --- | --- | --- |
| **Variables** | **Valid Classes** | **Invalid Classes** |
| |  | | --- | | Bid Amount |  |  | | --- | |  | | |  | | --- | | 1. Numeric value greater than current bid. |  |  | | --- | |  | | 1. Value less than or equal to current bid. 2. Non-numeric input. |
| |  |  |  | | --- | --- | --- | | |  | | --- | | User Role |  |  | | --- | |  | |  |  | | --- | |  | | 1. Positive numeric value greater than 0. | 1. Admin or invalid role. |

Table 50 Bidding Function Valid Classes and Invalid Classes

**Online Payment**

|  |  |  |
| --- | --- | --- |
| **Variables** | **Valid Classes** | **Invalid Classes** |
| |  |  |  | | --- | --- | --- | | |  | | --- | | Payment Method |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | 1. Valid credit card or digital wallet. |  |  | | --- | |  | | 1. Invalid payment method. 2. Empty field. |
| |  |  |  | | --- | --- | --- | | |  | | --- | | Amount |  |  | | --- | |  | |  |  | | --- | |  | | 1. Amount is valid and matches the order total. | 1. Amount less than the total amount. 2. Non-numeric values. |
| Card Information | 1. Valid card number, expiry date, and CVV. | 1. Invalid card details (e.g., incorrect number or CVV). |

Table 51 Online Payment Valid Classes and Invalid Classes

**Add to Cart**

|  |  |  |
| --- | --- | --- |
| **Variables** | **Valid Classes** | **Invalid Classes** |
| |  |  |  | | --- | --- | --- | | |  | | --- | | Product Availability |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 1. Product is in stock. |  |  | | --- | |  | |  |  | | --- | |  | | |  |  |  | | --- | --- | --- | | |  | | --- | | 1. Product is out of stock. |  |  | | --- | |  | |  |  | | --- | |  | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | |  | | --- | | Quantity |  |  | | --- | |  | |  |  | | --- | |  | |  |  | | --- | |  | | 1. Quantity is available and valid. | 1. Negative or zero quantity. 2. Invalid quantity (non-numeric). |

Table 52 Add to cart Valid Classes and Invalid Classes

**User Registration management**

|  |  |  |
| --- | --- | --- |
| **Variables** | **Valid Classes** | **Invalid Classes** |
| |  |  |  | | --- | --- | --- | | |  | | --- | | User Status |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | 1. New user (pending approval). |  |  | | --- | |  | | 1. User is already approved or rejected. |
| |  |  |  | | --- | --- | --- | | |  | | --- | | Admin Action |  |  | | --- | |  | |  |  | | --- | |  | | 1. Admin can approve or reject the registration. | 1. Admin does not have permission to approve/reject. |

Table 53 URM Valid Classes and Invalid Classes

* + 1. Boundary Value Analysis

**Create Profile**

* **Username**:
  + Minimum length: 5 characters.
  + Valid range: 5-15 characters.
  + Invalid range: Less than 5 characters or empty field.
* **Email**:
  + Valid format: e.g., user@example.com.
  + Invalid format: e.g., user@com, missing @ or ..
* **Password**:
  + Minimum length: 6 characters.
  + Valid range: 6-20 characters.
  + Invalid range: Less than 6 characters or empty field.

**View Profile**

* **Login Status**:
  + Valid: User is logged in and authenticated.
  + Invalid: User is not logged in or has invalid credentials.
* **Profile Type**:
  + Valid: Consumer profile or Farmer profile.
  + Invalid: Missing or incorrect profile type.

**Update Profile**

* **Username**:
  + Minimum length: 5 characters.
  + Valid range: 5-15 characters.
  + Invalid range: Less than 5 characters or empty field.
* **Email**:
  + Valid format: e.g., user@example.com.
  + Invalid format: e.g., user@com, missing @ or ..
* **Phone Number**:
  + Valid: Phone number format: 10-15 digits.
  + Invalid: Invalid format (e.g., letters), or empty field.

**Product Listing Creation**

* **Product Name**:
  + Minimum length: 3 characters.
  + Valid range: 3-50 characters.
  + Invalid range: Less than 3 characters or empty field.
* **Price**:
  + Valid range: Greater than 0.
  + Invalid range: Negative value, empty field, non-numeric characters.
* **Quantity**:
  + Valid range: Greater than 0.
  + Invalid range: Zero or negative quantity, non-integer values.

**Bidding Function**

* **Bid Amount**:
  + Valid: Greater than current highest bid.
  + Invalid: Less than or equal to the current bid, empty field.
* **User Role**:
  + Valid: Consumer role can place bids.
  + Invalid: Admin or incorrect role cannot place bids.

**Online Payment**

* **Payment Method**:
  + Valid: Credit card or digital wallet.
  + Invalid: Unsupported payment method.
* **Amount**:
  + Valid: Matches the total order amount.
  + Invalid: Amount is less than the total order amount.
* **Card Information**:
  + Valid: Card number (16 digits), expiry date, CVV.
  + Invalid: Incorrect card details, expired card, incorrect CVV.

**User Registration Management**

* **User Status**:
  + Valid: Pending approval (new user).
  + Invalid: Already approved or rejected.
* **Admin Action**:
  + Valid: Admin can approve or reject registration.
  + Invalid: Admin does not have permission to approve/reject.

**Add to Cart**

* **Product Availability**:
  + Valid: Product is in stock.
  + Invalid: Product is out of stock.
* **Quantity**:
  + Valid: Greater than 0.
  + Invalid: Zero or negative quantity.

**Voice Narrator**

* **Text Availability**:
  + Valid: Text is displayed on the page (e.g., product description, instructions).
  + Invalid: No text available for narration.
* **Audio Output**:
  + Valid: Voice narration reads the text correctly.
  + Invalid: Audio output is unavailable or not supported by the system.
* **Functionality**:
  + Valid: Text is narrated when the user clicks on the "Voice Narrator" button.
  + Invalid: Text is not read aloud or the button does not trigger the narration.
    1. Decision Table Testing

Decision Table is a testing method, which aims to ensure that each one of the possible branch from each decision point is executed at least once and thereby ensuring that all reachable code is executed.

**Create Profile**

**Conditions:**

* C1: Username length is valid (>= 5 characters).
* C2: Email format is valid.
* C3: Password length is valid (>= 6 characters).
* C4: All required fields are filled.

**Actions:**

* A1: Profile is successfully created.
* A2: Error message for invalid input.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **C1** | **C2** | **C3** | **C4** | **A1** | **A2** |
| **Yes** | **Yes** | **Yes** | **Yes** | **Profile created** |  |
| **Yes** | **Yes** | **No** | **Yes** |  | **Password too short** |
| **Yes** | **No** | **Yes** | **Yes** |  | **Invalid email format** |
| **No** | **Yes** | **Yes** | **Yes** |  | **Username too short** |
| **Yes** | **Yes** | **Yes** | **No** |  | Missing field |

### ****2. Product Listing Creation****

**Conditions:**

* C1: Product name length is valid (>= 3 characters).
* C2: Price is valid (greater than 0).
* C3: Quantity is valid (greater than 0).

**Actions:**

* A1: Product listing is successfully created.
* A2: Error message for invalid product details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C1 | C2 | C3 | A1 | A2 |
| Yes | Yes | Yes | Product created |  |
| Yes | No | Yes |  | Invalid price |
| Yes | Yes | No |  | Invalid quantity |
| No | Yes | Yes |  | Invalid product name |

### ****3. Bidding Function****

**Conditions:**

* C1: User is a consumer (valid role).
* C2: Bid amount is higher than the current bid.
* C3: Product is available for bidding.

**Actions:**

* A1: Bid is successfully placed.
* A2: Error message for invalid bid (e.g., bid lower than current bid).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C1 | C2 | C3 | A1 | A2 |
| Yes | Yes | Yes | Bid placed |  |
| Yes | No | Yes |  | Bid lower than current bid |
| Yes | Yes | No |  | Product not available for bid |
| No | Yes | Yes |  | Invalid user role |

### ****4. Online Payment****

**Conditions:**

* C1: Payment method is valid (e.g., credit card or digital wallet).
* C2: Payment amount is correct (matches order total).
* C3: Card details are valid.

**Actions:**

* A1: Payment successfully processed.
* A2: Error message for invalid payment.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C1 | C2 | C3 | A1 | A2 |
| Yes | Yes | Yes | Payment processed |  |
| Yes | No | Yes |  | Incorrect payment amount |
| Yes | Yes | No |  | Invalid card details |
| No | Yes | Yes |  | Unsupported payment method |

### ****5. Add to Cart****

**Conditions:**

* C1: Product is in stock.
* C2: Quantity requested is valid (greater than 0).
* C3: User is logged in.

**Actions:**

* A1: Product added to cart.
* A2: Error message for invalid cart addition.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C1 | C2 | C3 | A1 | A2 |
| Yes | Yes | Yes | Product added |  |
| No | Yes | Yes |  | Product out of stock |
| Yes | No | Yes |  | Invalid quantity |
| Yes | Yes | No |  | User not logged in |

### ****6. Voice Narrator****

**Conditions:**

* C1: Text is available for narration.
* C2: Audio output is supported (device capable of playing audio).
* C3: User clicks on "Voice Narrator" button.

**Actions:**

* A1: Text is read aloud by the voice narrator.
* A2: Error message for issues with audio output.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C1 | C2 | C3 | A1 | A2 |
| Yes | Yes | Yes | Text narrated |  |
| No | Yes | Yes |  | No text available |
| Yes | No | Yes |  | No audio output |
| Yes | Yes | No |  | Button does not trigger |

* + 1. State transition Testing

State Transition testing, a black box testing technique, in which outputs are triggered by changes to the input conditions or changes to 'state' of the system. In other words, tests are designed to execute valid and invalid state transitions.

**User Registration management:**

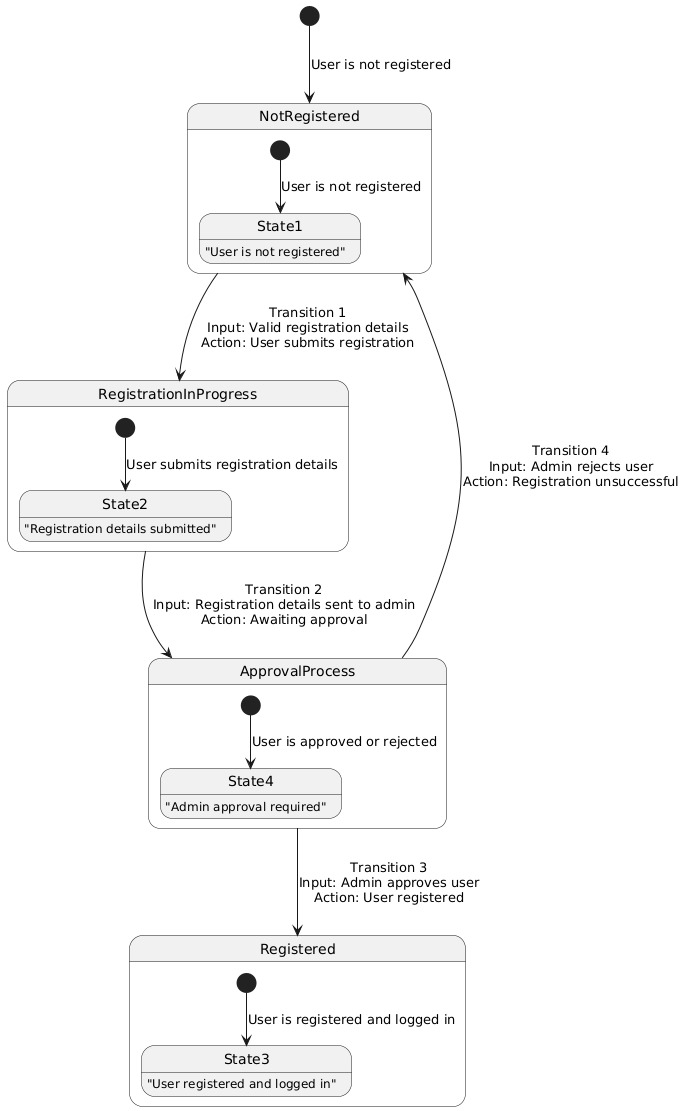


Figure 46 State transition testing diagram of URM

**Add to cart:**

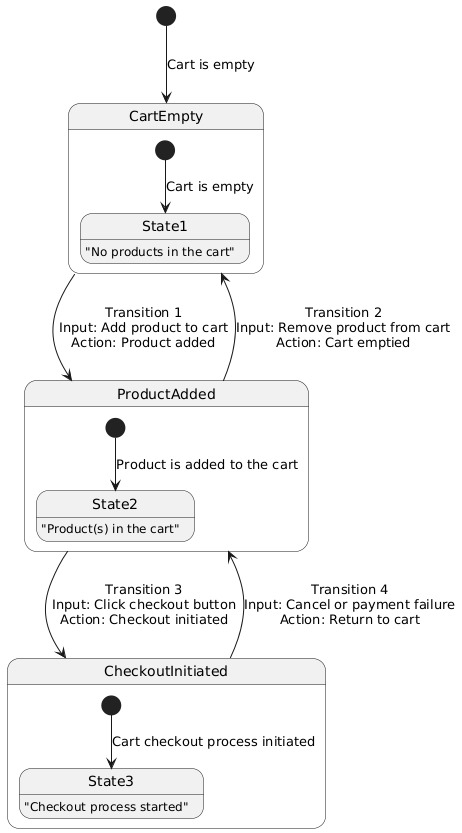


Figure 47state transition testing for Add to cart

**Product List:**

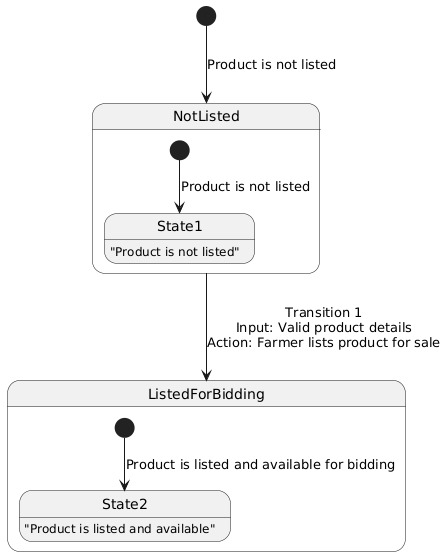


Figure 48 State transition testing for Product List

**Voice narrator:**

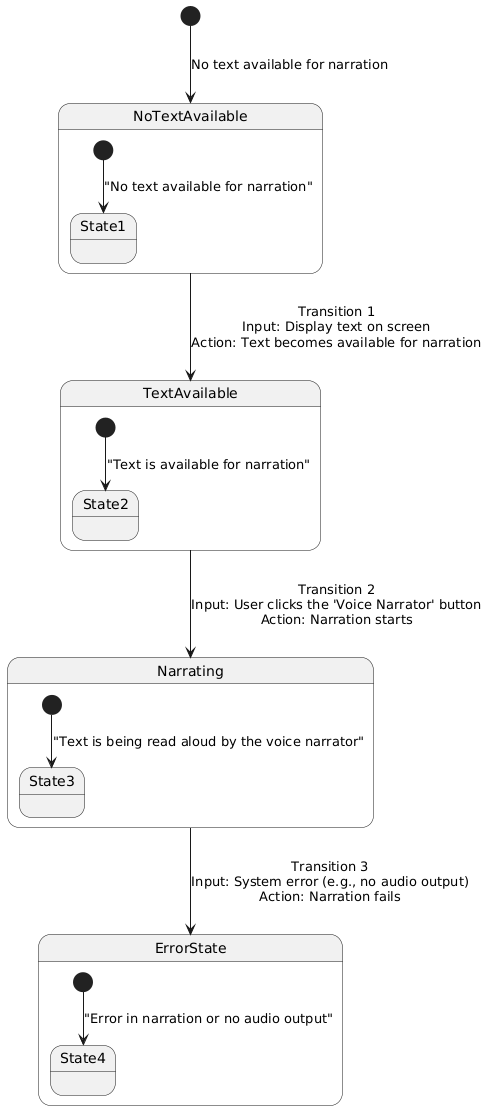


Figure 49 State transition testing for voice narration

**Bidding system:**

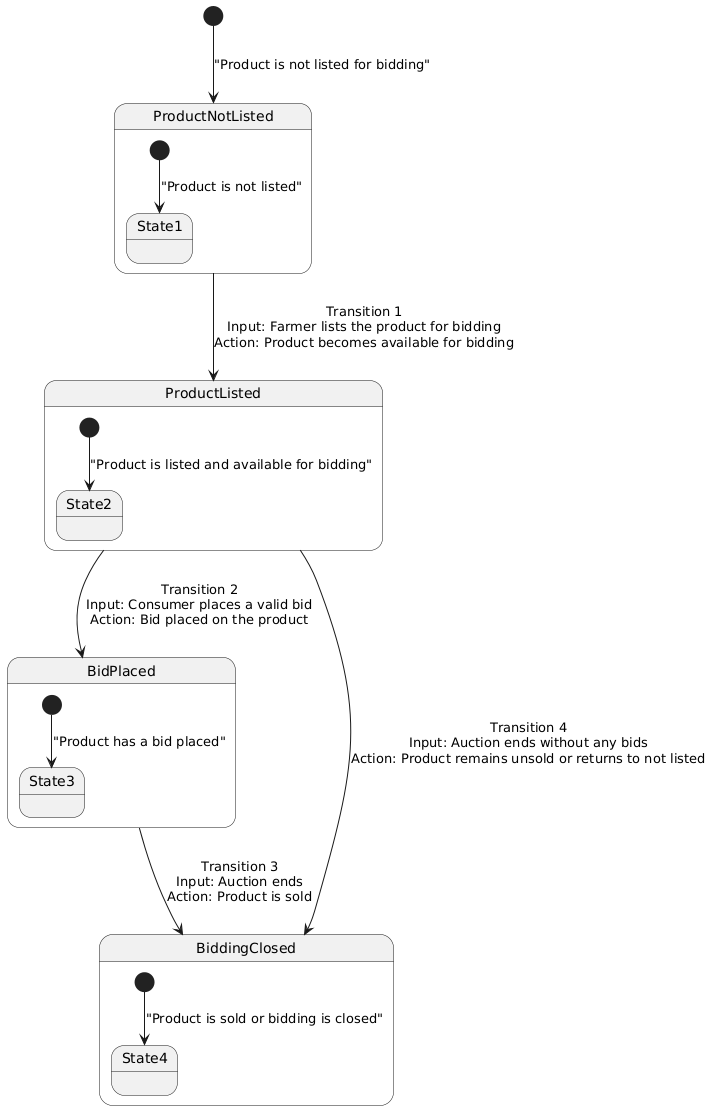


Figure 50 State transition testing for Bidding system

* + 1. Use Case Testing

Use Case Testing is a functional black box testing technique that helps testers to identify test scenarios that exercise the whole system on each transaction basis from start to finish.

**Product listing:**

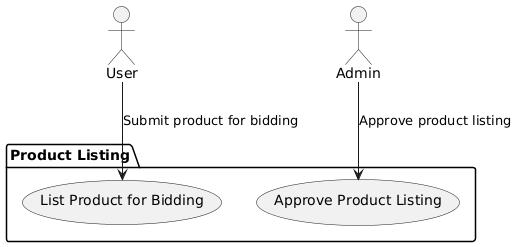


Figure 51Use Case Testing for Product listing

**Bidding system:**

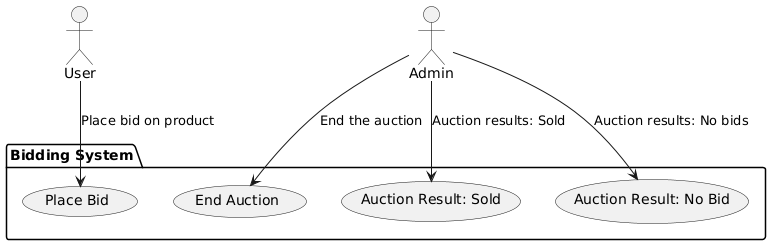


Figure 52 Use Case Testing Bidding system

**Add to cart:**



Figure 53 Use Case Testing Add to Cart

**Voice narrator:**

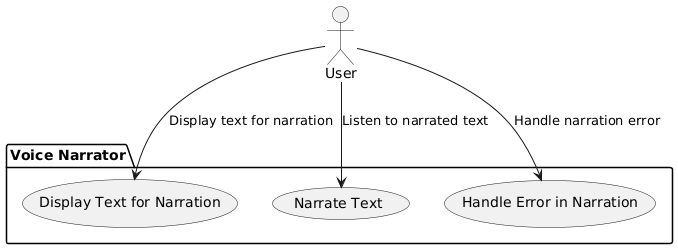


Figure 54 Use Case Testing for Voice narrator

**Payment process:**

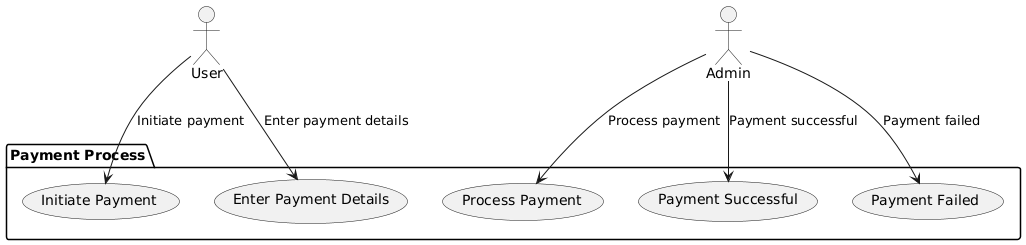


Figure 55 Use Case Testing for Payment process

#### White Box Test Cases

**User Registration**

code

def validate\_registration(username, email, password):

if len(username) < 5:

return "Username too short"

if "@" not in email or "." not in email:

return "Invalid email format"

if len(password) < 6:

return "Password too short"

return "Valid Registration"

**Test Case 1: Valid Registration Flow**

* **Objective**: Verify that the user can successfully register with valid details.
* **Input**: Valid username, email, password.
* **Expected Outcome**: User is added to the database, and a confirmation message is displayed.

**Test Case 2: Invalid Email Format**

* **Objective**: Verify that the system detects an invalid email format.
* **Input**: Invalid email (user.com).
* **Expected Outcome**: Error message indicating "Invalid email format."

**Test Case 3: Empty Password**

* **Objective**: Verify that the system handles an empty password input.
* **Input**: Empty password.
* **Expected Outcome**: Error message indicating "Password is required."

**Product Listing Creation**

Code

def validate\_product\_listing(name, price, quantity):

if len(name) < 3:

return "Product name too short"

if price <= 0:

return "Invalid price"

if quantity <= 0:

return "Invalid quantity"

return "Product Listed"

**Test Case 1: Valid Product Listing**

* **Objective**: Verify that a product is successfully listed with valid details.
* **Input**: Valid product name, price, and quantity.
* **Expected Outcome**: Product is saved in the database and visible in the product listing.

**Test Case 2: Invalid Price (Negative Value)**

* **Objective**: Verify that the system detects an invalid price (negative value).
* **Input**: Price = -50.
* **Expected Outcome**: Error message indicating "Invalid price."

**Test Case 3: Empty Product Name**

* **Objective**: Verify that the system handles an empty product name.
* **Input**: Empty product name.
* **Expected Outcome**: Error message indicating "Product name is required."

**Bidding function**

Code

def place\_bid(current\_bid, user\_bid):

if user\_bid <= current\_bid:

return "Bid must be higher than current bid"

return "Bid placed"

**Test Case 1: Valid Bid Placement**

* **Objective**: Verify that the user can place a valid bid higher than the current bid.
* **Input**: Bid amount = 100 (current bid = 50).
* **Expected Outcome**: The system accepts the bid and updates the product's bid history.

**Test Case 2: Bid Lower Than Current**

* **Objective**: Verify that the system rejects a bid lower than the current highest bid.
* **Input**: Bid amount = 30 (current bid = 50).
* **Expected Outcome**: Error message indicating "Bid must be higher than current bid."

**Test Case 3: Bidding Period Ends**

* **Objective**: Verify that the product status is updated when the bidding period ends.
* **Input**: Bidding period ends with a valid bid placed.
* **Expected Outcome**: Product status updated to "Sold."

**Voice Narrator**

Code

def trigger\_voice\_narrator(text, audio\_device):

if not text:

return "No text available"

if not audio\_device:

return "Audio device not available"

return "Text narrated"

**Test Case 1: Text Narration**

* **Objective**: Verify that the system reads the displayed text aloud when the "Voice Narrator" button is clicked.
* **Input**: Text displayed on the screen and user clicks the "Voice Narrator" button.
* **Expected Outcome**: Text is read aloud correctly.

**Test Case 2: No Text Available for Narration**

* **Objective**: Verify that the system does not attempt to narrate when no text is available.
* **Input**: No text displayed on the screen.
* **Expected Outcome**: Error message indicating "No text available."

**Test Case 3: Audio Output Failure**

* **Objective**: Verify that the system handles failure of audio output (e.g., no sound device).
* **Input**: Click "Voice Narrator" when audio is not supported.
* **Expected Outcome**: Error message indicating "Audio device not available."
  + 1. Cyclometric complexity

**Cyclomatic Complexity** is a metric used to measure the complexity of a program. It is based on the number of linearly independent paths through the program's control flow graph. Cyclomatic complexity helps in determining the number of test cases needed to achieve **complete code coverage** for a particular function.

To calculate cyclomatic complexity, we use the following formula:

V(G)=E−N+2PV(G) = E - N + 2PV(G)=E−N+2P

Where:

**E** = Number of edges in the control flow graph.

**N** = Number of nodes in the control flow graph.

**P** = Number of connected components (for most programs, **P = 1**).

Let's estimate the **Cyclomatic Complexity** for each of the following functionalities based on the program's **control flow**:

**Signup (User Registration)**

**Nodes**:

* + Start
  + Validate inputs (username, email, password)
  + Check if the username is available
  + Check if the email is valid
  + Check password length
  + Save user to the database
  + Display success/failure message

**Edges**:

* + Path between the nodes based on conditions (e.g., valid username, valid email).

**Cyclomatic Complexity Calculation:**

**E** = 7 (number of edges)

**N** = 6 (number of nodes)

**P** = 1 (one connected component)

V(G)=E−N+2P=7−6+2(1) =3

**Cyclomatic Complexity** for **Signup**: **3**

**Login (User Authentication)**

**Nodes**:

* + Start
  + Enter login details (username, password)
  + Validate username
  + Validate password
  + Check if credentials are correct
  + Display success/failure message

**Edges**:

* + Path between conditions based on valid/invalid credentials.

**Cyclomatic Complexity Calculation:**

**E** = 6

**N** = 5

**P** = 1

V(G)=E−N+2P=6−5+2(1) =3

**Cyclomatic Complexity** for **Login**: **3**

**Product Listing**

**Nodes**:

* + Start
  + Enter product details (name, price, quantity)
  + Validate product name
  + Validate price
  + Validate quantity
  + Save product to database
  + Display success/failure message

**Edges**:

* + Path between the conditions based on valid product name, price, and quantity.

**Cyclomatic Complexity Calculation:**

**E** = 7

**N** = 6

**P** = 1

V(G)=E−N+2P=7−6+2(1) =3

**Cyclomatic Complexity** for **Product Listing**: **3**

**Bidding Function**

**Nodes**:

* + Start
  + Enter bid amount
  + Validate bid amount (greater than current bid)
  + Update bid history
  + Display success/failure message

**Edges**:

* + Path between conditions based on valid/invalid bid amount.

**Cyclomatic Complexity Calculation:**

**E** = 5

**N** = 4

**P** = 1

V(G)=E−N+2P=5−4+2(1)=3

**Cyclomatic Complexity** for **Bidding Function**: **3**

**Add to Cart**

**Nodes**:

* + Start
  + Select product
  + Validate product availability
  + Add product to cart
  + Display success/failure message

**Edges**:

* + Path between conditions based on product availability and cart update.

**Cyclomatic Complexity Calculation:**

**E** = 5

**N** = 4

**P** = 1

V(G)=E−N+2P=5−4+2(1) =3

**Cyclomatic Complexity** for **Add to Cart**: **3**

**Voice Narrator**

**Nodes**:

* + Start
  + Detect text availability
  + Validate audio output device
  + Play the narration
  + Display success/failure message

**Edges**:

* + Path between conditions based on text availability and audio device validation.

**Cyclomatic Complexity Calculation:**

**E** = 5

**N** = 4

**P** = 1

V(G)=E−N+2P=5−4+2(1) =3

**Cyclomatic Complexity** for **Voice Narrator**: **3**

**Payment Process**

**Code Structure:**

**Nodes**:

* + Start
  + Validate payment method
  + Validate payment details
  + Process payment
  + Display success/failure message

**Edges**:

* + Path between conditions based on valid/invalid payment method and details.

**Cyclomatic Complexity Calculation:**

**E** = 6

**N** = 5

**P** = 1

V(G)=E−N+2P=6−5+2(1) =3

**Cyclomatic Complexity** for **Payment Process**: **3**

#### Performance testing

Performance testing is a **non-functional testing** technique that is critical for evaluating how your **Agri Shop Web Application** behaves under various loads, stress, and workloads. This type of testing helps to determine the responsiveness, scalability, reliability, and resource usage of the system, ensuring that it can perform well under real-world conditions.

**Goals of Performance Testing for Agri Shop Web Application:**

1. **Responsiveness**: How quickly does the application respond to user interactions (e.g., adding a product to the cart, processing payments, etc.)?
2. **Scalability**: How well does the system handle increasing loads? For example, how does the application handle multiple users browsing, adding items to their cart, or making purchases simultaneously?
3. **Reliability**: Does the system maintain stability and continue functioning correctly under various conditions, even when exposed to high traffic?
4. **Resource Usage**: How efficiently does the application utilize resources like memory, CPU, and network bandwidth when performing operations such as product listing, payments, or product bidding?

#### Stress Testing

**Stress Testing for Agri Shop Web Application**

Stress testing is an important non-functional testing technique that assesses the stability and recovery of the Agri Shop Web Application when subjected to extreme conditions. This testing technique involves subjecting the system to high levels of load beyond its typical capacity to observe how it behaves under stress. It also monitors how the system recovers once the stress is removed.

**Objectives of Stress Testing:**

1. **Ensure System Stability**: Test the system’s ability to function correctly under extreme load conditions (e.g., large numbers of users, high transaction volume).
2. **Assess Recovery**: Monitor the system's ability to recover and resume normal operations after being subjected to stress.
3. **Identify Break Points**: Determine the system's failure points or bottlenecks when it becomes overloaded.
4. **Understand System Behavior Under Stress**: Ensure that the application does not crash or produce incorrect outputs when subjected to high traffic.

#### System Testing

System Testing (ST) is a black-box testing technique that involves evaluating the complete system to ensure its compliance with the specified requirements. Unlike unit testing, which focuses on individual components, system testing tests the entire system from an end-to-end perspective, validating both functional and non-functional aspects of the application. The goal of System Testing is to verify that the Agri Shop Web Application operates as intended across all modules, features, and scenarios, under various conditions. This includes testing user workflows, interactions, and system behavior to confirm that all requirements are met.

**Objectives of System Testing:**

1. **Validate Functional Requirements**: Ensure that all functionalities of the system work as expected.
2. **Verify Non-Functional Requirements**: Test aspects such as performance, scalability, security, and usability.
3. **End-to-End Workflow Testing**: Ensure that the system behaves correctly when different features interact.
4. **Check Compatibility**: Verify that the system works on different devices, browsers, and operating systems.
5. **Verify System Integration**: Ensure that all integrated components (e.g., payment gateways, third-party services) function correctly.

#### 5.7 Regression Testing

Regression testing is a critical aspect of maintaining the stability and reliability of the Agri Shop Web Application. It involves re-running previously executed tests after changes (e.g., bug fixes, new features, or updates) to ensure that these changes have not introduced new defects or broken existing functionality. This type of testing ensures the integrity of the system as it evolves over time.

**Types of Regression Testing for Agri Shop Web Application:**

**Final Regression Testing**:

**Performed when**: A build has remained unchanged for a period of time and is ready to be deployed to customers.

**Objective**: Validate the final version of the application to ensure that all features still work as expected, and no new defects have been introduced.

**Normal Regression Testing**:

**Performed when**: Code changes have been made, typically for defect fixes, bug fixes, or enhancements.

**Objective**: Verify that recent code changes do not break any existing functionality and the application remains stable.

**Selecting Regression Tests for Agri Shop Web Application:**

**Knowledge of the System**:

Understanding the core functionalities of the Agri Shop Web Application and how code changes might affect them is key to selecting appropriate regression tests. This includes understanding how different modules like product listing, cart management, checkout process, and payment processing are integrated.

**Frequent Defects**:

Focus on areas of the application that have historically been prone to defects, such as the payment gateway, user registration, and product search.

**Areas of Frequent Code Changes:**

Identify areas that have undergone frequent code changes, such as the product listing page, checkout process, and user authentication.

**Critical Features**:

Test critical features that are essential for the core functionality of the website, such as:

**Product Listing** (searching, filtering).

**Cart Management** (adding/removing items).

**Checkout** (processing orders).

**Payment Process** (handling payments).

**Regression Testing Steps for Agri Shop Web Application:**

**Select the Tests for Regression:**

**Tests to include**: Choose tests based on the areas of the application that are critical, have frequent defects, or have undergone recent code changes.

**Example**: If new payment methods are integrated, focus on testing the payment processing and checkout flow.

**Choose the Appropriate Tool:**

Use a test automation tool that can handle regression testing efficiently, such as:

**Selenium**: For automated UI testing and end-to-end testing of web pages (e.g., product browsing, adding items to the cart).

JUnit or TestNG: For unit and integration testing (e.g., testing individual functions like login, user registration, or payment processing).

**Jenkins**: To automate the regression testing in the CI/CD pipeline.

**Apache JMeter**: For performance regression testing, especially for load and stress testing during the checkout process.

**3. Automate the Regression Tests:**

Automation is essential for frequent regression testing to save time and ensure consistency.

**Example**: Automate the tests for user login, product listing, cart management, and payment processing to quickly check if recent changes affect existing functionalities.

1. **Verify Applications with Checkpoints:**

Checkpoints are used to verify that critical functionalities work as expected.

**Example**: After testing the product listing.

Verify that products appear with the correct details.

Verify that the search function works as expected.

**5. Manage and Update Regression Tests:**

Update the tests whenever new features are added or changes are made to existing features. This ensures that the tests remain relevant and effective.

**Example**: After a new payment method is added (like PayPal), update the paymentprocessing test suite.

**6. Schedule the Tests:**

Schedule the regression tests to run at regular intervals and after every code change.

For example, after bug fixes, feature additions, or code refactoring.

**7. Integrate with the Builds:**

Integrate regression tests with the build process using tools like Jenkins to run automated tests after every build, ensuring immediate feedback on code quality.

**Example**: Run regression tests after each code commit or release to validate that the build hasn’t broken any existing features.

**8. Analyze the Results:**

After running the regression tests, analyze the results to identify any failures or discrepancies.

**Example**: If a test fails, the regression test team should investigate and communicate the issue to the development team for resolution.

**Regression Testing Example Scenarios for Agri Shop Web Application:**

**Test Suite for User Registration and Login**:

**Tests**:

Verify that new users can successfully register with a valid email, username, and password.

Verify that users can log in using correct credentials.

Verify that an error is shown when the user tries to log in with incorrect credentials.

**Test Suite for Product Listing and Search:**

**Tests**:

Verify that products are listed correctly, with details such as price, description, and stock status.

Verify the search functionality: search by product name, category, and price range.

**Test Suite for Cart Management:**

**Tests**:

Verify that users can add products to their cart and that the cart count is updated.

Verify that users can remove items from the cart and update the quantity of items.

**Test Suite for Payment Processing:**

**Tests**:

Verify that payments can be processed successfully using various payment methods (credit card, PayPal).

Verify that the system correctly handles invalid payments (e.g., expired card or insufficient funds).

Verify that the system displays a confirmation message once payment is successful.

**Regression Testing Example for User Registration:**

**Test Case: Verify User Registration Flow:**

**Objective**: Ensure that the registration flow works correctly after recent changes (e.g., new validation or enhancements).

**Steps**:

Navigate to the "Sign Up" page.

Enter a valid username, email, and password.

Submit the registration form.

Verify that the user is redirected to the login page with a confirmation message.

**Expected Outcome**: User should be successfully registered and redirected to the login page without any errors.

**Regression Testing Example for Product Listing:**

**Test Case: Verify Product Listing and Search:**

**Objective**: Ensure that the product listing page displays products correctly after any changes to the listing functionality.

**Steps**:

Navigate to the product listing page.

Verify that products are listed with correct details (name, description, price).

Use the search bar to search for a product by name, category, and price.

**Expected Outcome**: Products should appear correctly in the listing, and search should return the expected results.

**Tools for Regression Testing:**

**Selenium**: Used for automating the testing of web pages (e.g., login, product browsing, cart management).

**JUnit/TestNG**: Popular testing frameworks for unit testing and integration testing.

**Jenkins**: Continuous integration tool that automates running tests with every build.

**SonarQube**: A static code analysis tool for identifying defects and potential issues in the codebase.

**Apache JMeter**: Performance testing tool for checking load and stress-related regressions.

# Chapter 6: Tools and Techniques

Technologies:

HTML, CSS, TAILWAND CSS, REACT, MONGOS DB, PYTHON

Tools:

VS CODE

# Chapter 7: Summary and Conclusion

**Summary**

The Agri Shop web application addresses critical challenges within Pakistan's agricultural supply chain by providing a platform for farmers to sell their products to consumers. The platform mitigates issues such as limited market access, reliance on intermediaries, and a lack of centralized marketplaces. Incorporating features like text-to-speech technology, it ensures inclusivity for farmers with literacy barriers. Additional features such as secure online payments, bidding systems, and user-generated reviews enhance usability and foster trust among users.

The application employs modern web technologies, including React for the frontend and MongoDB for the backend, ensuring scalability and reliability. The project has undergone rigorous testing to ensure functionality, usability, and performance, reflecting its potential to transform agricultural trading into a more accessible, transparent, and equitable process.

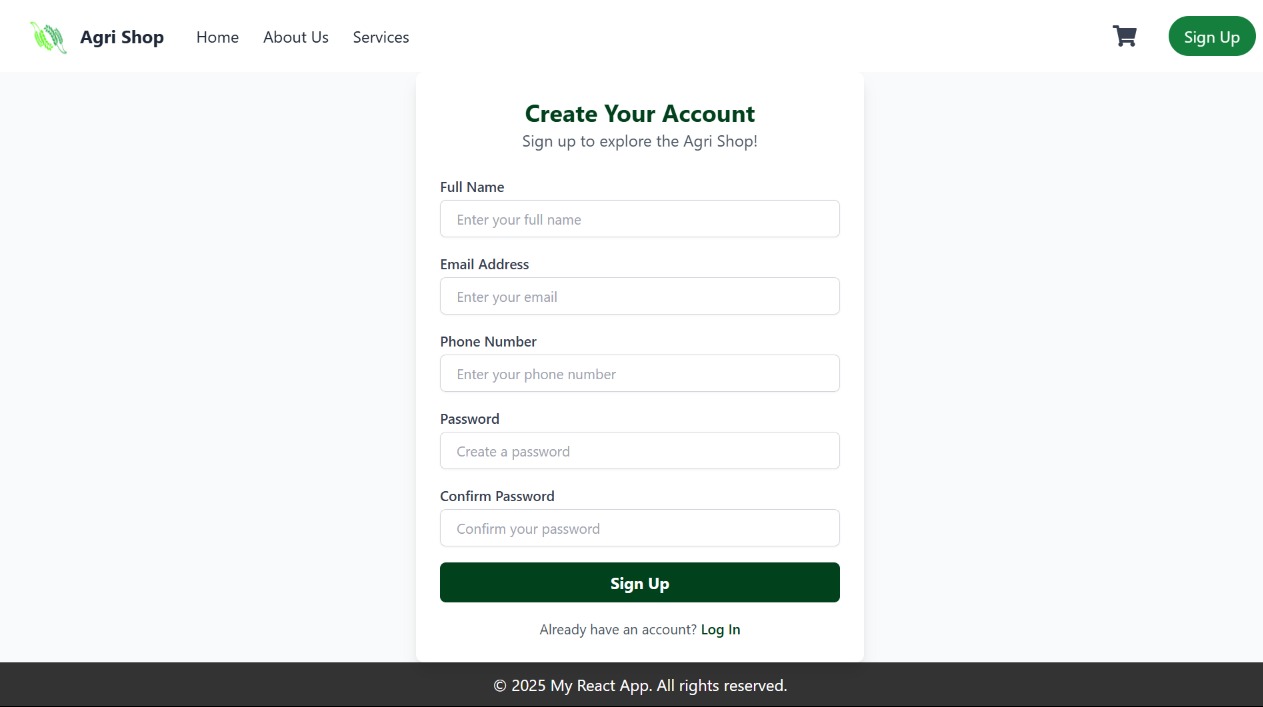
**Conclusion**

The Agri Shop web application successfully achieves its primary goal of streamlining the farm-to-consumer trade process, empowering farmers, and enabling them to access broader markets at fair prices. Its innovative use of accessibility features like real-time text-to-speech ensures inclusivity, while tools like bidding systems and user reviews encourage transparency and competitive pricing.

This project lays a strong foundation for a sustainable and user-focused agricultural e-commerce solution. Future enhancements could include expanded language support, advanced analytics, and integration with logistics services to manage product delivery. By addressing these possibilities, the Agri Shop application could further solidify its role as a pivotal tool for agricultural advancement in Pakistan.

# Chapter 8: User Manual

**Sign Up**

****

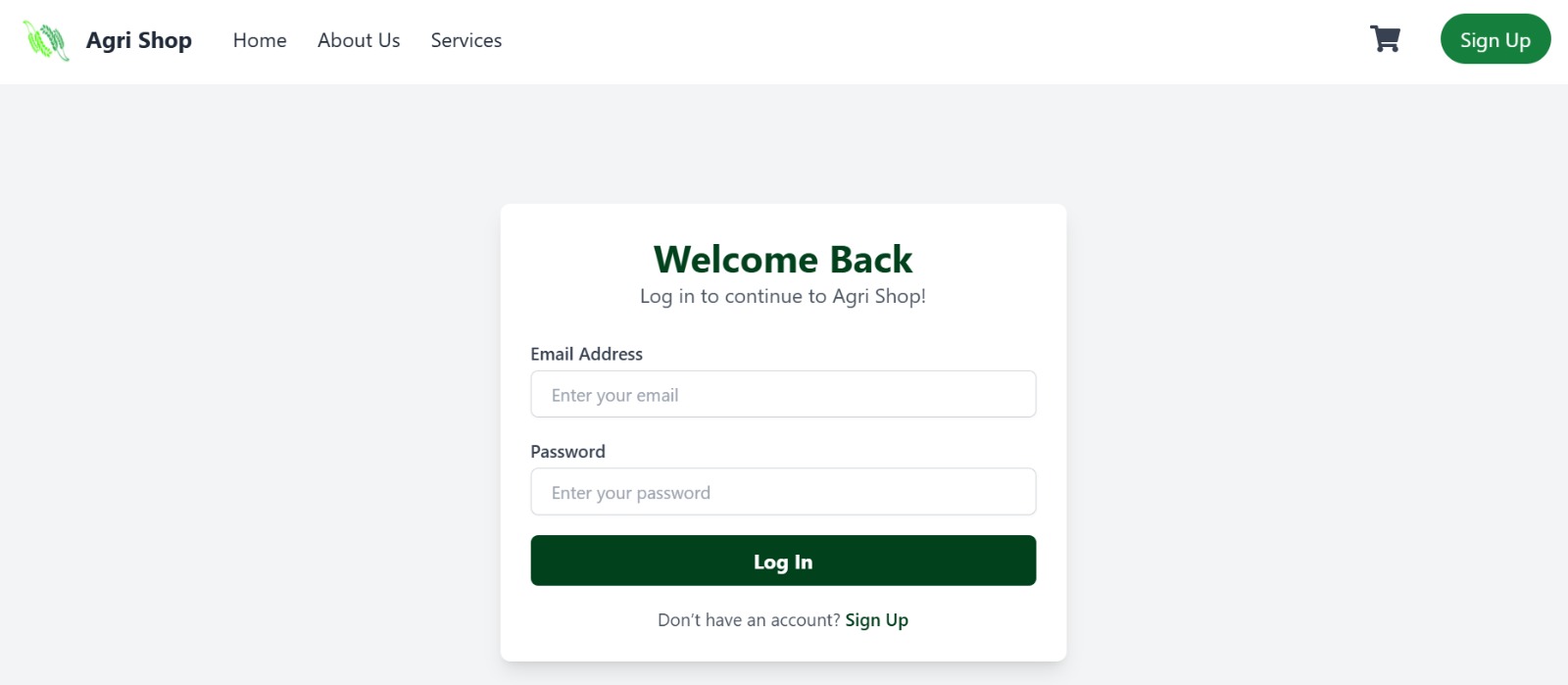
Launch your preferred browser.

1. In the address bar, type the website’s URL and press **Enter**.
2. On the homepage, look for the **Sign-Up** button, located in the top right corner or in the main navigation menu.
3. Add details

* **Full Name**: Enter your first and last name.
* **Email Address**: Type your valid email address.
* **Username**: Choose a unique username for your account.
* **Password**: Create a strong password (usually with a mix of uppercase letters, lowercase letters, numbers, and special characters).
* **Confirm Password**: Re-enter your password to ensure accuracy.
* **Phone Number (optional)**: Some sites may request your phone number for account verification.

1. Click on the **Sign-Up** button to be directed to the registration page.

**Login Page:**

****

Step 1: Navigate to the Login Page

1. Open the website in your browser.
2. If you are not logged in, click the Sign In or Login option (usually in the top-right corner).
3. You will be directed to the login page.

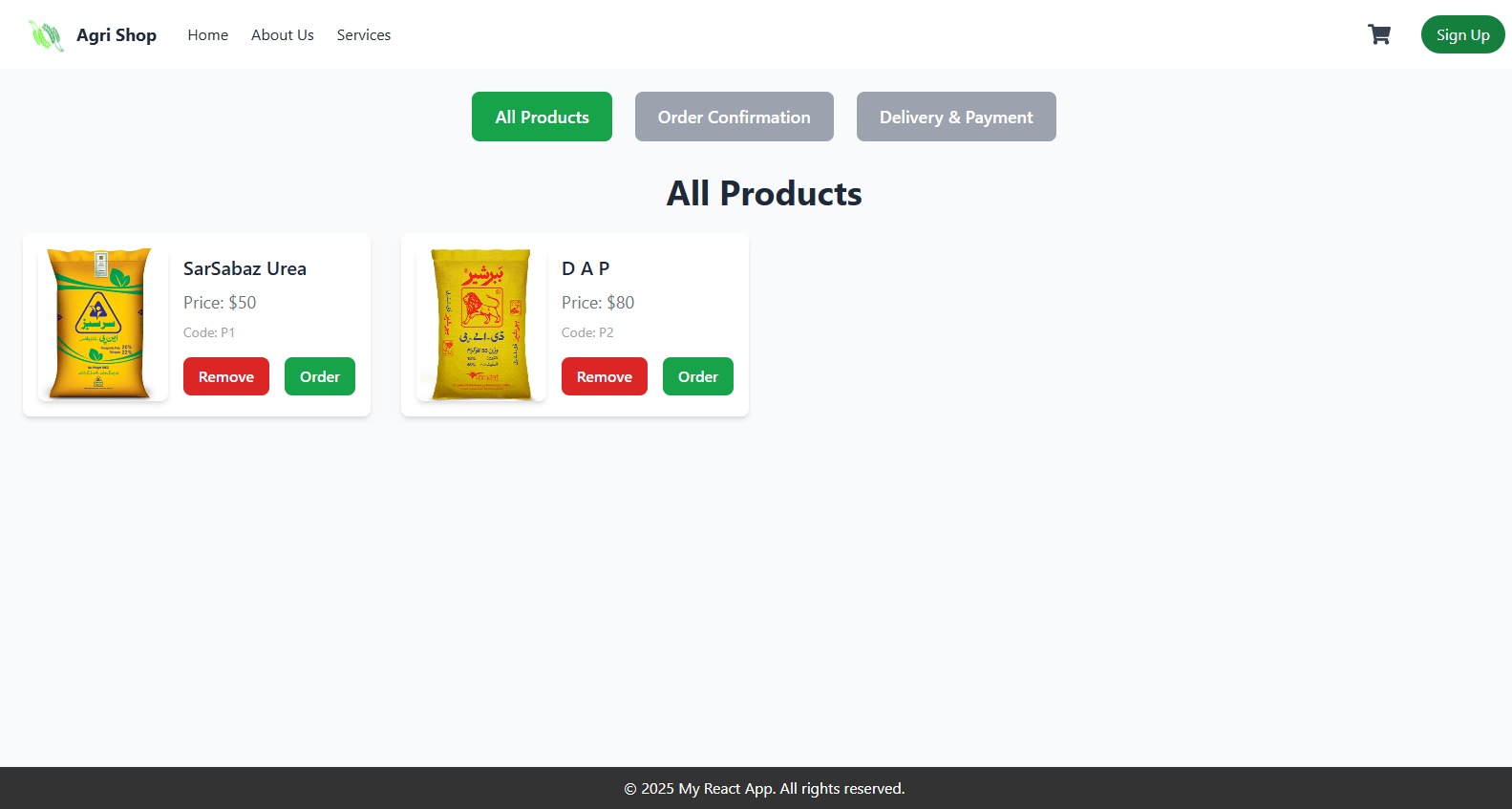
Step 2: Enter Your Credentials

1. In the Email Address field, type the email address associated with your account.
2. In the Password field, enter your password.

Step 3: Log In

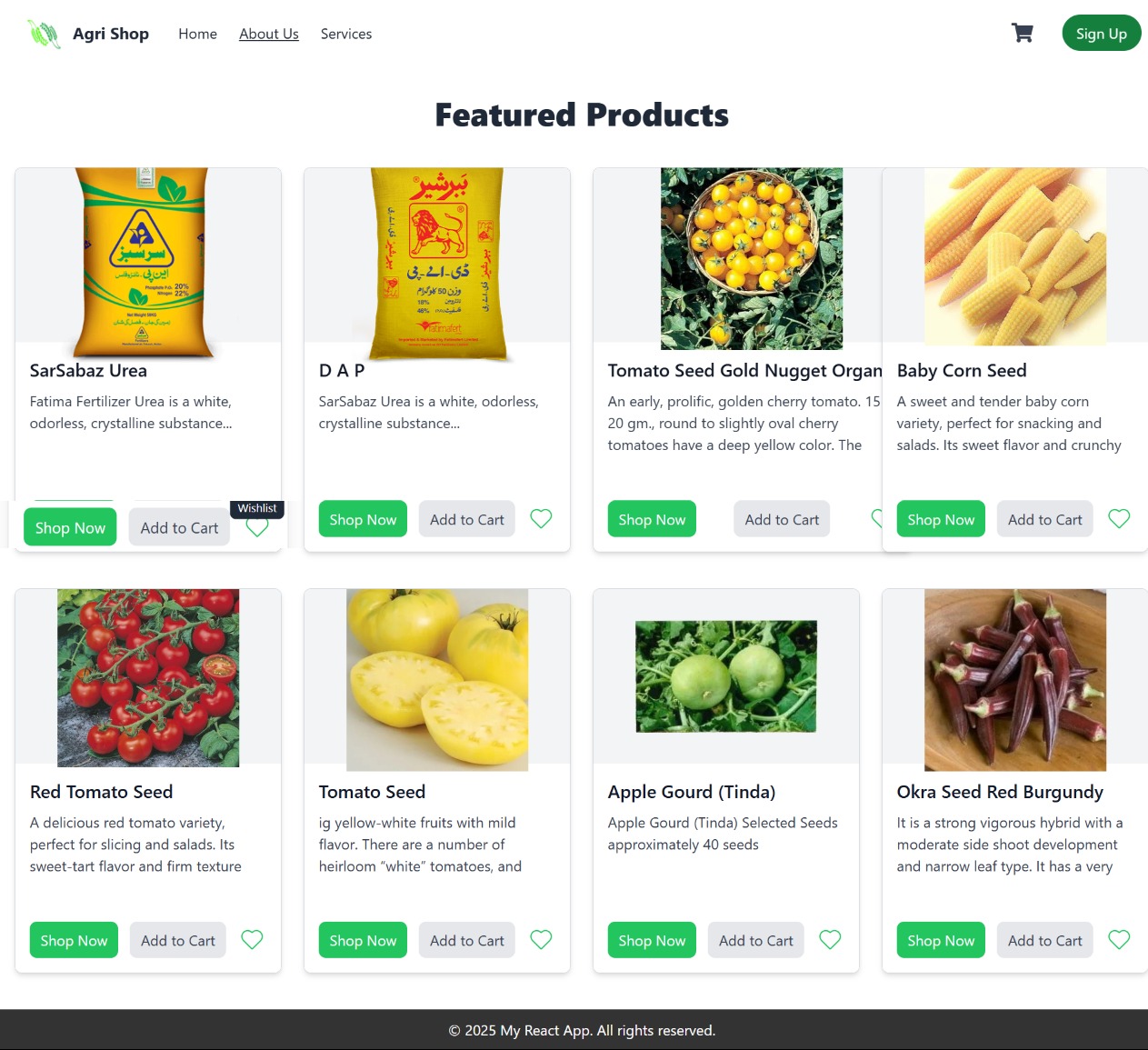
1. Click the Log In button to access your account.

**Product page:**

****

1. Open your preferred web browser.
2. Enter the website’s URL in the address bar and press **Enter**.
3. Browse the **homepage** to find the product you’re looking for.
4. Once you’ve located the product, click on the product image or product name to go to its detailed page.

**Featured products:**

****

Step 1: Navigate to the Product Page

1. Open your preferred browser.
2. In the address bar, type the website’s URL and press Enter.
3. Once the website loads, you will land on the homepage where Featured Products are displayed.

Step 2: Explore the Product Thumbnails

1. The product page displays a series of products with images, names, and brief descriptions.
2. For each product, you can see:

* Product Name: This tells you what the product is, e.g., "SarSabaz Urea" or "Baby Corn Seed."
* Brief Description: Each product has a short description beneath it, explaining its features or benefits.

Step 3: View More Details

1. If you want to know more about a product:

* Click the "Shop Now" button to navigate to the full product page.
* On the product page, you will likely find detailed descriptions, specifications, and additional images.

1. Images: Some products will have multiple images that you can scroll through or zoom in to view more details.

Step 4: Add to Cart or Wishlist

1. Add to Cart: If you want to buy the product, click the "Add to Cart" button. This will add the item to your shopping cart.

* Once the item is added, you can choose to continue shopping or proceed to checkout.

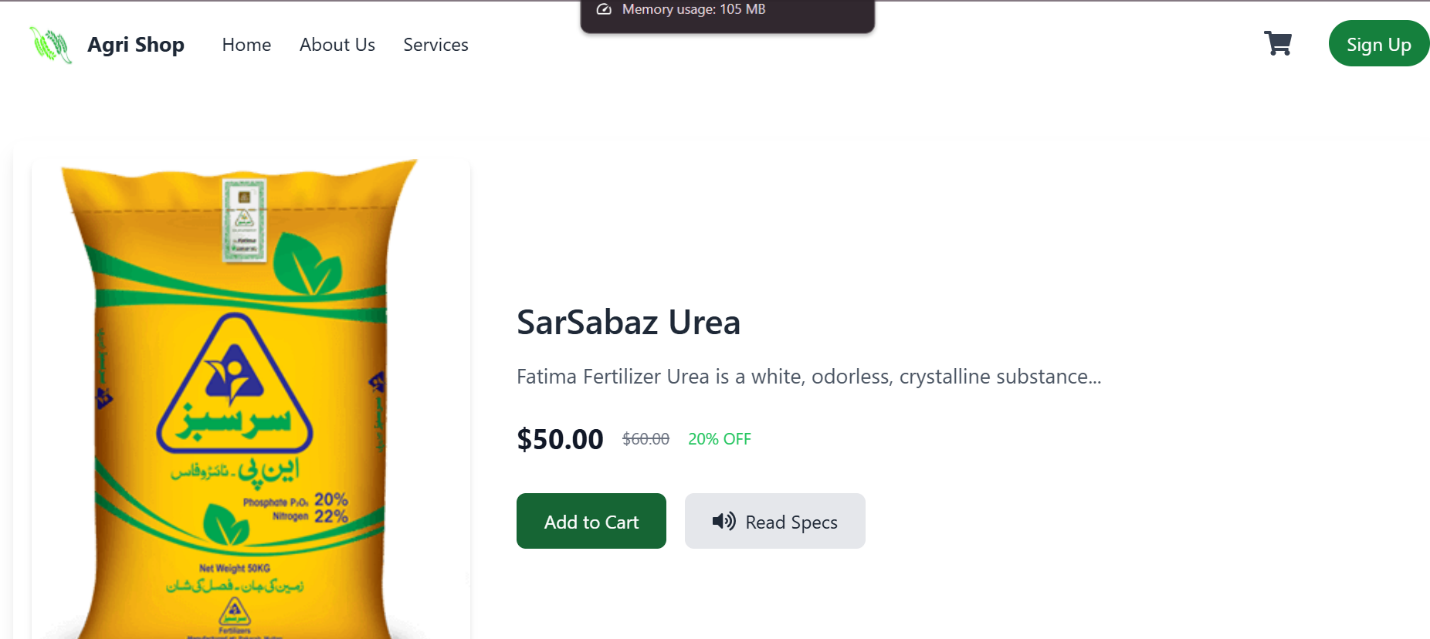
1. Wishlist: If you're interested in the product but not ready to buy, you can add it to your Wishlist by clicking the heart icon. This allows you to save it for later.

Step 5: Continue Browsing or Checkout

1. Once you've added items to your cart, you can:

* Click the cart icon at the top-right corner to view all the items in your cart.
* Review the products, adjust quantities, or remove any unwanted items.
* Once you're ready, proceed to checkout to finalize your purchase.

**Text to Speech:**

****

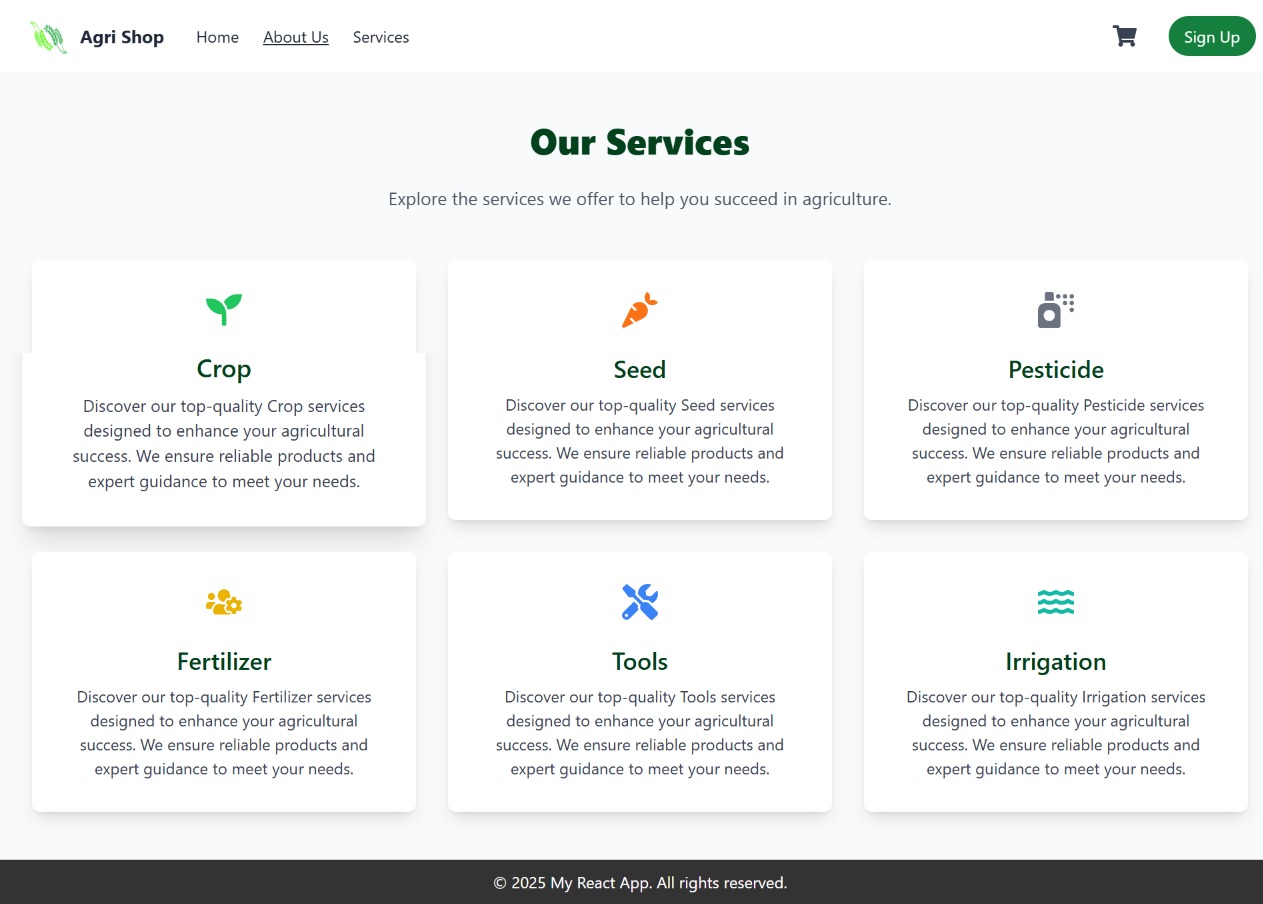
Step 1: Locate the Text-to-Speech Button

1. On the product page, below the product details and price, you will see a **Text-to-Speech button** labeled **"Read Specs"**.

Step 2: Activate the Text-to-Speech Feature

1. Click on the **"Read Specs"** button to activate the text-to-speech feature.
2. The website will read aloud the product specifications and details displayed on the page.

**Services page:**

****

**About US:**

****

# Chapter 9: Lessons Learnt and Future Work

.

1. **Understanding User Needs:** A significant insight was gained into the challenges faced by farmers and consumers in the agricultural sector. This understanding highlighted the importance of accessibility features like text-to-speech and user-friendly interfaces to address literacy and technology adoption barriers.
2. **Importance of Testing and Iteration:** The iterative development process and rigorous testing ensured a reliable application. Functional testing, regression testing, and usability testing were crucial in identifying and resolving bugs while refining features to meet user expectations.
3. **Scalability and Integration:** Developing scalable architecture and integrating third-party services like payment gateways and text-to-speech engines emphasized the importance of seamless integration and robust backend systems.
4. **Collaboration and Teamwork:** The project emphasized the significance of clear communication and effective collaboration among team members and stakeholders to manage project complexities.
5. **Balancing Features and Usability:** A key learning was how to balance implementing advanced features (e.g., bidding systems) with maintaining simplicity and usability for a diverse user base.

**Future Work**

1. **Enhanced Logistics Integration:** Develop partnerships with logistics providers to manage delivery and tracking services, ensuring end-to-end functionality for users.
2. **Advanced Analytics and Insights:** Incorporate features like data analytics dashboards for farmers and consumers, enabling better decision-making through market trends, demand analysis, and pricing insights.
3. **Expanded Accessibility Features:** Improve accessibility by introducing multi-language support, enhanced text-to-speech capabilities, and localized content tailored to different regions and demographics.
4. **Mobile Application Development:** Expand the platform to include a mobile app for Android and iOS, ensuring greater accessibility and convenience for users on the go.
5. **Integration with Government and NGOs:** Collaborate with government bodies and NGOs to provide additional resources like subsidies, crop insurance, and farming tips through the platform.
6. **Global Market Expansion:** Explore avenues for enabling cross-border trade, allowing farmers to access international markets and compete globally.

# References

# 

The references shall be quoted in the following format as provided by IEEE:

**Book**

[Ref number] Author’s initials. Author’s Surname, Book Title, edition (if not first). Place of publication: Publisher, Year.

1. Alex Banks, Functional Web Development with React and Redux, 3rd ed. [Eve Porcello](https://www.google.com.pk/search?sca_esv=e480bd3b9f4b789b&hl=en&sxsrf=ADLYWIIiNv6W5H1RfXe91v1MgPC6058bVQ:1736306055505&q=inauthor:%22Eve+Porcello%22&udm=36) · 2017

**Journal article**

1. [M.H. O'Malley](https://ieeexplore.ieee.org/author/37087958369)., "Text-to-speech conversion technology," Volume: 23, [Issue: 8](https://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=2059&punumber=2), August 1990