

**HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION**  
**FACULTY FOR HIGH QUALITY TRAINING**



**CAPSTONE PROJECT**  
**BUILD AN AGRICULTURAL**  
**PRODUCT TRADING SYSTEM**

**Instructor:**

**Nguyen Thien Bao, PhD**

**Students:**

<b>Pham Viet Anh</b>	-	<b>19110151</b>
<b>Tran Dang Khoa</b>	-	<b>19110145</b>
<b>Nguyen Le Minh Nhut</b>	-	<b>19110100</b>

--- Ho Chi Minh, July 2023 ---

**HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION**  
**FACULTY FOR HIGH QUALITY TRAINING**

----------



**CAPSTONE PROJECT**  
**BUILD AN AGRICULTURAL**  
**PRODUCT TRADING SYSTEM**

**Instructor:**

**Nguyen Thien Bao, PhD**

**Students:**

<b>Pham Viet Anh</b>	-	<b>19110151</b>
<b>Tran Dang Khoa</b>	-	<b>19110145</b>
<b>Nguyen Le Minh Nhut</b>	-	<b>19110100</b>

--- Ho Chi Minh, July 2023 ---

\*\*\*\*\*

## GRADUATION THESIS

**Student Name:** Pham Viet Anh

**Student ID:** 19110151

**Student Name:** Tran Dang Khoa

**Student ID:** 19110145

**Student Name:** Nguyen Le Minh Nhut

**Student ID:** 19110100

1. **Major:** Software Engineering.

2. **Project title:** Build an agricultural product trading system.

3. **Advisor:** Nguyen Thien Bao, PhD.

4. Original data and documents:

- Research and learn Angular, React JS, React Native, Spring Boot.

- Research and learn Tensorflow for detecting fruits.

- Research and learn MySql for database.

5. Content implementation of the topic:

- Learn about Angular, React JS, React Native, Spring Boot, MySql, Material, Tensorlow.

- Building a system of buying and selling agricultural products between merchants and farmers:

- For admin: login/logout, view information about farmers, merchants, products, orders, etc...
- For farmer: login/logout, manage agricultural products, manage profiles, manage orders, etc...
- For merchant: login/logout, find agricultural products, manage profiles, manage orders, etc...

**INSTRUCTOR**

Nguyen Thien Bao

\*\*\*\*\*

## ADVISOR'S EVALUATION SHEET

**Student Name:** Pham Viet Anh

**Student ID:** 19110151

**Student Name:** Tran Dang Khoa

**Student ID:** 19110145

**Student Name:** Nguyen Le Minh Nhut

**Student ID:** 19110100

**Major:** Software Engineering.

**Project title:** Build an agricultural product trading system.

**Advisor:** Dr. Nguyen Thien Bao

### EVALUATION

#### 1. Content of the project:

- a. Theory: Angular, React JS, React Native, Spring Boot, MySQL, Tensorflow.
- b. Experiments:
  - Research renowned websites about agricultural products to decide what features and functions to include in our projects. To determine the project's requirements, identify key points and significant business - build a system to trade agricultural products, meaning a system of buying and selling agricultural products between merchants and farmers.
  - Discover Angular, SpringBoot, React JS and React Native. Effectively using additional libraries, apply it to the project.
  - Build Back-end (with SpringBoot) and front-end (React Native, React JS and Angular). Connect them together libraries to build up the construction of an online can interact between a website and a mobile device with both basic and advanced features.
  - Use Tensorflow to detect fruit with high-resolution image.

c. Links: [https://github.com/nohoopes/KLTN\\_K19\\_CLA\\_Farmhome](https://github.com/nohoopes/KLTN_K19_CLA_Farmhome)

## 2. Strengths:

- The system incorporates seamless file uploading, internet-based hosting, an agricultural detection model, and thorough error testing.
- The interface is improved based on instructor feedback for wider user suitability. Statistical indicators are carefully sourced and consulted with experts. Optimizing simple functions maximizes efficiency and effectiveness.
- Contains the basic functions used for trading agricultural products in the “Merchant” and “Farmer” side.
- There will be some functions that used to support for the trading.
- Collecting data to show the price to the users so they can know about the price in the market right now.
- Notifications will be sent to the user’s phone to make sure they will notice when there is an event to their orders.
- Fruit detection helps users identify different types of fruits, facilitating the easy recording of information through images using a pre-trained model.

## 3. Weaknesses:

- The current system lacks some features such as message, language switching, notification in website, not support for IOS right now in mobile, social media login, etc...
- The application's lack of optimization can lead to slower access speed, potential errors, and misclassification of new agricultural products by the machine learning model used for classification.
- In real-world scenarios, there may be instances where certain features are missing or insufficient, which can present challenges for users.

- Limited advanced permission management features may compromise system security and control by hindering effective user role and access level management.
- The absence of a dedicated reference price management feature for farmers complicates the process of setting competitive prices based on reliable market references.

**4.** Approval for oral defense? (Approved or denied)

.....

**5.** Overall evaluation: (Excellent, Good, Fair, Poor)

.....

**6.** Mark: ..... (in words: .....)

Ho Chi Minh City, July 2023

**ADVISOR**

(*Sign with full name*)

\*\*\*\*\*

## PRE-DEFENSE EVALUATION SHEET

**Student Name:** Pham Viet Anh

**Student ID:** 19110151

**Student Name:** Tran Dang Khoa

**Student ID:** 19110145

**Student Name:** Nguyen Le Minh Nhut

**Student ID:** 19110100

**Major:** Software Engineering.

**Project title:** Build an agricultural product trading system.

**Reviewer:** .....

### EVALUATION

#### 1. Content of the project:

- Theory: Angular, React JS, React Native, Spring Boot, MySQL, Tensorflow.
- Experiments:

- Research renowned websites about agricultural products to decide what features and functions to include in our projects. To determine the project's requirements, identify key points and significant business - build a system to trade agricultural products, meaning a system of buying and selling agricultural products between merchants and farmers.

- Discover Angular, SpringBoot, React JS and React Native. Effectively using additional libraries, apply it to the project.

- Build Back-end (with SpringBoot) and front-end (React Native and Angular). Connect them together libraries to build up the construction of an online can interact between a website and a mobile device with both basic and advanced features.

- Use Tensorflow to detect fruit with high image.

**2. Strengths:**

.....  
.....  
.....

**3. Weaknesses:**

.....  
.....  
.....

**4. Approval for oral defense? (Approved or denied)**

.....

**5. Overall evaluation: (Excellent, Good, Fair, Poor)**

.....

**6. Mark: .... (in words: .....)**

Ho Chi Minh City, July 2023

**REVIEWER**

*(Sign with full name)*

\*\*\*\*\*

## EVALUATION SHEET OF DEFENSE COMMITTEE MEMBER

**Student Name:** Pham Viet Anh

**Student ID:** 19110151

**Student Name:** Tran Dang Khoa

**Student ID:** 19110145

**Student Name:** Nguyen Le Minh Nhut

**Student ID:** 19110100

**Major:** Software Engineering.

**Project title:** Build an agricultural product trading system.

**Name of Defense Committee Member:**

.....  
.....  
.....

### EVALUATION

**1. Content of the project:**

- a. Theory: Angular, React JS, React Native, Spring Boot, MySQL, Tensorflow.
- b. Experiments:
  - Research renowned websites about agricultural products to decide what features and functions to include in our projects. To determine the project's requirements, identify key points and significant business - build a system to trade agricultural products, meaning a system of buying and selling agricultural products between merchants and farmers.

- Discover Angular, SpringBoot, React JS and React Native. Effectively using additional libraries, apply it to the project.
- Build Back-end (with SpringBoot) and front-end (React Native and Angular). Connect them together libraries to build up the construction of an online can interact between a website and a mobile device with both basic and advanced features.
- Use Tensorflow to detect fruit with high image.

**2. Strengths:**

.....  
.....  
.....

**3. Weaknesses:**

.....  
.....  
.....

**4. Approval for oral defense? (Approved or denied)**

.....

**5. Overall evaluation: (Excellent, Good, Fair, Poor)**

.....

**6. Mark: ..... (in words: .....)**

Ho Chi Minh City, July 2023

**COMMITTEE MEMBER**

*(Sign with full name)*

## **ACKNOWLEDGEMENT**

Mr. Nguyen Thien Bao, the instructor of our course, deserves special thanks from our team. Throughout this course, you have constantly assisted us with our difficulties in the study process as well as provided us with excellent feedback and recommendations to help us grow as developers in the future. As a result, you have become our inspiration for this final project, and we would like to express our gratitude for the opportunity to learn from you in this course.

Everyone on our team has various talents, and we have discovered our shortcomings in each other over this final assignment, but we've all managed to face this challenge as a team. As a result, everyone on our team is really grateful for the opportunity to collaborate.

Thank you to the teachers in the Faculty for High Quality Training for always being enthusiastic, dedicated, and dedicated to answering our questions. Besides, we would like to thank our classmates and seniors who did not hesitate to share useful information and experiences to help us improve our thesis.

This final project is the result of our tireless efforts throughout the semester. Because we are still novices, we are aware that our design has numerous shortcomings. We welcome any feedback as well as suggestions for improving our project. We appreciate it. Sincerely!

Ho Chi Minh city, July ...., 2023

Group

Pham Viet Anh

Tran Dang Khoa

Nguyen Le Minh Nhut

## **TABLE OF CONTENTS**

CHAPTER 1: INTRODUCTION.....	1
1.1. Urgency of the topic .....	1
1.2. The purpose of project.....	1
1.3. Object and scope of the study.....	2
1.4. Expected results.....	2
CHAPTER 2. THEORY FUNDAMENTALS .....	3
2.1. Angular .....	3
2.1.1. Definition.....	3
2.1.2. Key features.....	3
2.1.3. Why we use it? .....	4
2.2. React Native .....	5
2.2.1. Definition.....	5
2.2.2. Key features.....	5
2.2.3. Why we use it? .....	6
2.3. ReacJs .....	7
2.3.1. Definition.....	7
2.3.2. Key features.....	8
2.3.3. Why we use it? .....	9
2.4. Spring Boot.....	10
2.4.1. Definition.....	10
2.4.2. Key features.....	10
2.4.3. Why we use it? .....	11
2.5. MySQL .....	12
2.5.1. Definition.....	12

2.5.2. Key features .....	13
2.5.3. Why we use it? .....	14
2.5. Postman .....	15
2.5.1. Definition.....	15
2.5.2. Key features.....	15
2.5.3. Why we use it? .....	16
2.6. Tensorflow.....	17
2.6.1. Definition.....	17
2.6.2. Key features.....	18
2.6.3. Why we use it? .....	19
<b>CHAPTER 3. STATE OF THE ART AND SYSTEM EQUIREMENT MODELING</b>	<b>21</b>
3.1. State of the art.....	21
3.1.2. Domestic .....	21
3.1.2. International .....	21
3.2. System requirements .....	22
3.3. Requirement analysis.....	23
3.3.1. Functional requirements .....	23
3.3.2. Non-functional requirements.....	28
3.4. Bussiness model .....	30
<b>CHAPTER 4. SYSTEM DESIGN .....</b>	<b>31</b>
4.1. List of actors and usecases .....	31
4.1.1. Identify actors and use cases in use case diagrams .....	31
4.1.2. Detailed description.....	32
4.1.3. Usecase summary .....	34

4.1.4. Usecase diagram .....	35
4.2. Usecase specifications .....	39
4.2.1. Description of Login.....	39
4.2.2. Description of Edit Account.....	39
4.2.3. Description of Register.....	40
4.2.4. Description of View List of agricultural products.....	40
4.2.5. Description of View list of agricultural products that farmers own.....	41
4.2.6. Description of Search .....	41
4.2.7. Description of Order.....	42
4.2.8. Description of Cancel the ordered agricultural product .....	43
4.2.9. Description of See the list of ordered agricultural products.....	43
4.2.10. Description of Deal new agricultural products's price.....	44
4.2.11. Description of Decision final deal price.....	44
4.2.12. Description of Manage agricultural product .....	45
4.2.13. Description of Validate agricultural products .....	45
4.3. Sequence diagrams .....	46
4.3.1. Login.....	46
4.3.2. Register.....	47
4.3.3. Create fruit.....	48
4.3.4. Show fruit .....	49
4.3.5. Create order .....	50
4.3.6. Accept order .....	51
4.3.7. Decline order .....	51
4.3.8. Report .....	52

4.3.9. Complete.....	52
4.3.10. Search fruit .....	53
4.4. Database design .....	53
4.4.1. Database table.....	53
4.4.2. Relational diagram.....	60
4.5. User interface.....	61
4.5.1. Mobile for merchant .....	61
4.5.2. Website for farmer.....	80
4.5.3. Website for admin .....	98
<b>CHAPTER 5: IMPLEMENTATION AND TESTING.....</b>	<b>109</b>
5.1. Overall architecture of the system.....	109
5.2. Tensorflow .....	111
5.3. Testing .....	112
<b>CHAPTER 6. CONCLUSION .....</b>	<b>133</b>
6.1. Achieved result.....	133
6.1.1. About knowledge and skills: .....	133
6.1.2. About the topic: .....	134
6.2. Strengths .....	134
6.3. Drawbacks .....	135
6.4. Future developments .....	136

## **LIST OF TABLES**

Table 1. 1. Business functional requirements of merchant department .....	24
Table 1. 2. Business functional requirements of farmer department.....	26
Table 1. 3. System functional requirements .....	28
Table 1. 4. Non-functional requirements.....	29
Table 2. 1. Identify actors and use cases in use case diagrams.....	32
Table 2. 2. Detailed description.....	34
Table 2. 3. Usecase summary .....	35
Table 2. 4. Description of Login .....	39
Table 2. 5. Description of Edit Account.....	40
Table 2. 6. Description of Register.....	40
Table 2. 7. Description of View List of agricultural products .....	41
Table 2. 8. Description of View list of agricultural products that farmers own.....	41
Table 2. 9. Description of Search .....	42
Table 2. 10. Description of Order.....	42
Table 2. 11. Description of Cancel the ordered agricultural product .....	43
Table 2. 12. Description of See the list of ordered agricultural products .....	44
Table 2. 13. Description of Deal new agricultural products's price.....	44
Table 2. 14. Description of Decision final deal price.....	44
Table 2. 15. Description of Manage agricultural product .....	45
Table 2. 16. Description of Validate agricultural products .....	45
Table 2. 17. District database table .....	54
Table 2. 18. Fruit database table.....	54
Table 2. 19. Fruit image database table .....	54

Table 2. 20. History database table .....	55
Table 2. 21. Location database table .....	55
Table 2. 22. News database table .....	56
Table 2. 23. Order database table .....	57
Table 2. 24. Province database table .....	57
Table 2. 25. Role database table.....	57
Table 2. 26. Status product database table .....	57
Table 2. 27. Status user database table.....	58
Table 2. 28. User database table .....	58
Table 2. 29. User role database table.....	59
Table 2. 30. Role database table.....	59
Table 2. 31. List of mobile screens for merchants .....	62
Table 2. 32. Mobile login screen event .....	63
Table 2. 33. Mobile register screen event.....	65
Table 2. 34. Home screen event .....	66
Table 2. 35. Me screen event.....	67
Table 2. 36. Explore screen event .....	68
Table 2. 37. Search screen event .....	68
Table 2. 38. Product detail screen event.....	70
Table 2. 39. Order event .....	71
Table 2. 40. Payment waiting list event .....	71
Table 2. 41. History list event .....	72
Table 2. 42. Order details screen event .....	74
Table 2. 43. Notification screen event.....	74

Table 2. 44. Message screen event .....	75
Table 2. 45. Change password screen event.....	76
Table 2. 46. Update profile screen event.....	78
Table 2. 47. List of message event .....	79
Table 2. 48. Change language screen event .....	80
Table 2. 49. List of website page for farmer .....	81
Table 2. 50. Login page event .....	82
Table 2. 51. Register page event .....	85
Table 2. 52. Home page event.....	86
Table 2. 53. Prices page event.....	87
Table 2. 54. Products page event.....	88
Table 2. 55. Update product event.....	89
Table 2. 56. Delete product event.....	89
Table 2. 57. Add product dialog event .....	91
Table 2. 58. Order page event .....	92
Table 2. 59. Accept order event.....	93
Table 2. 60. Discuss price event.....	93
Table 2. 61. Decline order event .....	94
Table 2. 62. Order with dealing status event .....	95
Table 2. 63. Order with canceled status event.....	96
Table 2. 64. History page event.....	96
Table 2. 65. Profile page event .....	97
Table 2. 66. Change password event.....	98
Table 2. 67. List of website page for admin .....	99

## **LIST OF FIGURES**

Figure 1. 1. Business model .....	30
Figure 2. 1. General usecase diagram.....	35
Figure 2. 2. Usecase for merchant .....	36
Figure 2. 3. Usecase for farmer .....	37
Figure 2. 4. Usecase for admin.....	38
Figure 2. 5. Login sequence diagram .....	46
Figure 2. 6. Register sequence diagram.....	47
Figure 2. 7. Create fruit sequence diagram.....	48
Figure 2. 8. Show fruit sequence diagram.....	49
Figure 2. 9. Create order sequence diagram .....	50
Figure 2. 10. Accept order sequence diagram .....	51
Figure 2. 11. Decline order sequence diagram .....	51
Figure 2. 12. Report sequence diagram .....	52
Figure 2. 13. Complete sequence diagram .....	52
Figure 2. 14. Search fruit sequence diagram .....	53
Figure 2. 15. Database relational diagram.....	60
Figure 2. 16. Mobile login screen.....	62
Figure 2. 17. Mobile register screen 1 .....	63
Figure 2. 18. Mobile register screen 2.....	64
Figure 2. 19. Home screen.....	65
Figure 2. 20. Me screen .....	66
Figure 2. 21. Explore screen.....	67
Figure 2. 22. Search screen.....	68

Figure 2. 23. Product detail screen .....	69
Figure 2. 24. Order bottom sheet.....	70
Figure 2. 25. Payment waiting list.....	71
Figure 2. 26. History list.....	72
Figure 2. 27. Order details screen.....	73
Figure 2. 28. Notification screen .....	74
Figure 2. 29. Message screen .....	75
Figure 2. 30. Change password screen.....	76
Figure 2. 31. Update profile screen 1 .....	77
Figure 2. 32. Update profile screen 2 .....	77
Figure 2. 33. List of message.....	79
Figure 2. 34. Change language screen.....	80
Figure 2. 35. Login page.....	81
Figure 2. 36. Register page .....	82
Figure 2. 37. Home page .....	85
Figure 2. 38. Reference page .....	86
Figure 2. 39. Reference price website .....	87
Figure 2. 40. Products page .....	87
Figure 2. 41. Update product .....	88
Figure 2. 42. Delete product .....	89
Figure 2. 43. Add product dialog.....	90
Figure 2. 44. Order page.....	91
Figure 2. 45. Accept order .....	92
Figure 2. 46. Discuss price .....	93

Figure 2. 47. Decline order.....	94
Figure 2. 48. Order with dealing status .....	95
Figure 2. 49. Order with canceled status .....	95
Figure 2. 50. History page .....	96
Figure 2. 51. Profile page .....	97
Figure 2. 52. Change password .....	98
Figure 2. 53. Login Page .....	100
Figure 2. 54. Statistic Page .....	100
Figure 2. 55. Merchant List Page .....	101
Figure 2. 56. Merchant detail page .....	101
Figure 2. 57. Farmer detail page.....	102
Figure 2. 58. News list page .....	104
Figure 2. 59. Category list page.....	104
Figure 2. 60. Report list page .....	105
Figure 2. 61. Statistic detail .....	106
Figure 3. 1. Architecture of the system .....	109
Figure 3. 2. Login page.....	113
Figure 3. 3. Home page .....	113
Figure 3. 4. Product page.....	114
Figure 3. 5. Add new product .....	114
Figure 3. 6. Detect product .....	115
Figure 3. 7. New product is added.....	115
Figure 3. 8. Edit product.....	116
Figure 3. 9. Delete product .....	117

Figure 3. 10. Login screen .....	118
Figure 3. 11. Explore screen.....	119
Figure 3. 12. Product details screen.....	120
Figure 3. 13. Order .....	121
Figure 3. 14. Choose location.....	122
Figure 3. 15. Go to payment waiting list .....	123
Figure 3. 16. Payment waiting list.....	123
Figure 3. 17. Order page.....	124
Figure 3. 18. Accept order .....	125
Figure 3. 19. Negotiating about the new price .....	126
Figure 3. 20. Decline order.....	126
Figure 3. 21. Discuss new price and amount.....	127
Figure 3. 22. Dealing tab .....	127
Figure 3. 23. Phone received notification .....	128
Figure 3. 24. Order details .....	129
Figure 3. 25. Pending tab.....	130
Figure 3. 26. Accept order .....	130
Figure 3. 27. History page .....	131
Figure 3. 28. History screen .....	131

## **CHAPTER 1: INTRODUCTION**

### **1.1. Urgency of the topic**

Vietnam is an agricultural nation, agricultural products continued to account for a sizable share of the country's exports.

The remaining agricultural products were previously mostly consumed at wholesale marketplaces, with purchasing agents in large centers before coming to the market, in addition to being supplied to supermarkets and import-export businesses. consumers.

The growth of the digital economy has altered how society currently views business. As can be seen, any good or service, including agricultural goods, can be advertised and made "for sale" online. Additionally, consumers want to know whether the agricultural products they buy are of good quality and safe for their health.

Therefore, managers need a more effective communication channel and website/mobile design is the chosen option. This is in addition to enhancing the quality of agricultural products, communicating the brand, and improving the packaging to uphold the quality reputation.

### **1.2. The purpose of project**

The portal allows farmers to first post agricultural items that are only available for a limited time, along with personal and contact information. The merchandiser's mobile application will then show the products, allowing them to select the ones that are best suited to their requirements. The retailer will provide a pricing after determining the appropriate goods and quantity, and will then send the gardener a request along with contact details like their phone number, address, and shipping address. If the farmer accepts, he will confirm and the merchant will receive a successful notification. In contrast, if the farmer rejects the request, it will be declined and a notice of the rejection will be given to the merchant. The farmer may also recommend a different price, and the two parties may speak over the phone to discuss it. Once both parties have consented,

the farmer can use the exchange button to display the form, fill it out with the new price, and submit it back to the merchant for confirmation before resuming the transaction. The merchant will receive updated information to review and validate. The deal is finished if everything goes according to plan and the farmer confirms; if not, it is canceled.

### **1.3. Object and scope of the study**

Two focus groups of participants, including those with and without prior knowledge of technology, were used in the investigation.

Which covers the Angular, React Native, React JS, MySQL database, Spring Boot, as well as a little application about tensorflow among the group of technical knowledge areas. Many useful Javascript libraries are adding to the list of research topics on technology expertise.

The research's main objectives are outlined, and the researcher has a general understanding of its subject matter and is able to apply each knowledge point to the finished product without heavily emphasizing theories, non-applicable.

### **1.4. Expected results**

Topic “BUILD A SYSTEM TO TRADE AGRICULTURAL PRODUCTS” serves following purposes:

- Building a website for farmers can management their products as well as orders.
- Building a mobile application for merchants can find and trade agricultural products with farmers.
- Building a system of buying and selling agricultural products between merchants and farmers with a user-friendly interface, easy to use.

## CHAPTER 2. THEORY FUNDAMENTALS

### 2.1. Angular

#### 2.1.1. Definition

Angular is a popular open-source framework for building web applications. It is developed and maintained by Google. Angular is designed to simplify the process of building dynamic, single-page applications by providing a structured framework and a set of tools and libraries.

One of the key features of Angular is its use of a component-based architecture. It allows developers to build modular and reusable components that encapsulate the application's logic and user interface. These components can be composed and combined to create complex applications.

Angular uses TypeScript, a superset of JavaScript, as its primary language for development. TypeScript adds static typing and other features to JavaScript, making it more robust and maintainable. This helps in catching errors at compile-time and provides better tooling support for code editors.

#### 2.1.2. Key features

*Component-Based Architecture:* Angular follows a component-based architecture, where the application is divided into modular and reusable components. Components encapsulate the logic, data, and user interface of different parts of the application, making it easier to develop, test, and maintain.

*Two-Way Data Binding:* Angular offers two-way data binding, which means changes made to the data in the model are automatically reflected in the view, and vice versa. This simplifies keeping the user interface in sync with the underlying data and reduces the amount of boilerplate code.

*Dependency Injection:* Angular has built-in support for dependency injection, a design pattern that helps manage dependencies between different parts of the application. Dependency injection promotes modularity, reusability, and testability by

making it easier to replace or update components without affecting other parts of the application.

*Templating and Directives:* Angular provides a powerful templating system based on HTML. It allows developers to define the structure and behavior of the user interface using declarative templates. Angular also introduces directives, which are custom HTML tags or attributes that extend the functionality of HTML and enable the creation of dynamic and interactive views.

*TypeScript Language:* Angular is written in TypeScript, a superset of JavaScript that adds static typing and other features to the language. TypeScript improves the maintainability and scalability of Angular applications by catching errors at compile-time and providing better tooling support for code editors.

*Reactive Programming:* Angular integrates well with reactive programming libraries like RxJS. Reactive programming allows developers to handle asynchronous operations, such as HTTP requests or user input, in a more declarative and manageable way. This leads to cleaner and more maintainable code.

### **2.1.3. Why we use it?**

*Enhanced Productivity:* Angular provides a powerful command-line interface (CLI) that automates many common development tasks, such as scaffolding components, generating code, and running tests. This helps developers to be more productive and focus on building features rather than configuring the project.

*Two-Way Data Binding:* Angular offers two-way data binding, which means any changes made to the data in the application's model are automatically reflected in the view, and vice versa. This simplifies the process of keeping the user interface in sync with the underlying data.

*Dependency Injection:* Angular has built-in support for dependency injection, which makes it easier to manage dependencies between different parts of the application. This promotes modularity, testability, and code reusability.

*Powerful Templating System:* Angular provides a declarative templating system that allows developers to define the structure and behavior of the user interface in HTML templates. These templates can contain expressions, directives, and other features to create dynamic and interactive views.

*Robust Community and Ecosystem:* Angular has a large and active community of developers, which means there are plenty of resources, tutorials, and libraries available to support development. Additionally, Angular integrates well with other popular libraries and frameworks, such as RxJS for reactive programming and NgRx for state management.

## 2.2. React Native

### 2.2.1. Definition

React Native is an open-source framework developed by Facebook that allows developers to build mobile applications using JavaScript and React, a popular JavaScript library for building user interfaces. React Native enables the development of native mobile apps for iOS and Android platforms while leveraging the benefits of React's component-based architecture.

### 2.2.2. Key features

*Cross-Platform Development:* React Native enables developers to write code once and deploy it on multiple platforms, such as iOS and Android. This significantly reduces development time and effort compared to building separate native applications for each platform.

*Native-Like Performance:* React Native uses native components and APIs, allowing applications to deliver near-native performance and user experience. It achieves this by rendering components using native UI elements rather than web views.

*Reusable Components:* React Native promotes a component-based architecture, where the application is built by combining reusable UI components. These components can

be shared across different parts of the application or even between different React Native projects, improving code reusability and maintainability.

*Hot Reloading:* React Native supports hot reloading, which allows developers to see the changes they make in the code immediately reflected in the running application. This accelerates the development process and provides a smooth feedback loop for developers.

*Access to Native APIs:* React Native provides a bridge that allows JavaScript code to interact with platform-specific APIs. This means developers can access various native functionalities, such as camera, GPS, storage, and push notifications, by writing JavaScript code.

*Third-Party Library Ecosystem:* React Native benefits from a vast ecosystem of third-party libraries and packages that can be easily integrated into projects. These libraries cover a wide range of functionalities, including UI components, state management, navigation, and more, saving development time and effort.

*Community and Support:* React Native has a large and active community of developers who contribute to its development, share knowledge, and provide support. This community-driven nature ensures the framework's continuous improvement, offers access to resources and tutorials, and helps resolve issues efficiently.

### **2.2.3. Why we use it?**

*Code Reusability:* With React Native, developers can write a single codebase that works across multiple platforms. This reduces the need for separate development teams or rewriting code from scratch, resulting in faster development cycles and reduced maintenance efforts.

*Faster Development:* React Native's hot reloading feature allows developers to see changes instantly, speeding up the development process. Additionally, the availability of pre-built UI components and third-party libraries enables rapid prototyping and development.

*Native-Like Performance:* React Native offers near-native performance by rendering components using native UI elements. This ensures smooth animations, gestures, and overall responsiveness, providing a native-like user experience.

*Cost-Effectiveness:* Building a mobile app using React Native can be more cost-effective compared to developing separate apps for different platforms. It reduces the resources required for development, maintenance, and ongoing updates.

*Developer Efficiency:* React Native leverages JavaScript, a widely adopted programming language, and React's component-based architecture. This allows developers with web development experience to transition into mobile app development more easily and efficiently.

*Platform-Specific Functionality:* React Native provides access to native APIs, allowing developers to access platform-specific features and functionalities. This ensures that the application can leverage the full capabilities of the underlying platform.

*Easy Updates and Maintenance:* With React Native, updates and bug fixes can be applied to the codebase quickly and efficiently. Changes made to the codebase are reflected across all supported platforms, ensuring consistent updates and reducing maintenance overhead.

While React Native offers numerous advantages, it's important to consider that certain platform-specific features or complex UI elements may require additional customization or native code integration. Additionally, apps with high-performance requirements or intensive graphics processing may still benefit

### 2.3. ReactJS

#### 2.3.1. Definition

React JS, also known as React, is an open-source JavaScript library for building user interfaces. It was developed by Facebook and is widely adopted by developers for creating interactive and dynamic web applications. React follows a component-based

architecture, allowing developers to build reusable UI components and efficiently manage the application state.

### 2.3.2. Key features

*Component-Based Architecture:* React organizes the UI into reusable components that encapsulate their own logic and state. Components can be composed together to build complex user interfaces. This component-based approach promotes reusability, maintainability, and modularity in the application development process.

*Virtual DOM:* React utilizes a virtual representation of the Document Object Model (DOM), known as the Virtual DOM. This lightweight abstraction allows React to efficiently update and render the components by comparing the virtual DOM with the actual DOM, minimizing the number of updates and improving performance.

*JSX Syntax:* React uses JSX (JavaScript XML), an extension to JavaScript, which allows developers to write HTML-like syntax directly within JavaScript code. JSX makes it easier to describe the structure and behavior of UI components, enhancing the readability and maintainability of the code.

*Unidirectional Data Flow:* React follows a unidirectional data flow pattern, also known as one-way data binding. This means that the data flows in a single direction, from parent components to child components. This pattern ensures predictable data updates and helps maintain a clear data flow within the application.

*React Hooks:* React Hooks are functions introduced in React 16.8 that allow developers to add state and other React features to functional components. Hooks provide a simpler and more concise way to manage component state and lifecycle events without the need for class components.

*React Router:* React Router is a popular routing library for React applications. It enables developers to handle navigation and routing within a single-page application, allowing for the creation of multiple views and providing a seamless user experience.

### 2.3.3. Why we use it?

*Component Reusability:* React's component-based architecture allows developers to create reusable UI components. These components can be composed and combined in various ways, promoting code reusability and reducing development time.

*Virtual DOM Efficiency:* React's use of the Virtual DOM enables efficient updates and rendering of components. By minimizing direct manipulations to the actual DOM, React optimizes performance and provides a smoother user experience.

*Developer Efficiency:* React's declarative syntax and component reusability improve developer productivity. The use of JSX makes it easier to visualize and create UI components, while React's component architecture facilitates code organization and maintenance.

*Large Community and Ecosystem:* React has a vast and active community of developers, which results in extensive documentation, tutorials, and third-party libraries. This active ecosystem provides access to a wide range of resources, tools, and solutions, accelerating development and problem-solving.

*SEO-Friendly:* React supports server-side rendering (SSR) and can render components on the server before sending the HTML to the client. This improves search engine optimization (SEO) by ensuring that search engines can crawl and index the application's content more effectively.

*Mobile App Development:* React Native, a framework based on React, allows for the development of native mobile applications for iOS and Android platforms. React developers can leverage their existing skills and codebase to build mobile apps, promoting code reuse and reducing development time.

*Stability and Backward Compatibility:* React has a strong commitment to maintaining backward compatibility. This means that React applications built using older versions of the library can often be easily upgraded to newer versions without significant code changes.

React JS is widely used in the industry due to its efficiency, developer-friendly features, and vibrant community support. Its focus on reusable components and efficient rendering makes it a popular choice for building complex and interactive user interfaces in web applications.

### 2.4. Spring Boot

#### 2.4.1. Definition

Spring Boot is an open-source Java framework designed to simplify the development of standalone, production-ready Spring applications. It builds on top of the popular Spring Framework and provides a convention-over-configuration approach, allowing developers to quickly create robust and scalable applications with minimal boilerplate code.

#### 2.4.2. Key features

*Auto-configuration:* Spring Boot provides automatic configuration for various components based on the dependencies present in the project's classpath. It eliminates the need for manual configuration by intelligently configuring defaults and reducing the amount of configuration code required.

*Standalone Applications:* Spring Boot allows you to create standalone applications that can be executed directly without the need for complex deployment procedures or external application servers. It embeds an embedded web server, such as Apache Tomcat or Jetty, within the application, making it self-contained and easy to run.

*Spring Actuator:* Spring Boot Actuator provides powerful features for monitoring and managing applications. It offers built-in endpoints that expose various metrics, health checks, and runtime information about the application, making it easier to monitor and manage the application in production environments.

*Opinionated Defaults:* Spring Boot follows an opinionated approach to provide sensible defaults for application configurations. It minimizes the configuration burden

on developers by making intelligent assumptions about the project's requirements, while still allowing customization when needed.

*Production-Ready Features:* Spring Boot incorporates several features that are essential for developing production-ready applications. It includes support for metrics, logging, security, caching, and database integration, among others. These features help developers focus on business logic rather than dealing with low-level infrastructure concerns.

*Developer Tools:* Spring Boot offers a range of developer tools that enhance the development experience. Features such as automatic application restart, live reloading of static resources, and interactive debugging make the development process faster and more efficient.

*Seamless Integration with Spring Ecosystem:* Spring Boot seamlessly integrates with the wider Spring ecosystem, including popular projects like Spring Data, Spring Security, Spring Cloud, and Spring Integration. It simplifies the integration of these components, enabling developers to leverage their functionalities effortlessly.

### 2.4.3. Why we use it?

*Rapid Application Development:* Spring Boot's convention-over-configuration approach and auto-configuration capabilities significantly reduce the amount of boilerplate code and manual configuration required. This speeds up the development process and allows developers to focus more on writing business logic rather than dealing with infrastructure concerns.

*Increased Productivity:* Spring Boot's opinionated defaults and pre-configured dependencies enable developers to get started quickly with a fully functional application. The out-of-the-box support for commonly used features and integrations eliminates the need for manual setup, saving time and effort.

*Microservices Architecture:* Spring Boot works well in microservices-based architectures. Its lightweight and standalone nature make it suitable for building and deploying independent microservices. The seamless integration with Spring Cloud

provides additional features for service discovery, load balancing, configuration management, and distributed tracing.

*Production-Ready Applications:* Spring Boot incorporates production-ready features such as health checks, metrics, and monitoring capabilities through Spring Actuator. These features enable efficient management and monitoring of applications in production environments, enhancing reliability and scalability.

*Extensive Community and Support:* Spring Boot has a vast and active community of developers, providing access to extensive documentation, tutorials, and community-driven support. The large ecosystem ensures that developers can find solutions to common challenges and leverage community-developed libraries and plugins.

*Industry Adoption:* Spring Boot has gained significant popularity and is widely adopted in the industry. Its reputation for reliability, scalability, and ease of development makes it a preferred choice for building enterprise-grade applications. The availability of skilled developers and resources further contributes to its popularity.

In summary, Spring Boot simplifies the development process, enhances productivity, and provides a solid foundation for building production-ready Java applications. Its opinionated defaults, auto-configuration, and seamless integration with the Spring ecosystem make it a powerful framework for developing a wide range of applications, from simple microservices to complex enterprise systems.

## 2.5. MySQL

### 2.5.1. Definition

MySQL is an open-source relational database management system (RDBMS) that is widely used for storing and managing structured data. It is one of the most popular databases in the world and is known for its ease of use, scalability, and performance. MySQL is used by a wide range of applications, from small personal projects to large enterprise systems.

### 2.5.2. Key features

*Relational Database Management:* MySQL is based on the relational model, which organizes data into tables with rows and columns. It provides a structured and organized approach to storing and retrieving data, allowing for efficient data management.

*Scalability and Performance:* MySQL is designed to handle large amounts of data and high traffic loads. It supports various indexing techniques, query optimization, and caching mechanisms to ensure fast and efficient data retrieval. It also offers options for horizontal scaling through replication and clustering.

*Multi-User and Concurrent Access:* MySQL supports multiple users accessing the database simultaneously, making it suitable for applications with concurrent access requirements. It provides mechanisms for managing concurrent data modifications and enforcing data consistency and integrity.

*Data Security:* MySQL offers robust security features to protect data from unauthorized access and ensure data integrity. It supports user authentication, access control, and encryption of data in transit and at rest. MySQL also provides mechanisms for auditing and logging activities for compliance and troubleshooting purposes.

*Wide Platform Support:* MySQL is available for various platforms, including Windows, macOS, Linux, and other Unix-like operating systems. It is compatible with different programming languages and frameworks, making it versatile and suitable for a wide range of application development scenarios.

*Replication and High Availability:* MySQL supports replication, allowing data to be copied across multiple database servers in real-time. This enables load balancing, data backup, and high availability scenarios where one server can take over in case of failure.

*Community and Ecosystem:* MySQL has a large and active community of developers, providing extensive documentation, tutorials, and forums for support. It also

has a rich ecosystem of third-party tools, libraries, and frameworks that integrate well with MySQL, further enhancing its functionality and ease of use.

### **2.5.3. Why we use it?**

*Ease of Use:* MySQL is known for its ease of installation, configuration, and administration. It provides a user-friendly command-line interface and graphical tools for managing databases and executing queries. Its simplicity makes it suitable for beginners and experienced developers alike.

*Performance and Scalability:* MySQL is optimized for performance and can handle large datasets and high traffic loads efficiently. It supports advanced indexing techniques, caching mechanisms, and query optimization to deliver fast response times, even with complex queries.

*Cost-Effective:* MySQL is an open-source database, which means it is free to use and comes with a vibrant community of contributors and support. This makes it a cost-effective choice for small projects and startups with limited budgets.

*Reliability and Stability:* MySQL has been around for many years and has a proven track record of stability and reliability. It is extensively tested, widely used, and has a large user base, which contributes to its robustness and ensures ongoing development and support.

*Compatibility and Integration:* MySQL is compatible with various programming languages, frameworks, and tools. It provides connectors and drivers for popular programming languages like Java, Python, PHP, and more, allowing easy integration with different application stacks.

*Wide Range of Applications:* MySQL is suitable for a wide range of applications, from small websites to large-scale enterprise systems. It can be used for content management systems, e-commerce platforms, data-driven applications, business intelligence, and analytics solutions, among others.

*Community and Support:* MySQL has a large and active community of developers and users who contribute to its development, share knowledge, and provide

support. The community-driven nature ensures that there are ample resources, documentation, and forums available for assistance.

In summary, MySQL is a powerful and widely adopted relational database management system. Its ease of use, scalability, performance, security features, and extensive community support make it a popular choice for managing and storing structured data in various applications.

### 2.5. Postman

#### 2.5.1. Definition

Postman is a popular collaboration platform and API development tool used by developers and API providers to streamline the process of building, testing, documenting, and sharing APIs. It provides a user-friendly interface for making HTTP requests, inspecting responses, and managing APIs efficiently. Postman simplifies the workflow involved in API development and allows developers to interact with APIs easily.

#### 2.5.2. Key features

*API Request Building:* Postman offers an intuitive interface for constructing and sending HTTP requests. Developers can specify request parameters, headers, authentication methods, and body content in a straightforward manner. It supports various HTTP methods like GET, POST, PUT, DELETE, etc., and allows customization of request details.

*Response Inspection and Validation:* Postman provides a robust response viewer that displays detailed information about API responses. Developers can inspect headers, status codes, response body, and response times. Additionally, it supports the validation of responses against predefined criteria, ensuring the accuracy and conformity of API data.

*Environment and Variables:* Postman allows the creation of environments to manage variables and configurations specific to different development environments.

This feature enables developers to switch seamlessly between environments, making it easier to work with multiple API endpoints or testing scenarios without the need to modify requests manually.

*Automated Testing:* Postman enables the creation of test scripts using JavaScript. Developers can define tests to verify the correctness of API responses, validate specific data elements, or perform complex workflow testing. These tests can be executed automatically, providing quick feedback on the functionality of APIs.

*Collections and Workspaces:* Postman allows the organization of requests into collections, which can be grouped by projects, APIs, or any desired criteria. Collections facilitate better project management and allow easy sharing and collaboration with team members. Workspaces provide a way to organize and share collections across teams, ensuring a centralized repository for API documentation and testing.

*Mock Servers and Documentation:* Postman allows the generation of mock servers based on API specifications or examples. Mock servers simulate the behavior of actual APIs, enabling frontend developers or third-party consumers to interact with the API even before it is fully implemented. Postman also provides tools for generating API documentation, making it easier to communicate API functionality to consumers.

*Integration and Collaboration:* Postman offers various integrations with version control systems like Git, CI/CD pipelines, and popular collaboration platforms. This allows seamless integration into the development workflow and facilitates collaboration among team members, making it easier to share and sync API-related activities.

### **2.5.3. Why we use it?**

*Rapid API Development:* Postman accelerates the API development process by providing an intuitive interface for constructing and testing API requests. It eliminates the need for manual command-line testing or custom scripts, saving time and effort during development.

*Testing and Validation:* Postman simplifies the testing and validation of API responses. It enables developers to define tests and assertions, ensuring the correctness

and integrity of the API. Automated testing and response validation provide confidence in the quality of the API.

*Collaboration and Documentation:* Postman facilitates collaboration among team members by allowing the sharing of collections, environments, and documentation. It provides a centralized platform for documenting APIs, making it easier to communicate API functionality, endpoints, and request/response structures to stakeholders and consumers.

*Mocking and Prototyping:* Postman's mock server feature enables frontend developers or third-party consumers to interact with the API even before it is fully implemented. This promotes early feedback and validation of API design and functionality.

*Integration and Workflow Support:* Postman integrates well with other development tools, version control systems, and collaboration platforms. It seamlessly fits into the development workflow, ensuring smooth integration with existing processes and tools.

*Community and Learning Resources:* Postman has a vibrant community of developers and provides extensive learning resources, tutorials, and documentation. It offers a platform for knowledge sharing, problem-solving, and staying updated with the latest developments in API development.

In summary, Postman is a powerful and user-friendly tool for simplifying API development, testing, documentation, and collaboration. Its rich feature set, including API request building, response inspection, automated testing, mocking, and documentation generation, makes it an essential tool for developers and API providers.

## 2.6. Tensorflow

### 2.6.1. Definition

TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem for building and deploying machine

learning models. TensorFlow is designed to handle both deep learning and traditional machine learning tasks, making it a versatile tool for a wide range of applications.

### 2.6.2. Key features

*Computational Graph:* TensorFlow represents computations as a directed graph, known as a computational graph. Nodes in the graph represent operations, while edges represent data flow. This graph-based approach allows for efficient execution and optimization of complex computations involved in training and deploying machine learning models.

*Flexible Architecture:* TensorFlow offers a flexible architecture that supports a variety of deployment options. It can run on a single CPU, multiple CPUs, or GPUs, as well as distributed computing environments. This flexibility enables scaling and parallelism, allowing for efficient utilization of hardware resources.

*Neural Networks and Deep Learning:* TensorFlow provides extensive support for building and training neural networks, particularly deep learning models. It offers a wide range of pre-built layers, activation functions, and optimization algorithms that simplify the development of deep learning architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

*Automatic Differentiation:* TensorFlow incorporates automatic differentiation, which simplifies the process of computing gradients for training models. It enables the efficient calculation of gradients required for gradient-based optimization algorithms, such as backpropagation, which is widely used in training neural networks.

*Model Serving and Deployment:* TensorFlow facilitates the deployment of machine learning models in production environments. It provides tools and libraries for exporting trained models, creating APIs, and serving predictions. TensorFlow Serving and TensorFlow Lite are examples of deployment options that enable efficient model serving in various contexts, including mobile devices and cloud-based systems.

*High-Level APIs and Abstractions:* TensorFlow offers high-level APIs and abstractions, such as Keras and Estimators, that simplify the development process.

These APIs provide intuitive interfaces for defining models, handling data preprocessing, and training models. They also encapsulate many low-level details, making it easier for beginners to get started with machine learning.

*Community and Ecosystem:* TensorFlow has a vibrant community of developers and researchers. It has a vast ecosystem of libraries, tools, and resources that extend its capabilities. This includes TensorFlow Hub for sharing and discovering pre-trained models, TensorFlow.js for machine learning in the browser, and TensorFlow Extended (TFX) for building end-to-end machine learning pipelines.

### 2.6.3. Why we use it?

*Flexibility and Scalability:* TensorFlow's flexible architecture allows developers to leverage different hardware resources and scale their machine learning workloads efficiently. It can handle projects of varying sizes, from small experiments to large-scale production systems.

*Deep Learning Support:* TensorFlow provides comprehensive support for building and training deep learning models. It offers a wide range of pre-built layers, activation functions, and optimization algorithms that simplify the development of sophisticated neural network architectures.

*Production-Ready Deployment:* TensorFlow offers tools and libraries for exporting, serving, and deploying trained models in production environments. It provides options for integrating models into web applications, mobile devices, and distributed systems, making it easier to operationalize machine learning models.

*Extensive Community and Ecosystem:* TensorFlow has a large and active community of developers and researchers. It benefits from continuous development, improvements, and contributions from the community. This results in a rich ecosystem of libraries, resources, and pre-trained models that enhance the capabilities of TensorFlow.

*Integration with Other Libraries:* TensorFlow integrates well with other popular libraries and frameworks used in the machine learning ecosystem. It can be combined

with libraries such as NumPy, Pandas, and scikit-learn, allowing seamless data manipulation, preprocessing, and post-processing tasks.

*Research and Experimentation:* TensorFlow is widely used in the research community due to its flexibility and extensibility. It provides a platform for experimenting with new machine learning techniques, exploring novel architectures, and pushing the boundaries of what is possible in the field of artificial intelligence.

In summary, TensorFlow is a powerful machine learning framework that provides a comprehensive set of tools and resources for building, training, and deploying machine learning models. Its flexibility, deep learning support, deployment options, extensive community, and integration capabilities make it a popular choice for machine learning practitioners and researchers.

## CHAPTER 3. STATE OF THE ART AND SYSTEM EQUIREMENT MODELING

### 3.1. State of the art

#### 3.1.2. Domestic

In the domestic context, the application of technology in agricultural product consumption is a topic that receives much attention and research. The research aims to provide solutions to enhance the consumption of agricultural products, reduce costs, ensure product quality, and solve the problem of congestion in goods and agricultural products.

Some ongoing research includes:

The study "Đề xuất hoạt động bán hàng trực tuyến trên sàn thương mại điện tử cho cửa hàng sepon 8s". In this research, the research team focuses on understanding the support for businesses to introduce, promote, and successfully apply e-commerce platforms to minimize intermediaries' costs and increase revenue for products, including locally characteristic agricultural products.

The study "Thương mại điện tử cho mặt hàng nông sản gắn với nông hộ". This research identifies the actual needs of farmers, the conditions and accessibility of farmers to e-commerce, and based on that, proposes the construction of a suitable e-commerce website for agricultural products, suitable for the target audience's operation, use, and transactions on the website.

#### 3.1.2. International

Currently, the application of technology in the trade of agricultural products is a widely researched field worldwide, with significant contributions to the development of agricultural economy and addressing issues related to food security. Here are some notable research studies in this field:

The study on applying Blockchain to e-commerce platforms for agricultural products. This research points out the shortcomings and limitations of existing e-

commerce platforms that can be overcome by applying the new electronic information technology, blockchain, to compensate for the deficiencies of the e-commerce platform in this stage.

The study on the development of e-commerce in the agricultural sector in Poland. The research has highlighted the development of e-commerce in Poland through different stages as well as the benefits that e-commerce platforms bring. The study also provides an evaluation of current e-commerce platforms for agricultural products and compares them with the actual situation in Poland.

### **3.2. System requirements**

Store information about farmers, agricultural products, merchants, registration information, ... Merchants who want to order agricultural products such as mango, watermelon, ... must log in to the system by mobile app. Farmers will be able to add and delete their agricultural products on management website. Process for agricultural products ordering:

- Merchants will log into the system.
- Merchants looking for agricultural that match their needs.
- After finding the agricultural products, they will proceed to order on the system.
- The system will send an order to farmer in order to accept or not.

#### **\* Archive:**

The system stores information about:

- Farmer information
- Merchant information
- Agricultural information...
- Optional information: offers, deals, ...

#### **\* Lookup:**

- Merchants:

- + Look up agricultural products infomation.
- + Look up farmers infomation.
- Farmers:
  - + Look up order history.

**\* Calculate:**

- Farmers:
  - + Number of their agricultural products.
  - + Number of merchants ordered.
- Merchants:
  - + Number of orders.
  - + Total order amount.

### 3.3. Requirement analysis

#### 3.3.1. Functional requirements

**\* Business functional requirements:**

*Department: Merchant*

No.	Work	Work type	Regulations/Related Formulas	Related forms	Note
1	View the list of agricultural products	Search			
2	Validate farmer's deal	Store			Accept/Deline the final price of the order

3	Agricultural products search	Search	Search by agricultural products name		
4	Orders	Store			
5	Delete the ordered agricultural product	Store			
6	See the list of ordered agricultural products	Search			See the list of agricultural products that have been ordered

*Table 1. 1. Business functional requirements of merchant department*

***Department: Farmer***

No.	Work	Work type	Regulations/Related Formulas	Related Forms	Note
1	View Agricultural products list	Search			View all the agricultural products that agricultural products has publiced.
2	Add Agricultural products	Store			Agricultural products's manager can add more agricultural products into the web page.
3	Delete Agricultural products	Store			Delete the agricultural products when that agricultural products does not want to public it anymore.
4	Update Agricultural products	Store			Update the agricultural products' information when

				ever there are any changes in the agricultural products' information.
5	View Ordered Agricultural products	Search		View all the agricultural products that have been ordered by the user and some information of that user for the agricultural products to contact.
6	Show The Agricultural products	Store		Show the agricultural products that are available for users to view and order.
7	Validate Ordered Agricultural products	Store		When Merchant order, farmer will accept or decline.
8	Deal orders	Store		Farmer can deal a new price with merchant.

*Table 1. 2. Business functional requirements of farmer department*



**\* System functional requirements:**

No.	Content	Detail description	Note
1	Description of permissions	<ul style="list-style-type: none"> <li>- <b>Merchants:</b> have the right to view agricultural products information, view prices, agricultural products owner, look up agricultural product.</li> <li>- <b>Farmers:</b> Each owner of each agricultural products will have full control over their agricultural products.</li> </ul>	

*Table 1. 3. System functional requirements*

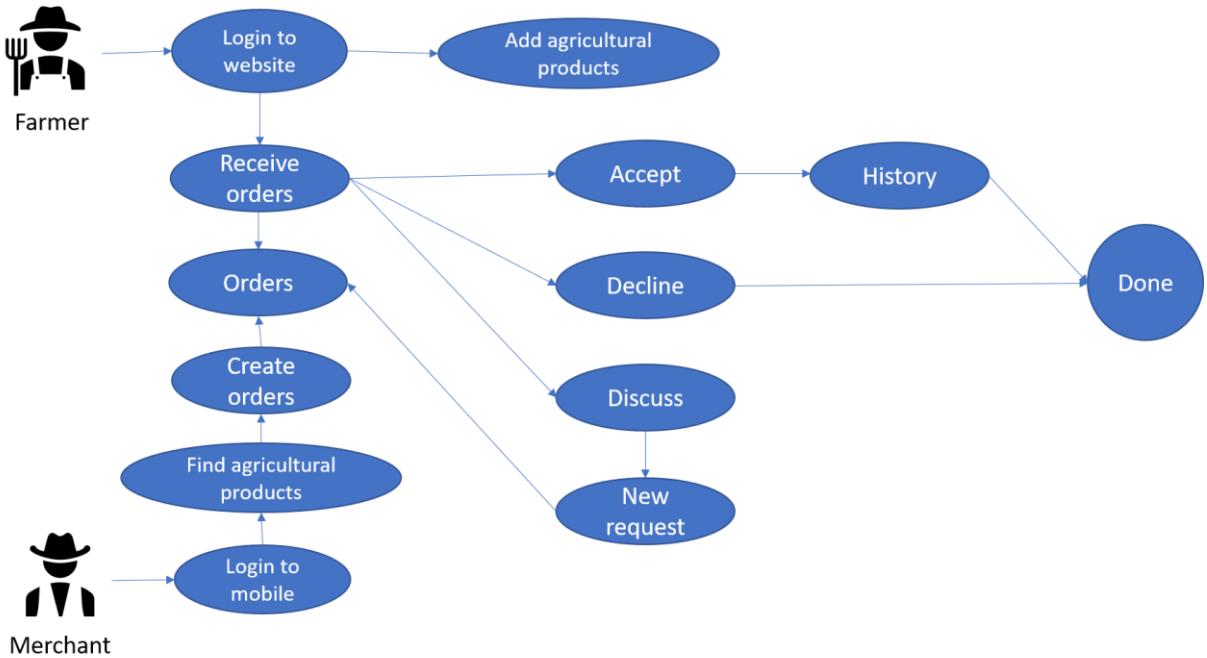
### 3.3.2. Non-functional requirements

No.	Content	Standard	Notes
1	The software can be edited and upgraded to suit the tastes of Merchants and needs of the agricultural products owner.	Evolution	
2	<ul style="list-style-type: none"> <li>- User-friendly.</li> <li>- Easy to manipulate.</li> <li>- The software interacts well with many devices.</li> <li>- The function buttons are arranged not too complicated and easy to see.</li> <li>- Functional monitors interact well with each other, creating convenience for users.</li> </ul>	Convenience	
3	<ul style="list-style-type: none"> <li>- Databases are stored safely and easily retrieved.</li> <li>- The software operates stably, fast access and processing speed.</li> </ul>	Effectiveness	

4	<ul style="list-style-type: none"> <li>- Software placed in different machines can share information with each other.</li> <li>- The software can meet the user's requirements without affecting the operation of other machines.</li> </ul>	Compatibility	
5	<p>The design and functionality of the software can be reused for later developments.</p> <p>The code can be used many times and can be applied to many different programs without having to change the code too much.</p>	Reusability	

*Table 1. 4. Non-functional requirements*

### 3.4. Business model



*Figure 1. 1. Business model*

The portal allows farmers to first post agricultural items that are only available for a limited time, along with personal and contact information. The merchandiser's mobile application will then show the products, allowing them to select the ones that are best suited to their requirements.

The retailer will provide a pricing after determining the appropriate goods and quantity, and will then send the gardener a request along with contact details like their phone number, address, and shipping address. If the farmer accepts, he will click "Confirm," and the merchant will receive a successful notification. In contrast, if the farmer rejects the request, it will be declined and a notice of the rejection will be given to the merchant. The farmer may also recommend a different price, and the two parties may speak over the phone to discuss it. Once both parties have consented, the farmer can use the exchange button to display the form, fill it out with the new price, and submit it back to the merchant for confirmation before resuming the transaction. The merchant will receive updated information to review and validate. The deal is finished if everything goes according to plan and the farmer confirms; if not, it is canceled.

## CHAPTER 4. SYSTEM DESIGN

### 4.1. List of actors and usecases

#### 4.1.1. Identify actors and use cases in use case diagrams

Actors	Usecases
Merchant	<ul style="list-style-type: none"> <li>- Login/Register/Logout</li> <li>- Manage profile</li> <li>- View the list of agricultural products</li> <li>- View the farmers</li> <li>- Search (text/image)</li> <li>- Orders</li> <li>- Decision final deal price (if have)</li> <li>- Delete the ordered agricultural product</li> <li>- See the list of ordered agricultural products</li> <li>- View other information</li> </ul>
Farmer	<ul style="list-style-type: none"> <li>- Login/Register/Logout</li> <li>- Manage profile</li> <li>- Manage agricultural products</li> <li>- Validate agricultural orders (Accept/Discuss/Decline)</li> <li>- View agricultural products' orders</li> <li>- Deal new agricultural products's price</li> <li>- View other information</li> <li>- Detect a product</li> </ul>

Admin	<ul style="list-style-type: none"> <li>- Login</li> <li>- View the statistic of the products and transactions</li> <li>- Filter the data of the statistic by days</li> <li>- View the farmer list and farmer detail</li> <li>- View the merchant list and merchant detail</li> <li>- Ban a user</li> <li>- View the reports of the users</li> <li>- View the news</li> <li>- View the categories of the fruit</li> <li>- Logout</li> </ul>
-------	--

*Table 2. 1. Identify actors and use cases in use case diagrams*

#### 4.1.2. Detailed description

Actors	Usecases
Merchant	<ul style="list-style-type: none"> <li>- Login</li> <li>- Register</li> <li>- Edit profile</li> <li>- View the list of agricultural products</li> <li>- View the farmers</li> <li>- Search agricultural products with text</li> <li>- Search agricultural products with image</li> <li>- Search farmer name</li> <li>- Orders</li> <li>- Decision final deal price (if have)</li> <li>- Delete the ordered agricultural product</li> <li>- See the list of ordered agricultural products</li> </ul>

## CHAPTER 4: SYSTEM DESIGN

---

	<ul style="list-style-type: none"> <li>- See ordered history</li> <li>- Log out</li> </ul>
Farmer	<ul style="list-style-type: none"> <li>- Login</li> <li>- Register</li> <li>- Edit profile</li> <li>- Add agricultural products</li> <li>- Delete agricultural products</li> <li>- Accept agricultural orders</li> <li>- Discuss agricultural orders</li> <li>- Decline agricultural orders</li> <li>- View agricultural products' orders</li> <li>- Deal new agricultural products's price</li> <li>- View agricultural product prices</li> <li>- Detect a product</li> <li>- Logout</li> </ul>
Admin	<ul style="list-style-type: none"> <li>- Login</li> <li>- View the statistic of the products and transactions</li> <li>- Filter the data of the statistic by days</li> <li>- View the farmer list and farmer detail</li> <li>- View the merchant list and merchant detail</li> <li>- Ban a user</li> <li>- View the reports of the users</li> <li>- View the news</li> <li>- View the categories of the fruit</li> <li>- Logout</li> </ul>

*Table 2. 2. Detailed description*

#### 4.1.3. Usecase summary

No	Usecases	Actors
1	Login	Merchant, Farmer, Admin
2	Edit account	Merchant, Farmer
3	Register	Merchant, Farmer
4	View list of agricultural products	Merchant, Farmer, Admin
5	View list of agricultural products that farmers own	Merchant, Farmer, Admin
6	Search	Merchant
7	Order	Merchant
8	Cancel the ordered agricultural product	Merchant
9	See the list of ordered agricultural products	Merchant, Farmer, Admin
10	Deal new agricultural products's price	Farmer
11	Decision final deal price	Merchant
12	Manage agricultural product	Farmer
13	Validate agricultural orders (Accept/Discuss/Decline)	Farmer

14	Detect agricultural product	Merchant, Farmer
15	Statistic	Admin

Table 2. 3. Usecase summary

#### 4.1.4. Usecase diagram

##### \*General usecase diagram

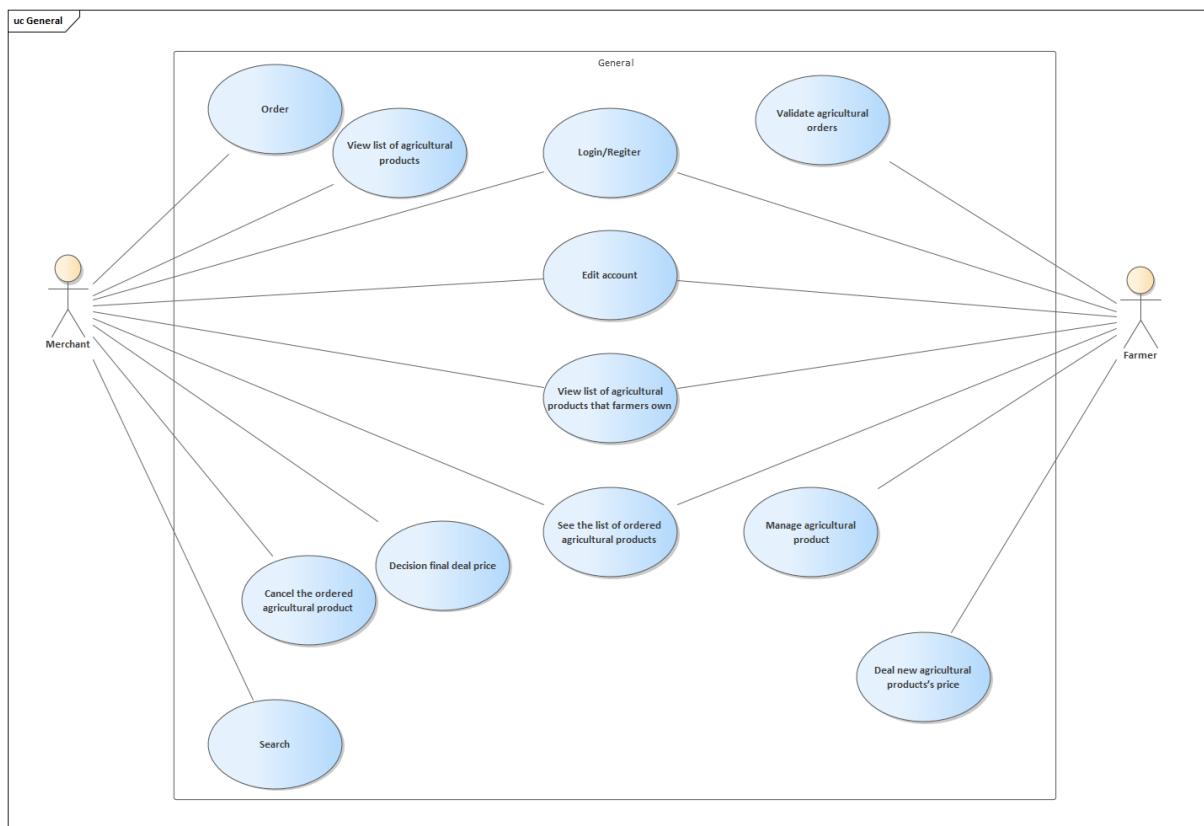


Figure 2. 1. General usecase diagram

**\*Usecase for merchant**

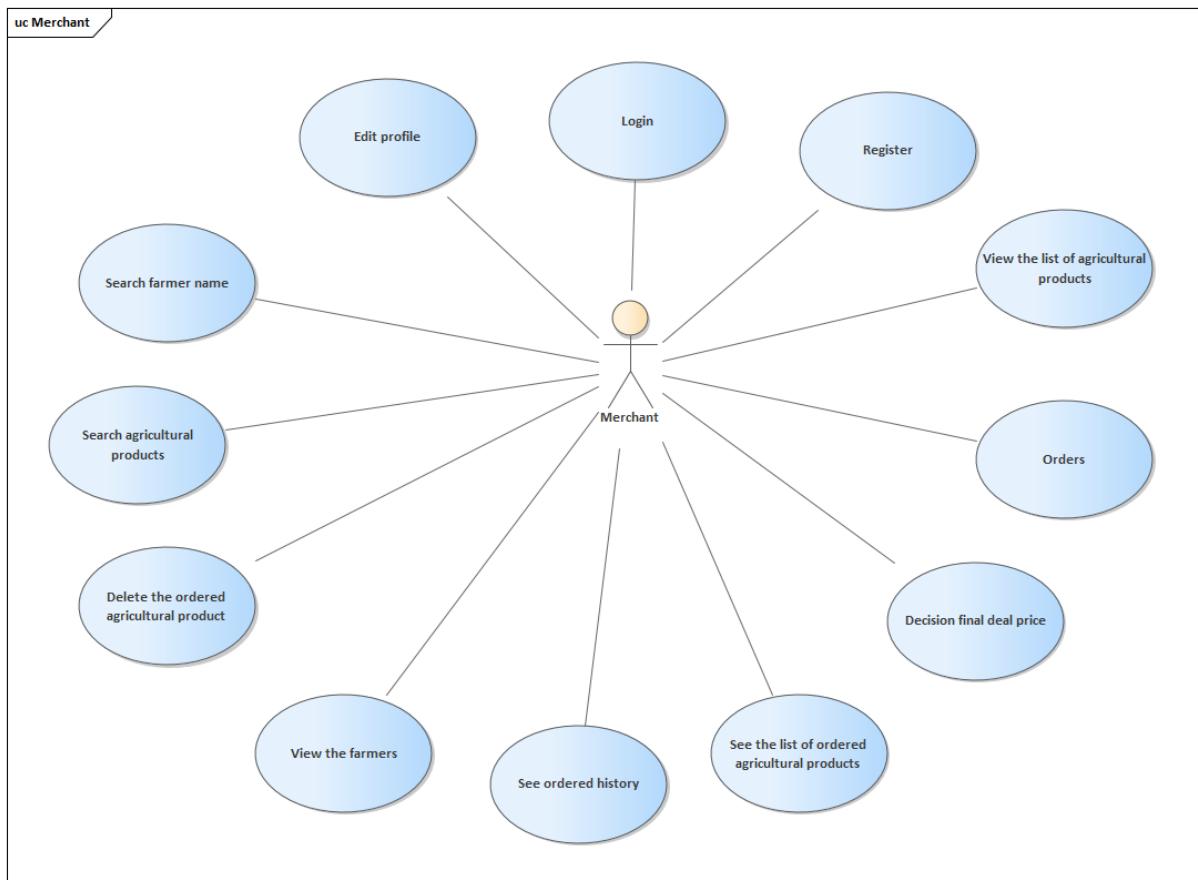


Figure 2. 2. Usecase for merchant

**\*Usecase for farmer**

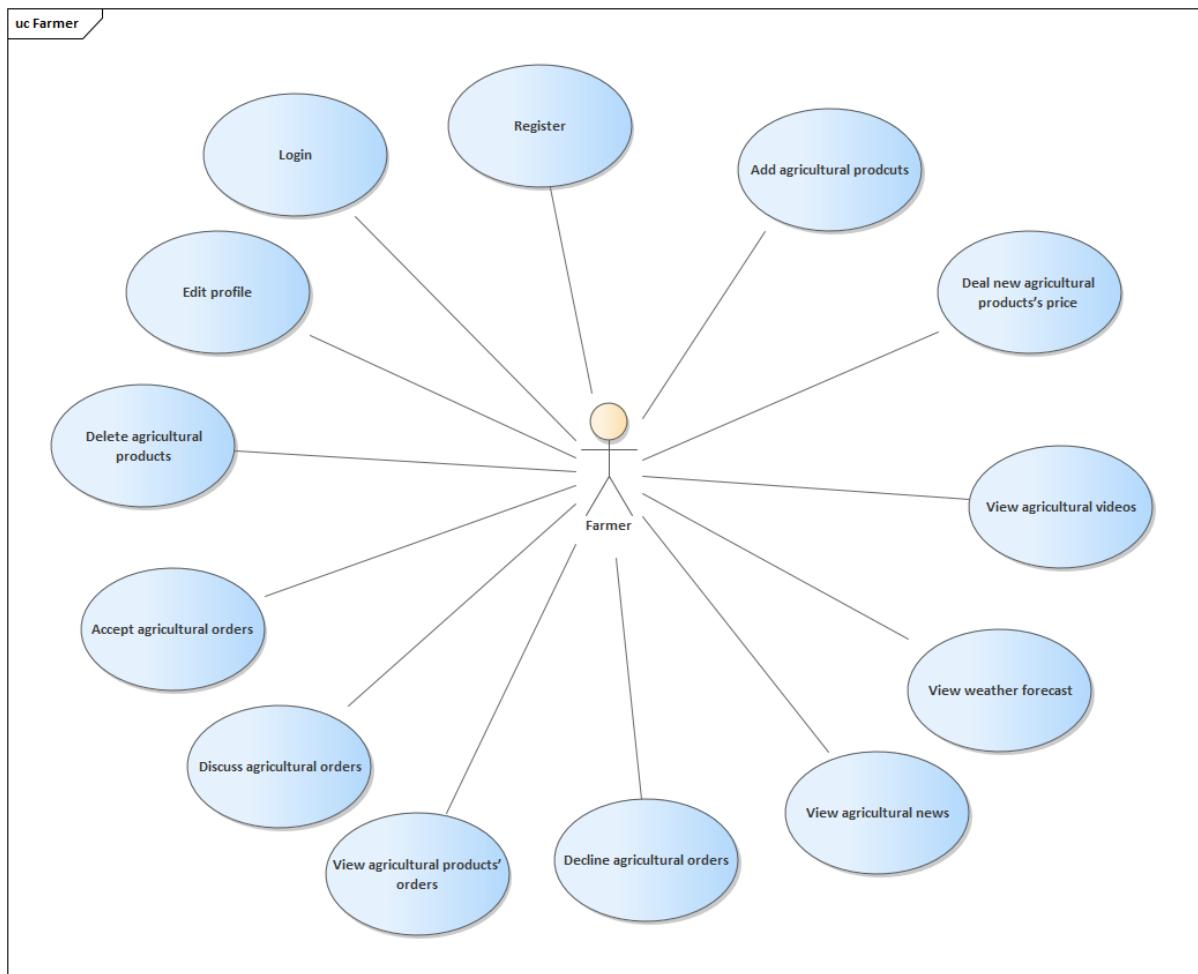


Figure 2. 3. Usecase for farmer

**\*Usecase for admin**

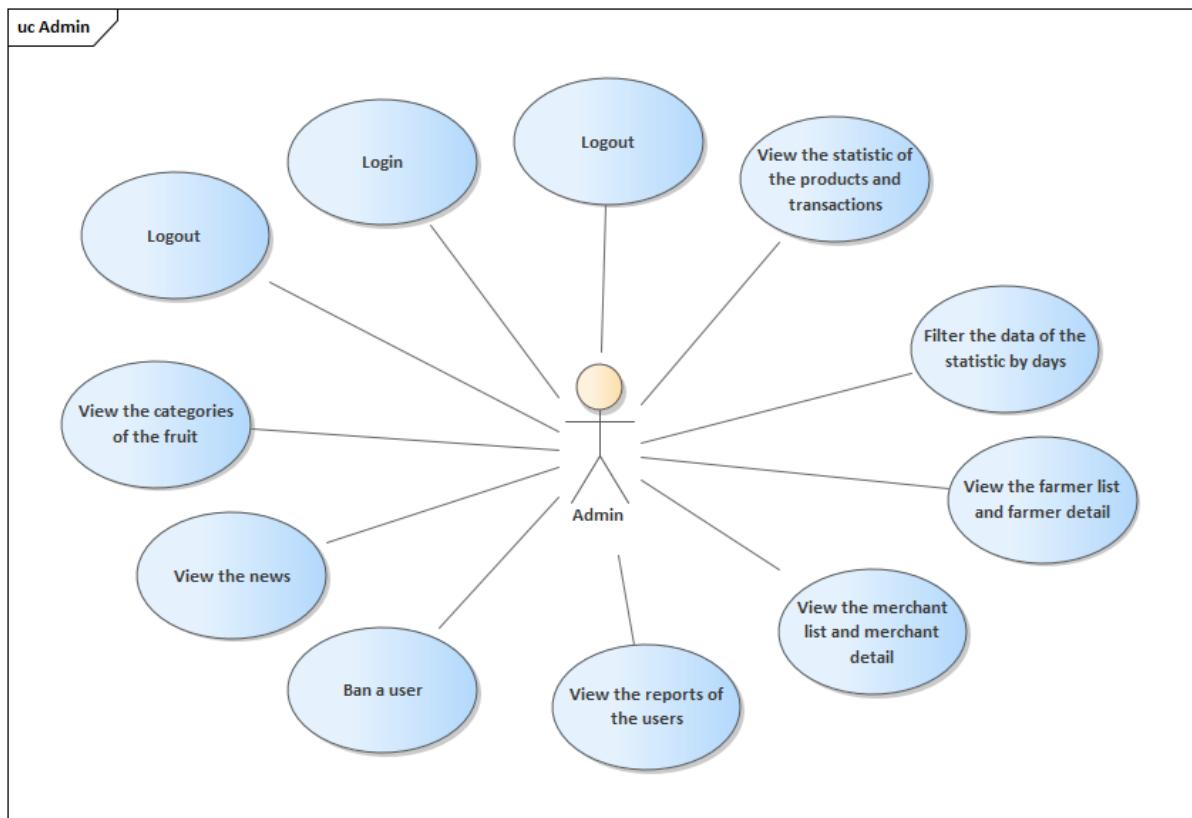


Figure 2. 4. Usecase for admin

## 4.2. Usecase specifications

### 4.2.1. Description of Login

Use Case Login	
<b>Description</b>	Actor logs into the system
<b>Actors</b>	Merchant (m), Farmer (f)
<b>Precondition</b>	When the actor wants to log into the system to do something
<b>Steps</b>	<p>(1) Actor open the mobile (m) / web (f)</p> <p>(2) Enter username and password into login form</p> <p>(3) Press login button</p> <p>(4) The system sends information to the database for validation.</p> <p>(5) Password and username are correct and the actor will log into the system with his/her correct role.</p>

*Table 2. 4. Description of Login*

### 4.2.2. Description of Edit Account

Use Case Edit Account	
<b>Description</b>	Actor edits the information of actor's account
<b>Actors</b>	Merchant (m), Farmer (f)
<b>Precondition</b>	When the actor wants to change the information on edit account
<b>Steps</b>	<p>(1) Actor open the mobile (m) / web (f)</p> <p>(2) Actor go to the setting information.</p> <p>(3) Actor can change the information (that needs to be checked for validation).</p>

	<b>(4)</b> Actor click on the Change button to update the new information and the data will be saved.
--	---

*Table 2. 5. Description of Edit Account*

#### 4.2.3. Description of Register

Use Case Register	
<b>Description</b>	Actor registers an account in the system
<b>Actors</b>	Merchant (m), Farmer (f)
<b>Precondition</b>	When the actor wants to have an account in the system to become a Merchant or a Farmer.
<b>Steps</b>	<p><b>(1)</b> Actor open the mobile (m) / web (f).</p> <p><b>(2)</b> Actor click to the Register (Sign up) Button and will be navigated to the Register Screen (Page).</p> <p><b>(3)</b> Actor need to fill in all required information and have that information checked for validation before sign up.</p> <p><b>(4)</b> Actor click on the Register to create a new account (after checking whether the account has existed or the password is correct or not).</p>

*Table 2. 6. Description of Register*

#### 4.2.4. Description of View List of agricultural products

Use View List of agricultural products	
<b>Description</b>	Actor sees a list of all Agricultural products.
<b>Actors</b>	Merchant (m)
<b>Precondition</b>	When the actor wants to see all Agricultural products.

<b>Steps</b>	<p>(1) Actor open the mobile (m)</p> <p>(2) Actor need to login the app and after login sucessful actor will be move to the Home Screen.</p> <p>(3) Actor click on the Explore Tab at the bottom and will see the catrgory and a list of products.</p>
--------------	--

*Table 2. 7. Description of View List of agricultural products*

#### **4.2.5. Description of View list of agricultural products that farmers own**

<b>Use Case View list of agricultural products that farmers own</b>	
<b>Description</b>	Actor sees all Agricultural products of a specific Farmer.
<b>Actors</b>	Merchant (m), Farmer (f)
<b>Precondition</b>	When the actor wants to see all Agricultural products of a specific Farmer.
<b>Steps</b>	<p>(1) Actor open the mobile (m) / web (f)</p> <p>Mobile:</p> <p>(2) In the Explore Tab, Actor can input the name of the farm and search.</p> <p>(3) The results will be shown (including the farm) the actor can click into the farm.</p> <p>(4) The products of that farm can be shown on that farm detail screen.</p>

*Table 2. 8. Description of View list of agricultural products that farmers own*

#### **4.2.6. Description of Search**

<b>Use Search</b>
-------------------

<b>Description</b>	Actor search for a specific Agricultural product in the system.
<b>Actors</b>	Merchant (m)
<b>Precondition</b>	When the actor wants to see a specific Agricultural product in the system.
<b>Steps</b>	<p>(1) Actor open the mobile (m)</p> <p>Mobile:</p> <p>(2) In the Explore Tab, Actor can input the name/image of the product and search.</p> <p>(3) The results will be shown (including the product) the actor can click into the product.</p> <p>(4) The product's detail of that farm can be shown on that product detail screen.</p>

*Table 2. 9. Description of Search*

#### 4.2.7. Description of Order

Use Case Order	
<b>Description</b>	Actor order an agricultural product.
<b>Actors</b>	Merchant (m)
<b>Precondition</b>	When the actor wants to buy an agricultural product in the system.
<b>Steps</b>	<p>(1) Actor open the mobile (m)</p> <p>(2) When found the need agricultural products, actor can order button</p> <p>(3) The result will be showed in order screen.</p>

*Table 2. 10. Description of Order*

#### 4.2.8. Description of Cancel the ordered agricultural product

Use Case Cancel the ordered agricultural product	
<b>Description</b>	Actor cancels an order of agricultural product.
<b>Actors</b>	Merchant (m)
<b>Precondition</b>	When the actor does not want the agricultural product.
<b>Steps</b>	<p>(1) Actor open the mobile</p> <p>(2) In the Waiting Order in the ‘Me’ Tab Actor can view the order that had been sent.</p> <p>(3) Actor can choose that order to view the detail screen of that order and choose the deleted button in the top right corner.</p>

Table 2. 11. Description of Cancel the ordered agricultural product

#### 4.2.9. Description of See the list of ordered agricultural products

Use Case See the list of ordered agricultural products	
<b>Description</b>	Actor sees the list of ordered agricultural products.
<b>Actors</b>	Merchant (m), Farmer (f)
<b>Precondition</b>	When the actor sees the list of ordered agricultural products.
<b>Steps</b>	<p>(1) Actor open the mobile (m) / web (f)</p> <p>Mobile:</p> <p>(2) In the Waiting Order in the ‘Me’ Tab Actor can view the order that had been sent.</p> <p>Web:</p> <p>(3) Open Order page to see.</p>

*Table 2. 12. Description of See the list of ordered agricultural products*

#### 4.2.10. Description of Deal new agricultural products's price

Use Case Deal new agricultural products's price	
<b>Description</b>	Actor accepts the deal of new agricultural products's price from the merchant.
<b>Actors</b>	Farmer (f)
<b>Precondition</b>	When the actor deals the new agricultural products's price from the merchant.
<b>Steps</b>	<p>(1) Actor open the web (f)</p> <p>(2) Click the discuss button in the order list.</p> <p>(3) A form to deal new price will be show for actor to fill information.</p> <p>(4) Click request button to sent the request.</p>

*Table 2. 13. Description of Deal new agricultural products's price*

#### 4.2.11. Description of Decision final deal price

Use Case Decision final deal price	
<b>Description</b>	Actor accepts the final deal price from the farmer.
<b>Actors</b>	Merchant (m)
<b>Precondition</b>	When the actor deals the final price of farmer's agricultural products.
<b>Steps</b>	<p>(1) Actor open the mobile (m).</p> <p>(2) If farmers deal new price, a message will be sent to actor.</p> <p>(3) The request will be showed in waiting screen with the mark "dealing"</p>

*Table 2. 14. Description of Decision final deal price*

#### 4.2.12. Description of Manage agricultural product

Use Case Manage agricultural product	
<b>Description</b>	Actor manages actor's agricultural products.
<b>Actors</b>	Farmer (f)
<b>Precondition</b>	When the actor wants to change actor's agricultural products.
<b>Steps</b>	<p>(1) Actor open the mobile web (f)</p> <p>(2) Open agricultural product page.</p> <p>(3) Fill all information to add new product or delete product.</p>

*Table 2. 15. Description of Manage agricultural product*

#### 4.2.13. Description of Validate agricultural products

Use Case Validate agricultural products	
<b>Description</b>	Actor validates the actor's agricultural products.
<b>Actors</b>	Farmer (f)
<b>Precondition</b>	When the actor validates the actor's agricultural products.
<b>Steps</b>	<p>(1) Actor open the web (f)</p> <p>(2) Open order page.</p> <p>(3) Here the actor can accept/discuss/decline the orders.</p>

*Table 2. 16. Description of Validate agricultural products*

### 4.3. Sequence diagrams

#### 4.3.1. Login

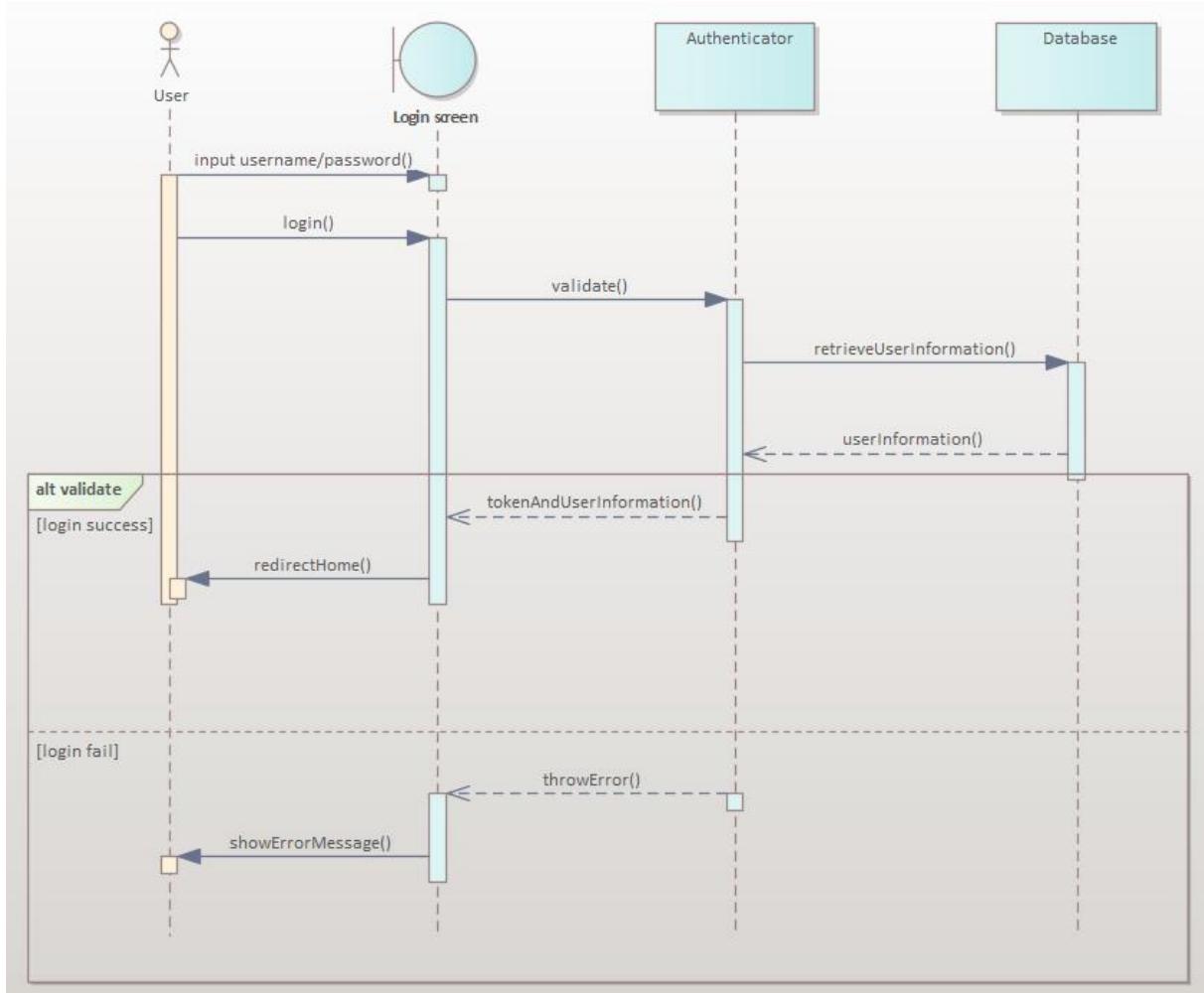


Figure 2. 5. Login sequence diagram

### 4.3.2. Register

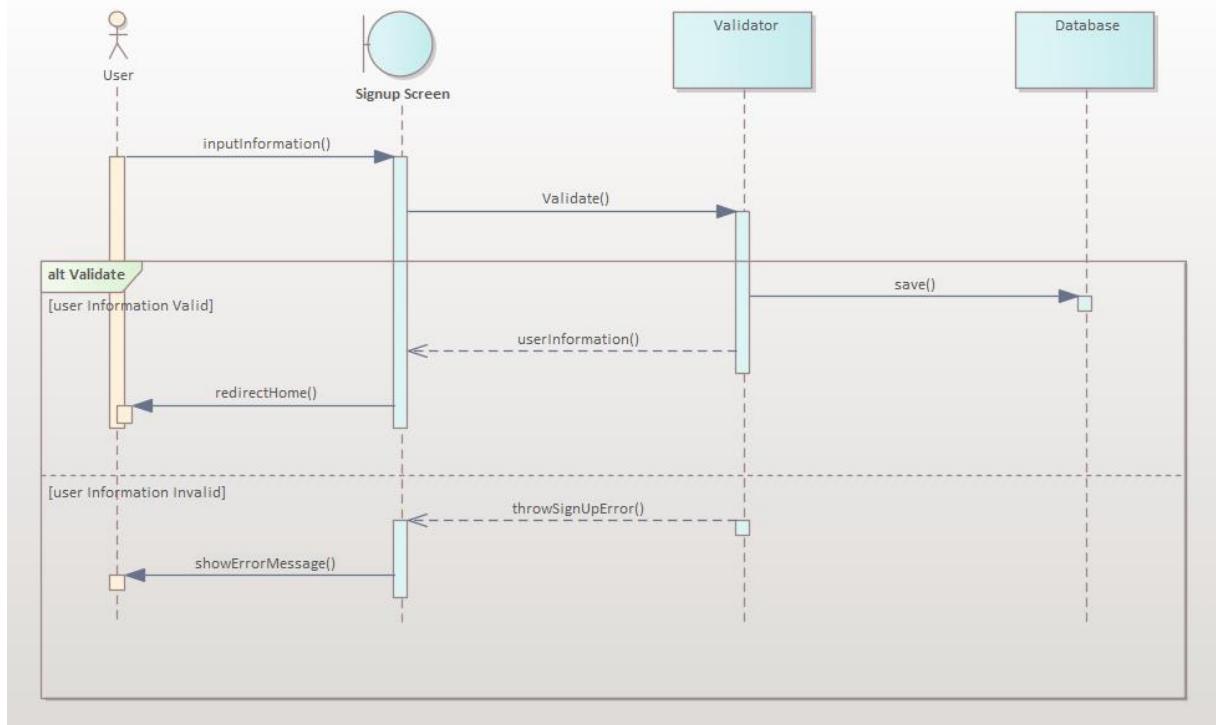
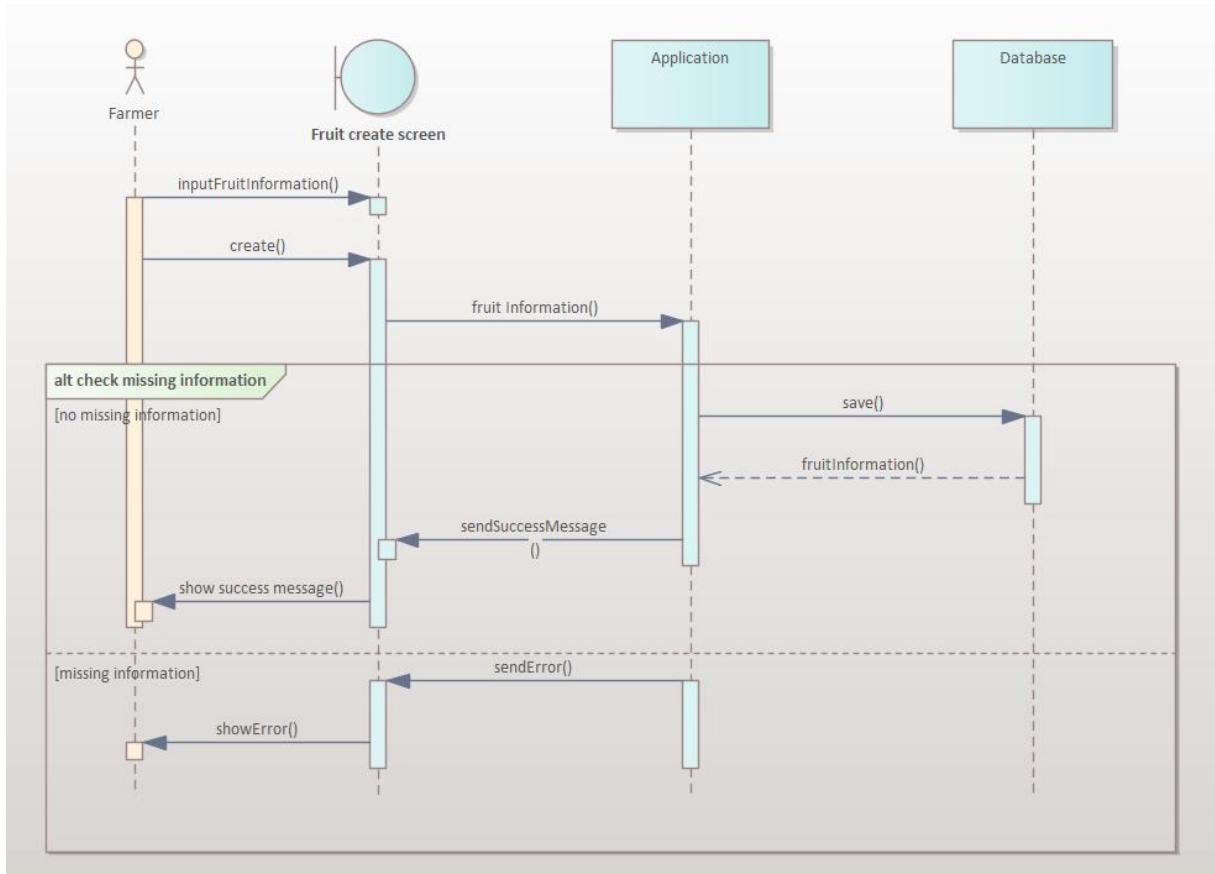


Figure 2. 6. Register sequence diagram

### 4.3.3. Create fruit



*Figure 2. 7. Create fruit sequence diagram*

#### 4.3.4. Show fruit

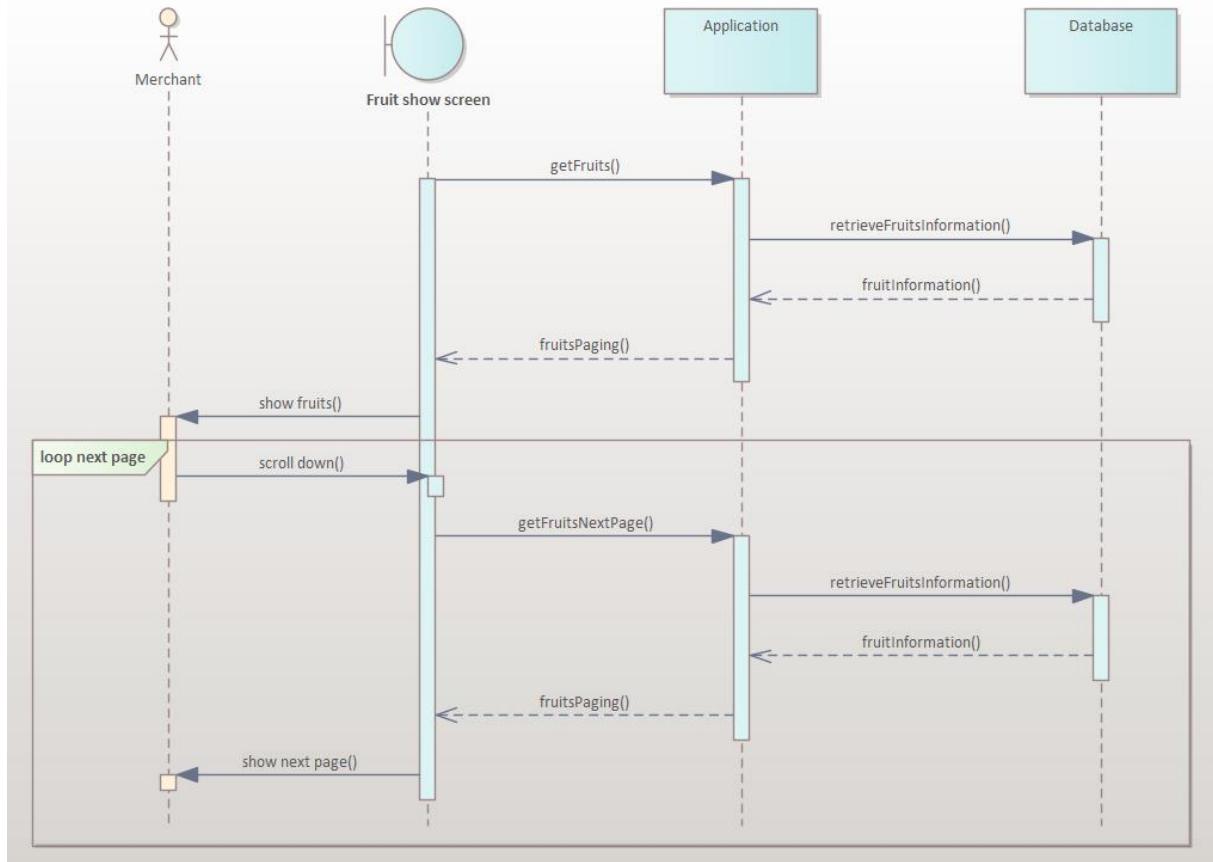


Figure 2. 8. Show fruit sequence diagram

### 4.3.5. Create order

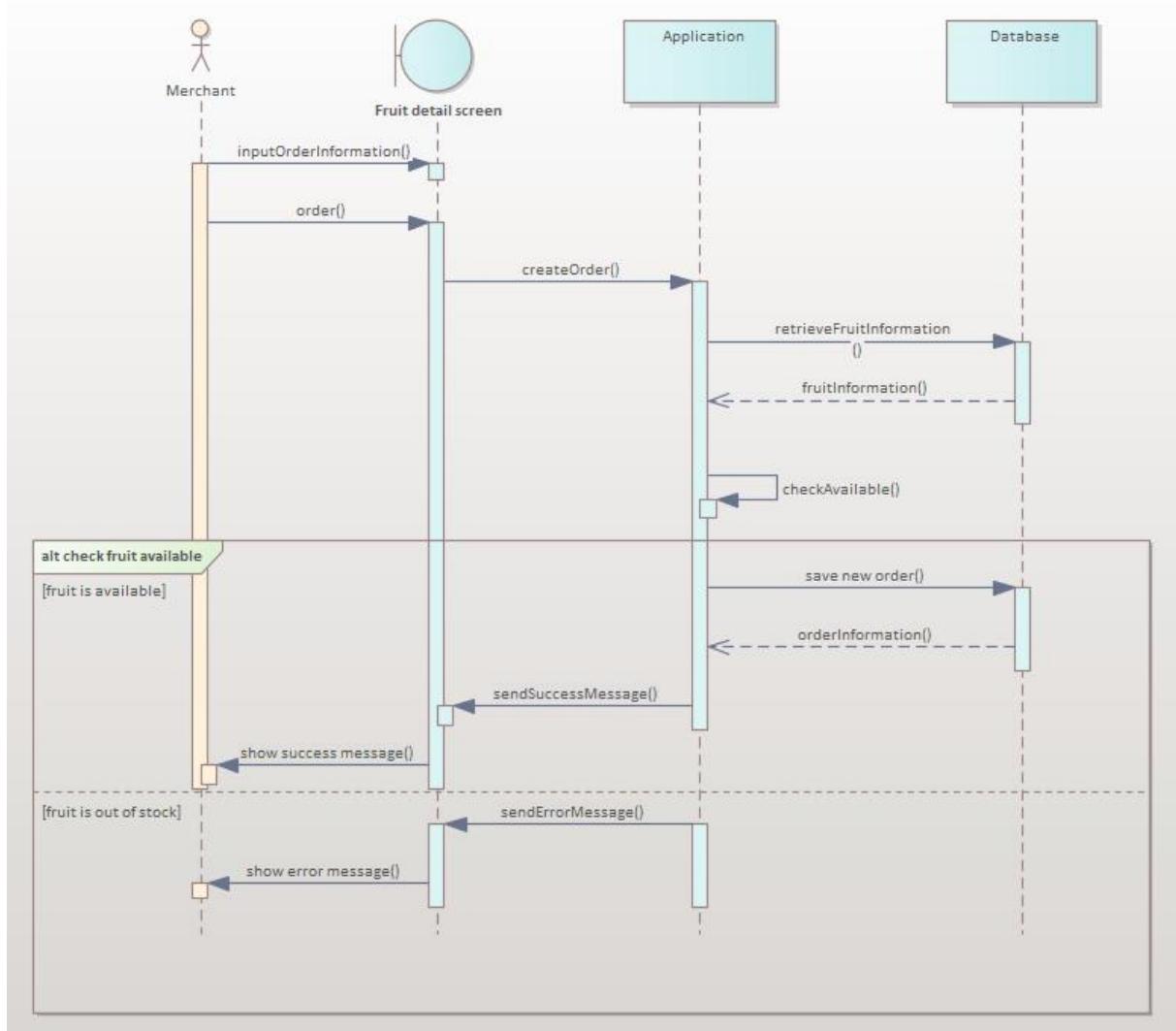


Figure 2. 9. Create order sequence diagram

#### 4.3.6. Accept order

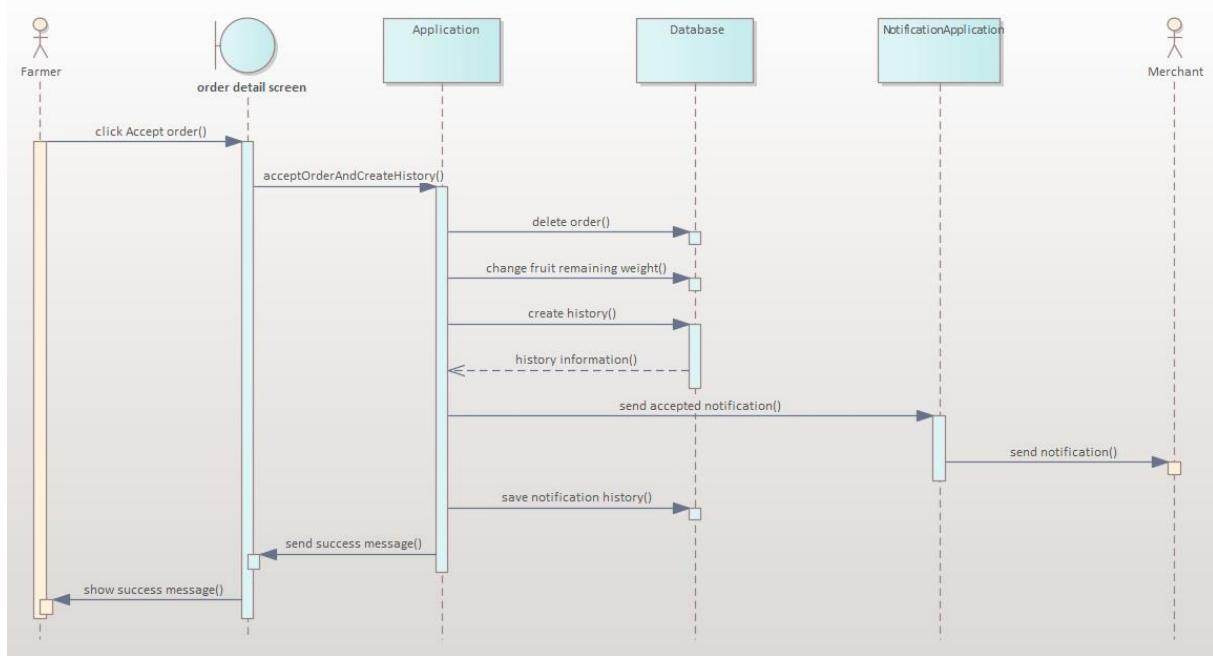


Figure 2. 10. Accept order sequence diagram

#### 4.3.7. Decline order

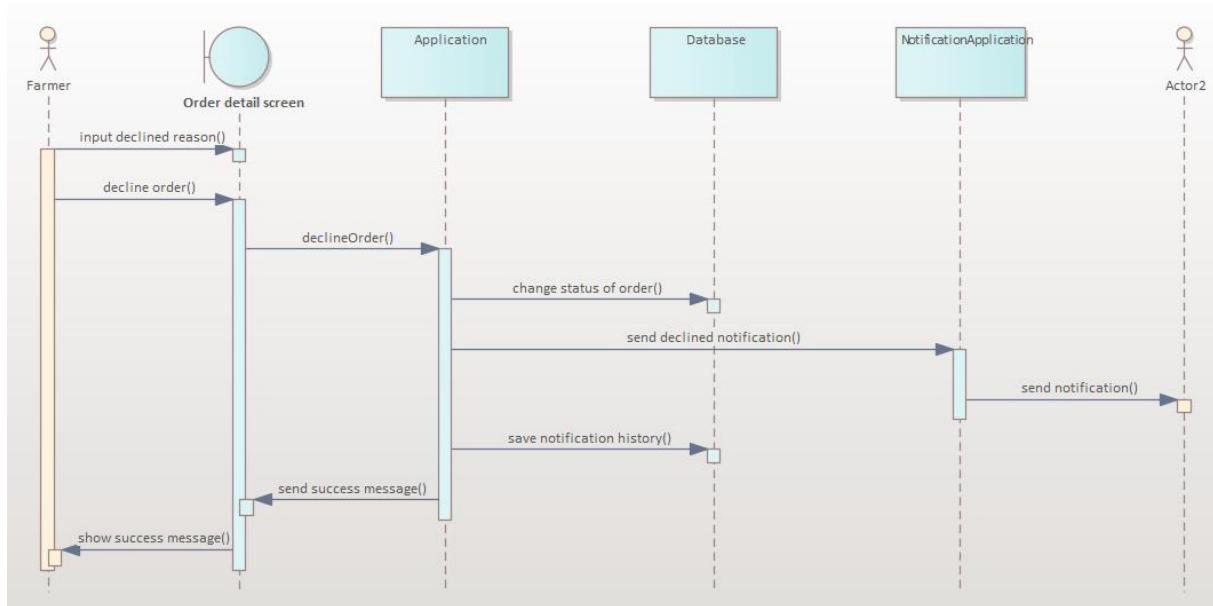


Figure 2. 11. Decline order sequence diagram

### 4.3.8. Report

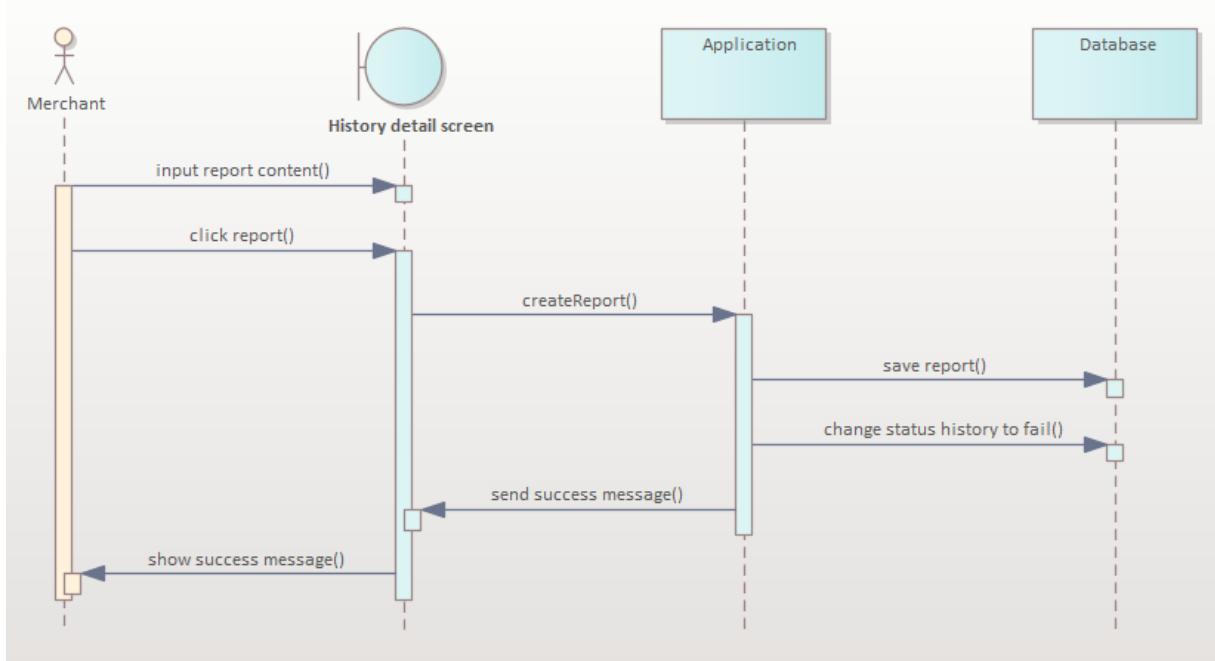


Figure 2. 12. Report sequence diagram

### 4.3.9. Complete

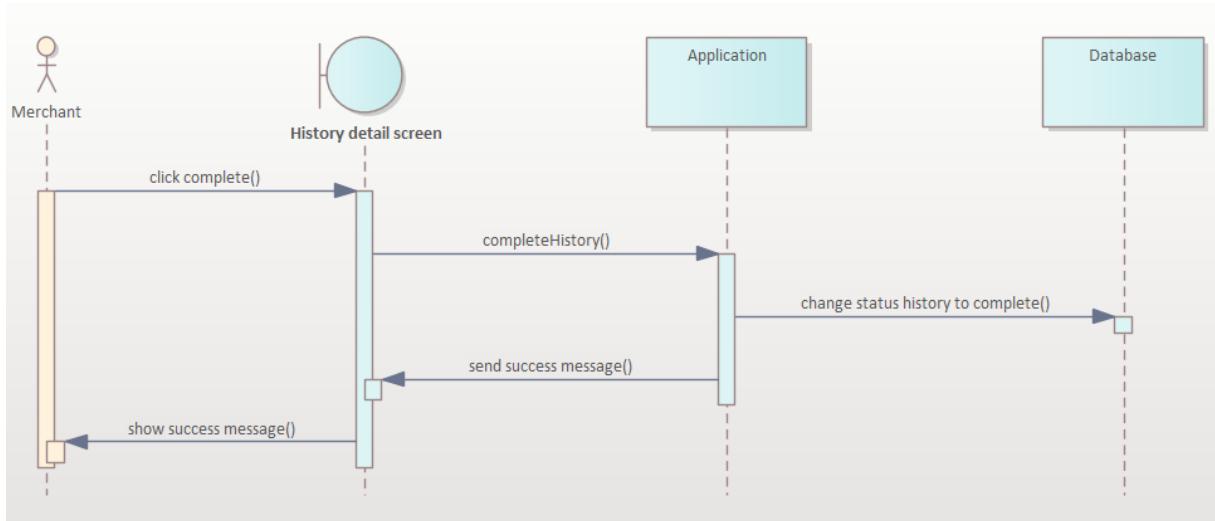


Figure 2. 13. Complete sequence diagram

#### 4.3.10. Search fruit

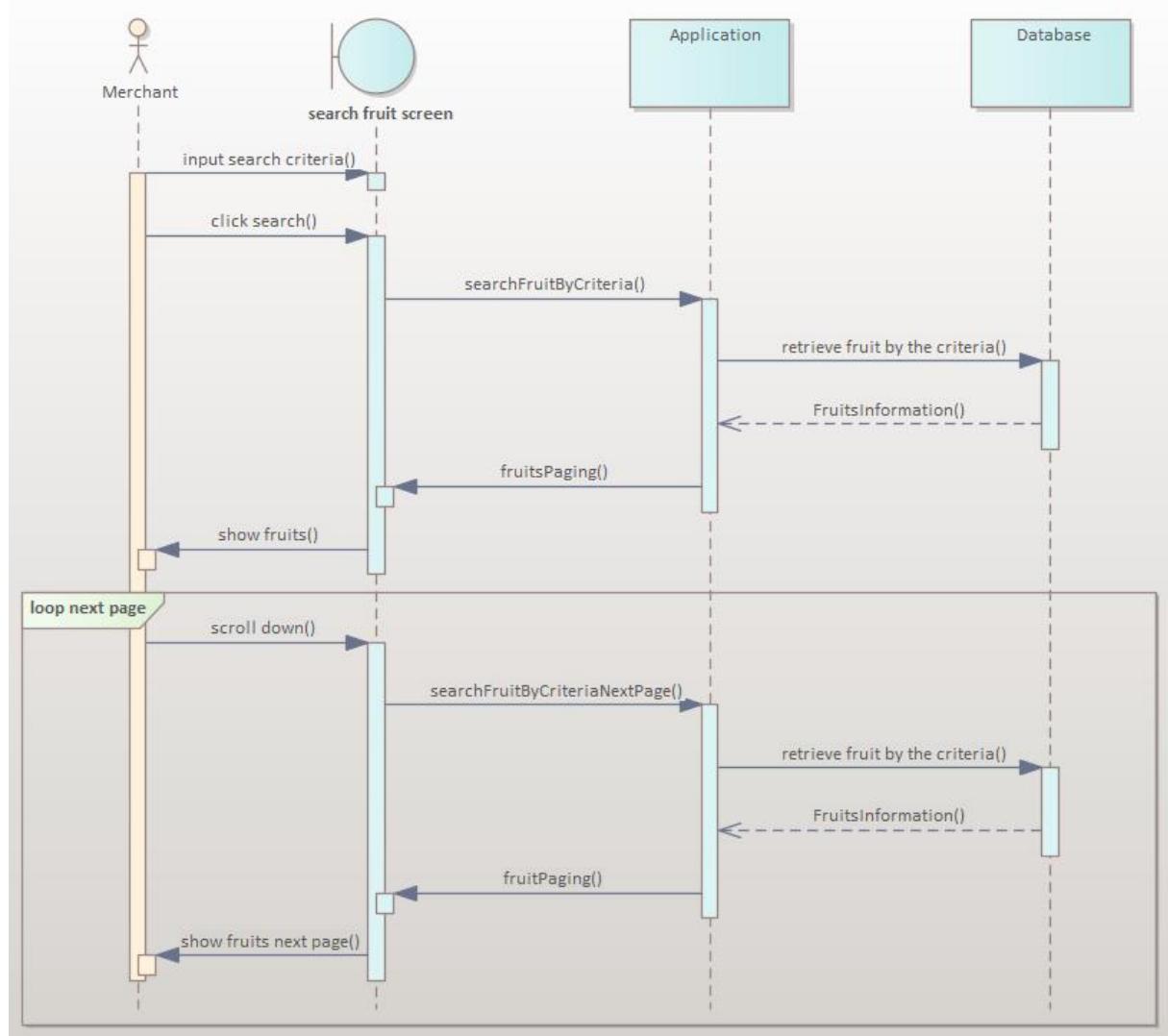


Figure 2. 14. Search fruit sequence diagram

#### 4.4. Database design

##### 4.4.1. Database table

###### 1. District

No	Property	Type	Description
1	id	int	Id
2	name	Varchar (255)	District Name

3	province_id	int	Province Id
---	-------------	-----	-------------

Table 2. 17. District database table

## 2. Fruit

No	Property	Type	Description
1	id	int	Id
2	date	date	Date
3	description	Varchar (255)	Description
4	name	Varchar (255)	Name
5	popular	Bit (1)	Popular
6	season	Varchar (255)	Season
7	unit	Varchar (255)	Unit
8	userid	int	User Id
9	weight	float	Weight

Table 2. 18. Fruit database table

## 3. Fruit image

No	Property	Type	Description
1	id	int	Id
2	fruit_id	int	Fruit Id
3	url	Varchar (255)	Url Of Image

Table 2. 19. Fruit image database table

## 4. History

No	Property	Type	Description
1	id	int	Id
2	amount	int	Amount
3	date	date	Date
4	farmer_id	int	Farmer Id
5	fruit_id	int	Fruit Id
6	merchant_id	int	Merchant Id
7	price	float	Price
8	transport	bool	Transport (Yes/No)

Table 2. 20. History database table

### 6. Location

No	Property	Type	Description
1	id	int	Id
2	address	Varchar (255)	Address
3	ward_id	int	Ward Id

Table 2. 21. Location database table

### 7. News

No	Column	Type	Description
1	id	int	Id
2	author	Varchar (255)	Author

3	category	Varchar (255)	Category
4	content	Varchar (255)	Content
5	date	date	Date
6	image_banner	Varchar (255)	Image Banner
7	image_content	Varchar (255)	Image Content
8	tittle	Varchar (255)	Tittle

Table 2. 22. News database table

### 8. Order

No	Property	Type	Description
1	id	int	Id
2	amount	int	Amount
3	date	date	Date
4	deal_amount	int	Deal Amount
5	deal_price	float	Deal Price
6	farmer_id	int	Farmer Id
7	fruit_id	int	Fruit Id
8	merchant_id	int	Merchant Id
9	price	float	Price
10	status_id	int	Status Id

11	transport	bool	Transport
----	-----------	------	-----------

Table 2. 23. Order database table

#### 9. Province

No	Property	Type	Description
1	id	int	Id
2	name	Varchar (255)	Name

Table 2. 24. Province database table

#### 10. Role

No	Property	Type	Description
1	id	int	Id
2	name	Varchar (255)	Name

Table 2. 25. Role database table

#### 11. Status product

No	Property	Type	Description
1	id	int	Id
2	name	Varchar (255)	Name

Table 2. 26. Status product database table

#### 12. Status user

No	Property	Type	Description
1	id	int	Id
2	name	Varchar (255)	Name

*Table 2. 27. Status user database table*

**13. User**

No	Property	Type	Description
1	id	int	Id
2	avatar	Varchar (255)	Avatar
3	birth_day	date	Birthday
4	create_day	date	Create Day
5	description	Varchar (255)	Description
6	email	Varchar (255)	Email
7	first_name	Varchar (255)	First Name
8	last_name	Varchar (255)	Last Name
9	location_id	int	Location Id
10	password	Varchar (255)	Password
11	phone	Varchar (255)	Phone
12	status_id	int	Status Id
13	username	Varchar (255)	Username

*Table 2. 28. User database table*

**14. User role**

No	Property	Type	Description
1	role_id	int	Role Id

2	user_id	int	User Id
---	---------	-----	---------

*Table 2. 29. User role database table*

**15. Role**

No	Property	Type	Description
1	id	int	Id
2	district_id	int	District Id
3	name	Varchar (255)	Name

*Table 2. 30. Role database table*

#### 4.4.2. Relational diagram

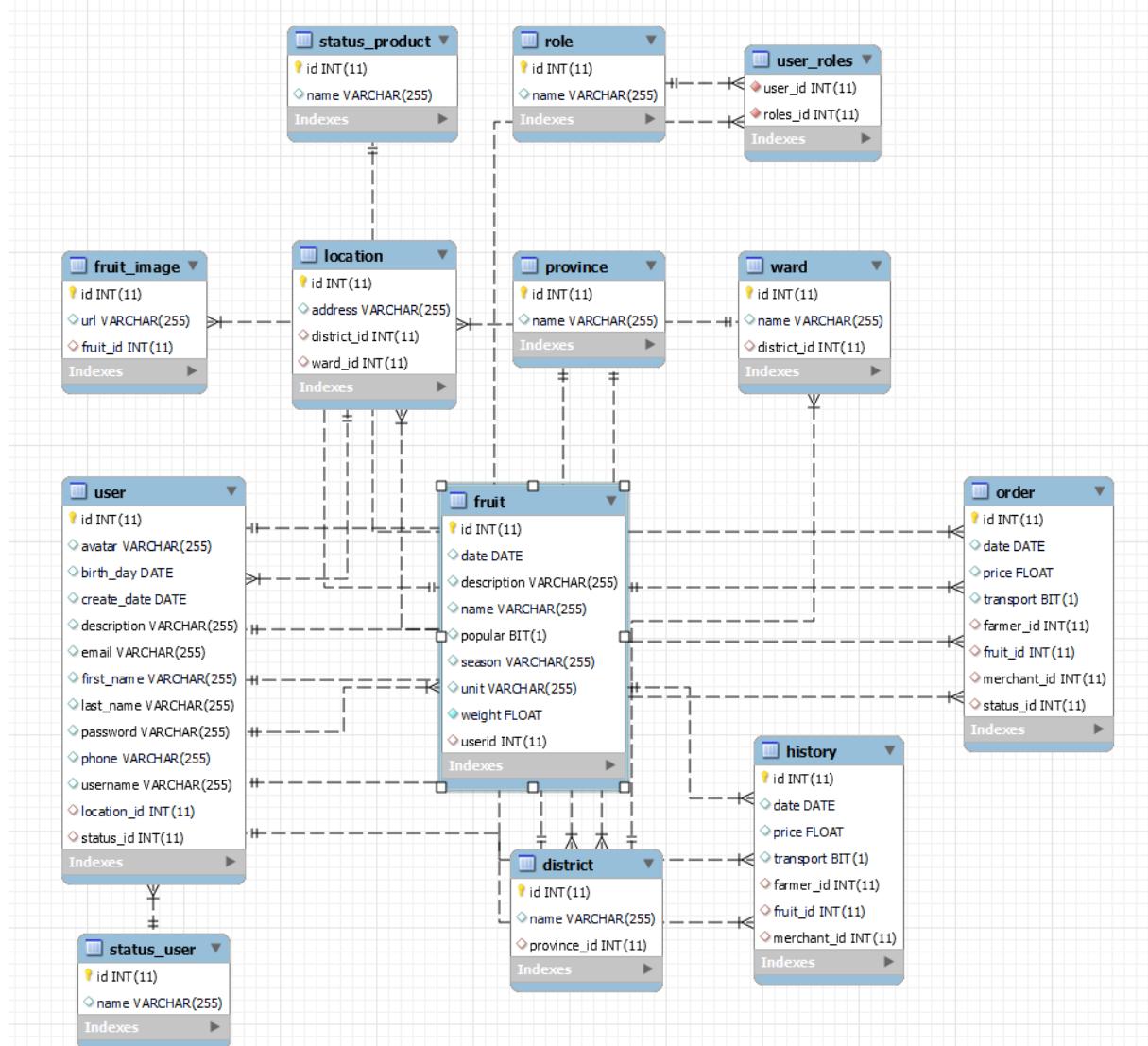


Figure 2. 15. Database relational diagram

## 4.5. User interface

### 4.5.1. Mobile for merchant

No.	Screen name	Description
1	Login	Login for registered users
2	Register	Users will be forced to fill in all of the fields in the form for registering an account
3	Home	Introduction about application
4	Me	Explore about payment and accountt
5	Explore	View and find information about products added by farmers
6	Category	View all categories of the products
7	Search	Help merchants can find which products they want by using text or image
8	Product detail	View all information about the product and merchants can order here
9	Payment waiting list	See a list of all ordered products
10	History	See a list of all completed orders
11	Order	View the details of the order and validate the discussion of new prices if have
12	Message	Chatting here

<b>13</b>	Notification	See all notification such as when farmers accept, discuss, decline, etc ...
<b>14</b>	Change password	Change the password if need
<b>15</b>	Change information	Change the personal information if need
<b>16</b>	Change language	Merchant can change language between English and Vietnamese

Table 2. 31. List of mobile screens for merchants



Figure 2. 16. Mobile login screen

No	Name	Type	Event
1	Input username	Textbox	For user input username
2	Input password	Textbox	For user input password

3	Login button	Button	Click to send user's account to backend for authorization
4	Signup button	Button	After clicking, go to register screen

Table 2. 32. Mobile login screen event

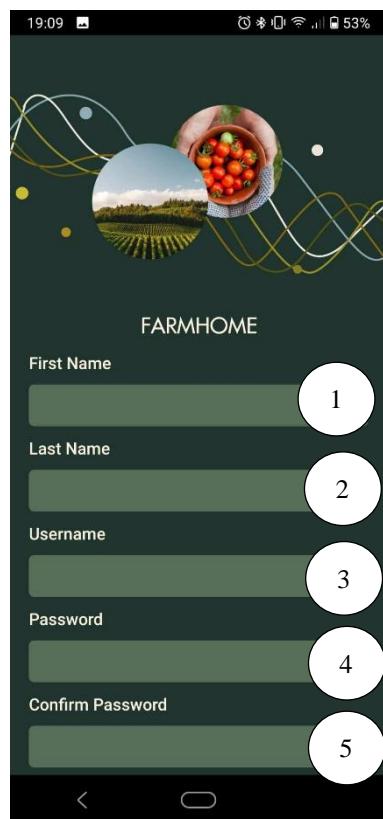


Figure 2. 17. Mobile register screen 1



Figure 2. 18. Mobile register screen 2

No	Name	Type	Event
1	Input firstname	Textbox	For user input firstname
2	Input lastname	Textbox	For user input lastname
3	Input username	Textbox	For user input username
4	Input password	Textbox	For user input password
5	Input confirm password	Textbox	For user input confirm password
6	Input email	Textbox	For user input email
7	Input phone	Textbox	For user input phone
8	Input birthday	Textbox	For user input birthday

9	Input city	Textbox	For user input city
10	Input district	Textbox	For user input district
11	Input ward	Textbox	For user input ward
12	Input address	Textbox	For user input address
13	Signup button	Button	Click to new user's information to send to backend for the register
14	Login button	Button	After clicking, go to login screen

Table 2. 33. Mobile register screen event

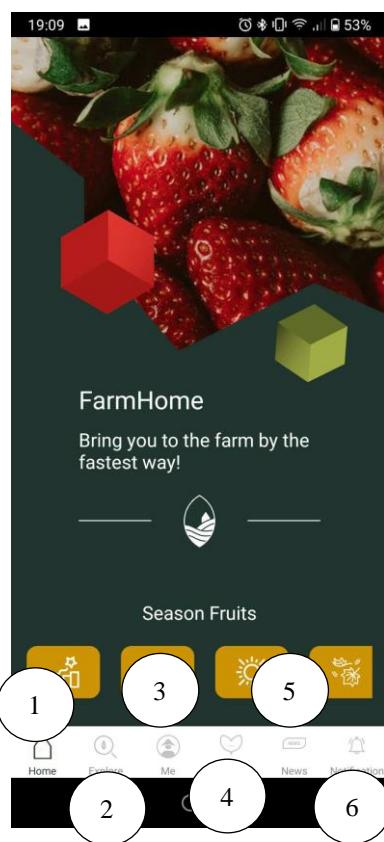


Figure 2. 19. Home screen

No	Name	Type	Event
----	------	------	-------

## CHAPTER 4: SYSTEM DESIGN

---

1	Home	Button	Go to home screen
2	Explore	Button	Go to explore screen
3	Me	Button	Go to presonal information screen
4	Message	Button	Go to message screen
5	News	Button	Go to news screen
6	Notification	Button	Go to notification screen

Table 2. 34. Home screen event

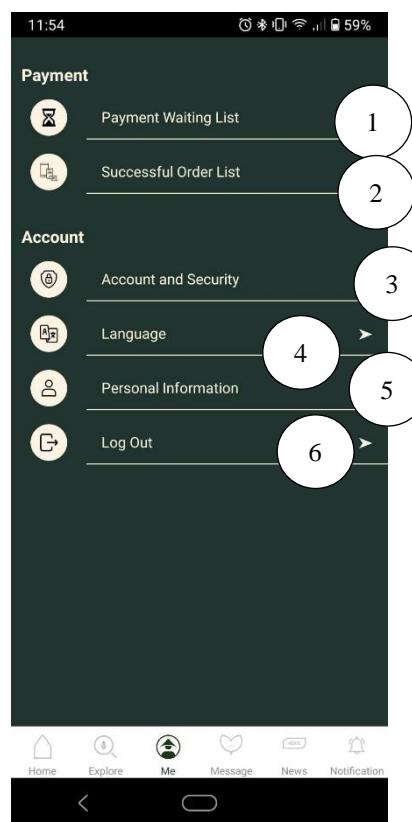


Figure 2. 20. Me screen

No	Name	Type	Event
1	Payment waiting list	Tab	Go to the screen that shows ordered list

2	Successful order list	Tab	Go to history of orders
3	Account and security	Tab	Go to presonal information screen
4	Language	Tab	Go to message screen
5	Personal information	Tab	Go to news screen
6	Log out	Tab	Go to notification screen

Table 2. 35. Me screen event



Figure 2. 21. Explore screen

No	Name	Type	Event
1	Search bar	Textbox	Help users search by text/image
2	Category	Tab	Filter with category

3	Product card	Card	Show some information about the product and can click to go to product details
---	--------------	------	--

Table 2. 36. Explore screen event

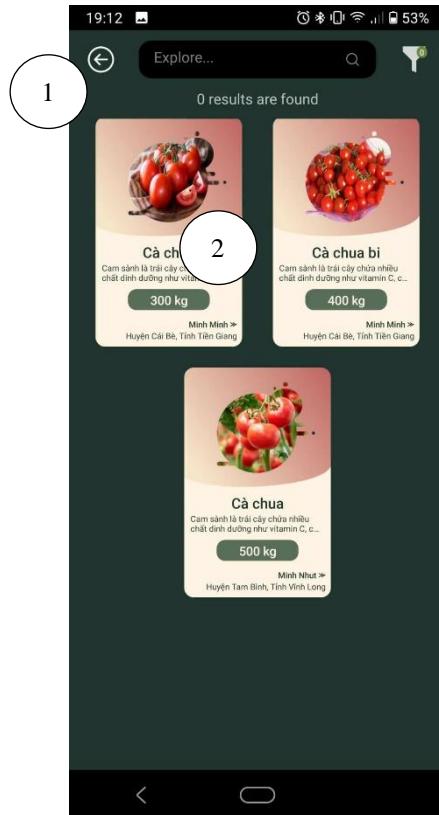


Figure 2. 22. Search screen

No	Name	Type	Event
1	Back button	Button	Back to previous screen
2	Product card	Card	Show some information about the product and can click to go to product details if found

Table 2. 37. Search screen event

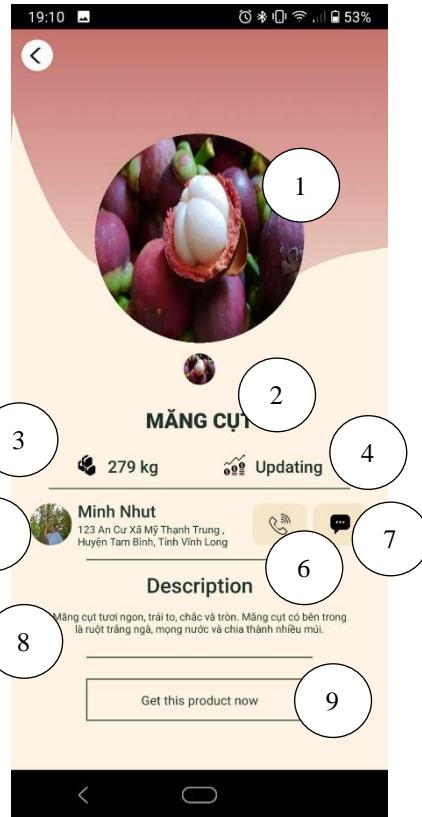


Figure 2. 23. Product detail screen

No	Name	Type	Event
1	Product image	Image	Show the image of product
2	Product name	Text	Show the name of product
3	Product weight	Text	Show the weight of product
4	Product market price	Text	Show the market price got from the reference market
5	Farmer information	Text	Show farmer image and information
6	Call button	Button	Make a phone call
7	Message button	Button	Go to chat

8	Description	Text	Show the description of product
9	Get product button	Button	Click to get the product

Table 2. 38. Product detail screen event

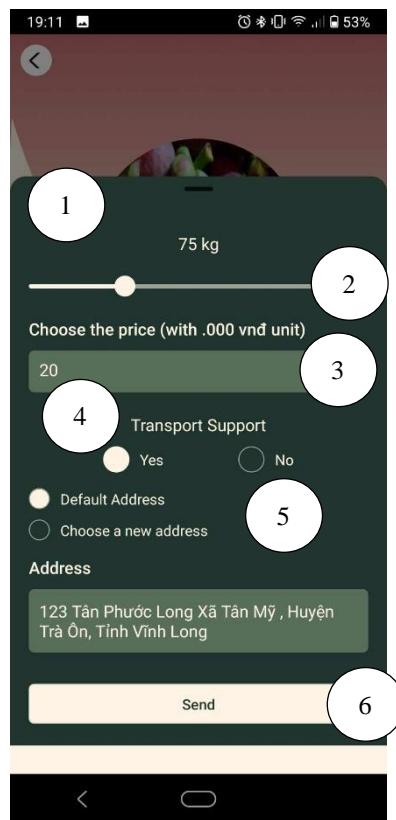


Figure 2. 24. Order bottom sheet

No	Name	Type	Event
1	Order sheet	Bottom sheet	Fill all information to get product
2	Weight slider	slider	Choose the weight user wanted
3	Price	Textbox	Input the price to buy the product
4	Transport support	Text	Show the market price got from the reference market

5	Address	Radio button	Choose the address if user need transport support
6	Send button	Button	Send order to farmer

Table 2. 39. Order event



Figure 2. 25. Payment waiting list

No	Name	Type	Event
1	Order card	Card	Show some information of orders and can click to go to the details of orders

Table 2. 40. Payment waiting list event



Figure 2. 26. History list

No	Name	Type	Event
1	History card	Card	Show history information

Table 2. 41. History list event

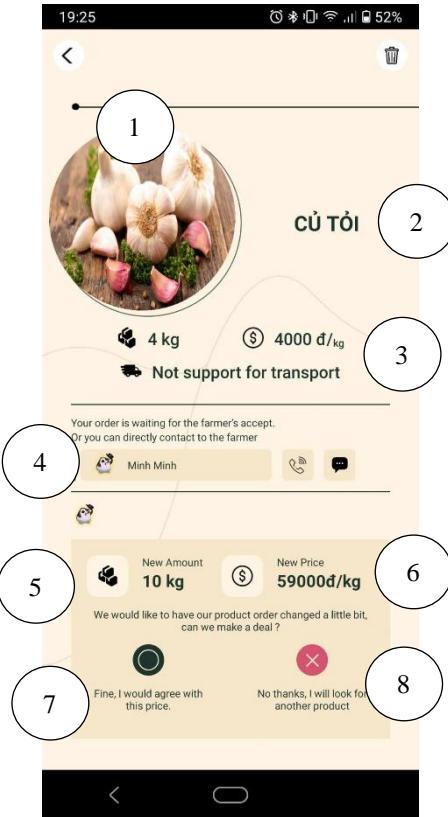


Figure 2. 27. Order details screen

No	Name	Type	Event
1	Product image	Image	Show the image of product
2	Product name	Text	Show the name of product
3	Order information	Text	Show the information of order: weight, suggested price, transport support
4	Farmer information	Text	Show the information of farmer
5	New amount	Text	New amount after discussion
6	New price	Button	New price after discussion
7	Confirm button	Button	Confirm information after discussion

8	Decline button	Button	Decline discussion information
---	----------------	--------	--------------------------------

Table 2. 42. Order details screen event

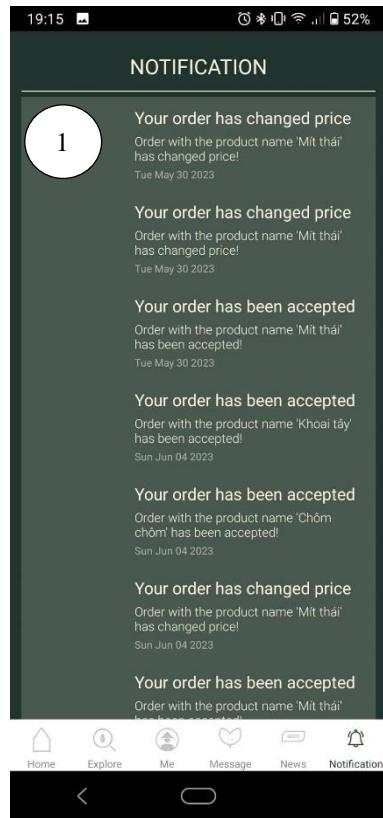


Figure 2. 28. Notification screen

No	Name	Type	Event
1	Notification	List	Show the list of notification

Table 2. 43. Notification screen event

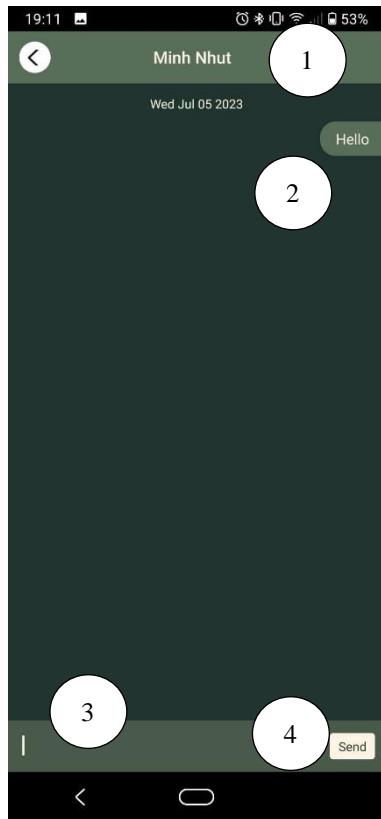


Figure 2. 29. Message screen

No	Name	Type	Event
1	Name	Text	Show the name of farmer
2	Chat area	Area	Show the message
3	Input text	Textbox	Input message
4	Send button	Button	Click to send the message

Table 2. 44. Message screen event

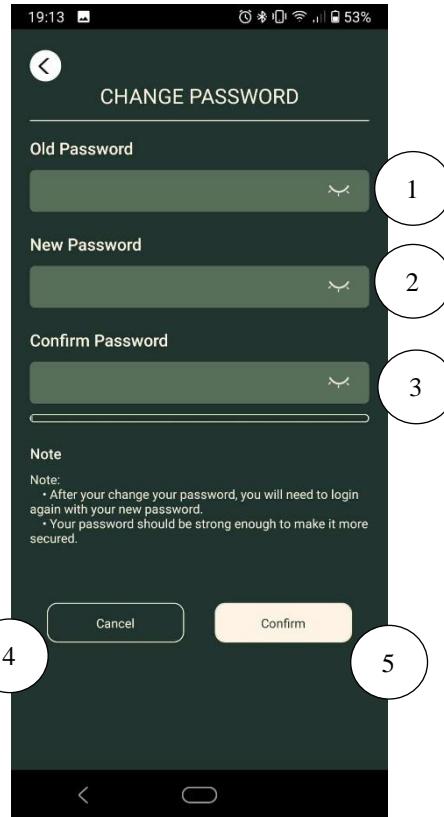


Figure 2. 30. Change password screen

No	Name	Type	Event
1	Input old password	Textbox	For user input old password
2	Input new password	Textbox	For user input new password
3	Input confirm password	Textbox	For user input confirm password
4	Cancel button	Text	Cancel the action
5	Confirm button	Button	Confirm the action

Table 2. 45. Change password screen event



Figure 2. 31. Update profile screen 1



Figure 2. 32. Update profile screen 2

No	Name	Type	Event
1	Avatar	Image	For user change image
2	Input firstname	Textbox	For user input firstname
3	Input lastname	Textbox	For user input lastname
4	Input email	Textbox	For user input email
5	Input phone	Textbox	For user input phone
6	Input birthday	Textbox	For user input birthday
7	Input city	Textbox	For user input city
8	Input district	Textbox	For user input district
9	Input ward	Textbox	For user input ward
10	Input address	Textbox	For user input address
11	Cancel button	Button	Cancel the action
12	Confirm button	Button	Click to update new information

Table 2. 46. Update profile screen event

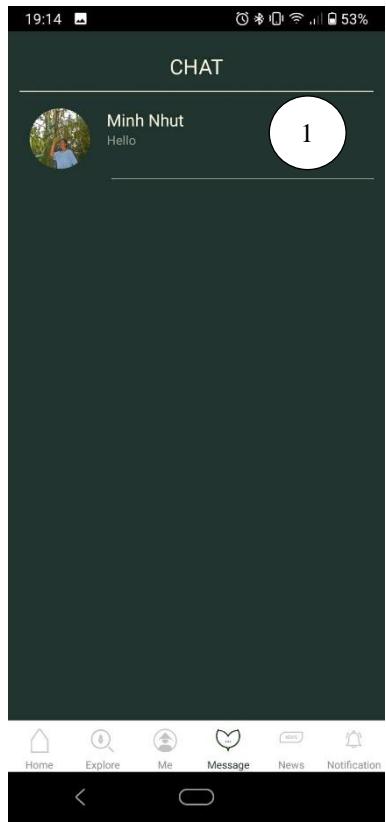


Figure 2. 33. List of message

No	Name	Type	Event
1	Message information	Card	Show some information of conservation

Table 2. 47. List of message event



Figure 2. 34. Change language screen

No	Name	Type	Event
1	English language	Radio button	Switch language to English
2	Vietnamese language	Radio button	Switch language to Vietnamese

Table 2. 48. Change language screen event

#### 4.5.2. Website for farmer

No.	Screen name	Description
1	Login	Login for registered users
2	Register	Users will be forced to fill in all of the fields in the form for registering an account

<b>3</b>	Home	Introduction about application
<b>4</b>	Overview	Some basic statistics
<b>5</b>	Prices	Go to reference market
<b>6</b>	Products	Display of information and product management
<b>7</b>	Orders	View the details of the order and validate the orders
<b>8</b>	History	See a list of all completed orders
<b>9</b>	Profile	Manage profile information
<b>10</b>	News	View news
<b>11</b>	Message	Chatting here

Table 2. 49. List of website page for farmer

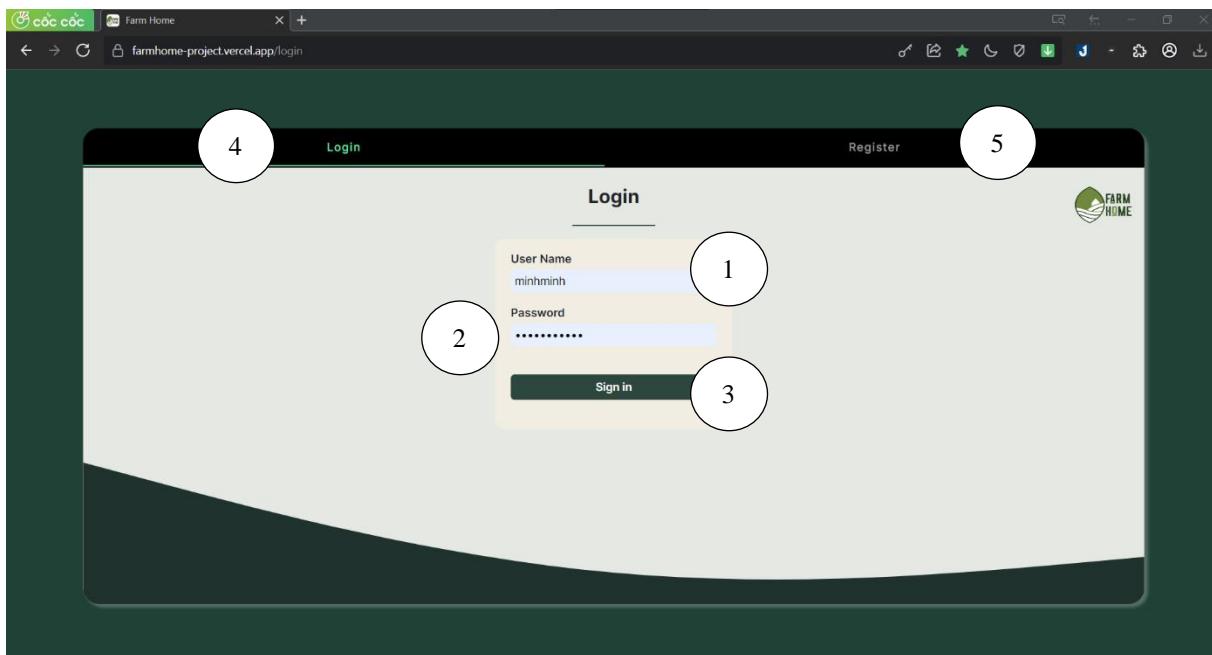


Figure 2. 35. Login page

No	Name	Type	Event
1	Input username	Textbox	For user input username
2	Input password	Textbox	For user input password
3	Login button	Button	Click to send user's account to backend for authorization
4	Login tab	Tab	After clicking, switch to login panel
5	Register tab	Tab	After clicking, switch to register panel

Table 2. 50. Login page event

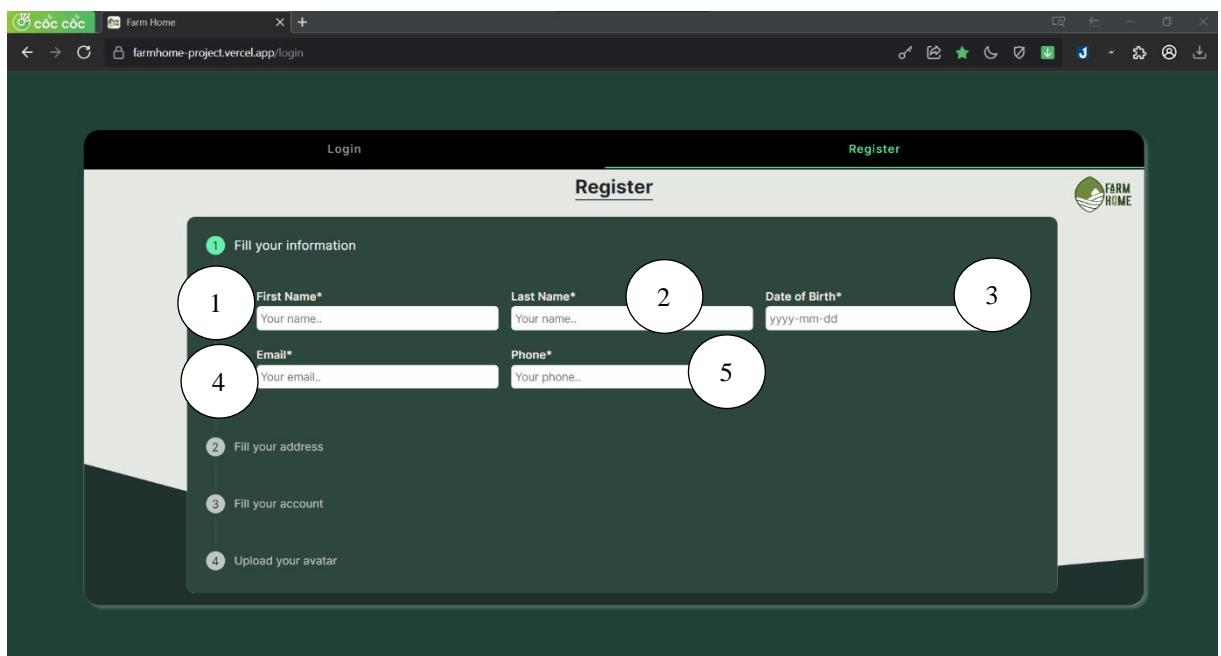
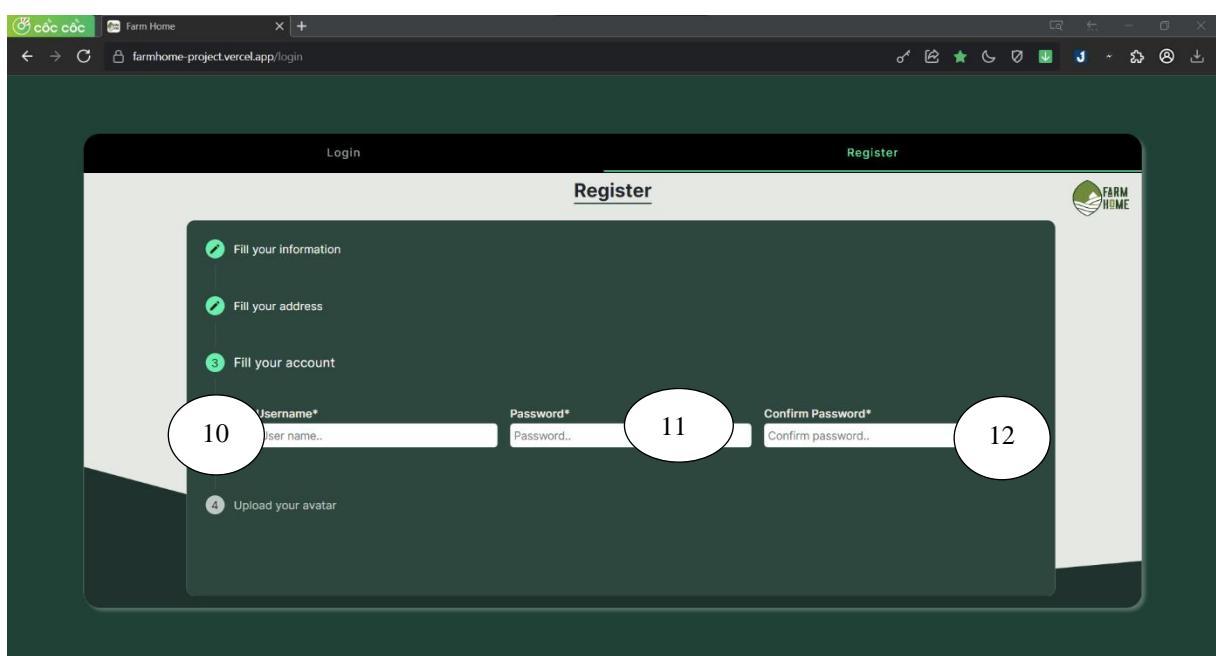
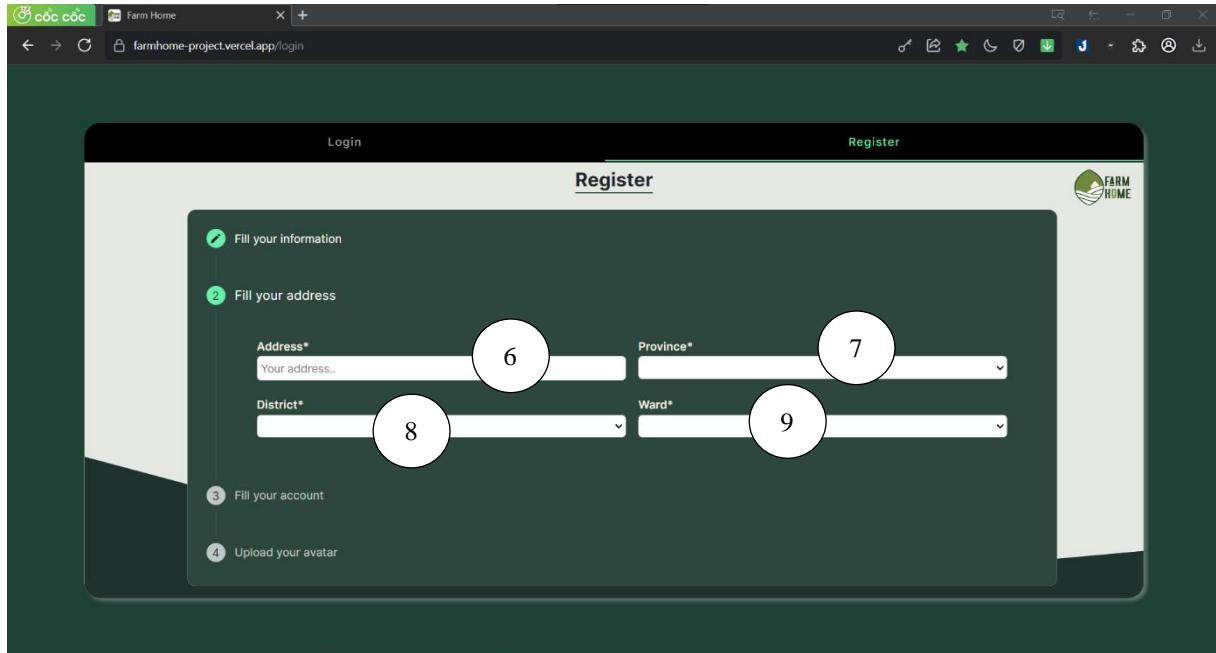


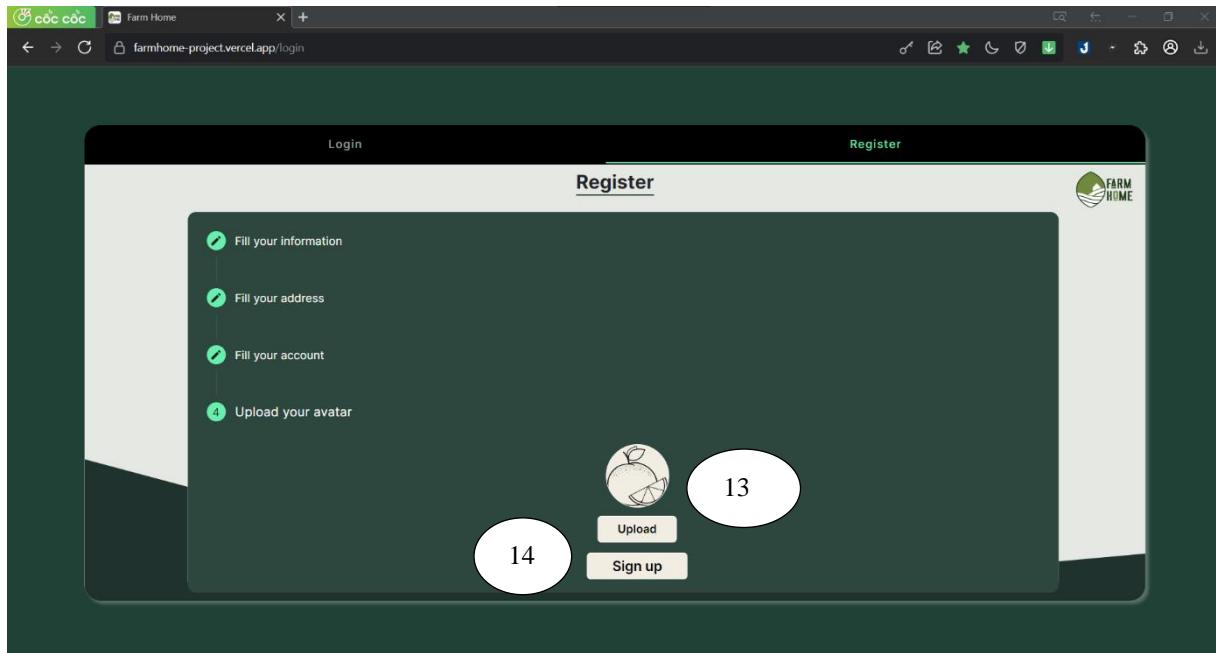
Figure 2. 36. Register page

## CHAPTER 4: SYSTEM DESIGN

---



## CHAPTER 4: SYSTEM DESIGN



No	Name	Type	Event
1	Input firstname	Textbox	For user input firstname
2	Input lastname	Textbox	For user input lastname
3	Input birthday	Textbox	For user input birthday
4	Input email	Textbox	For user input email
5	Input phone	Textbox	For user input phone
6	Input address	Textbox	For user input address
7	Input city	Textbox	For user input city
8	Input district	Textbox	For user input district
9	Input ward	Textbox	For user input ward
10	Input username	Textbox	For user input username
11	Input password	Textbox	For user input password

12	Input confirm password	Textbox	For user input confirm password
13	Upload image button	Button	Help user choose avatar
14	Register button	Button	Click to new user's information to send to backend for the register

Table 2. 51. Register page event

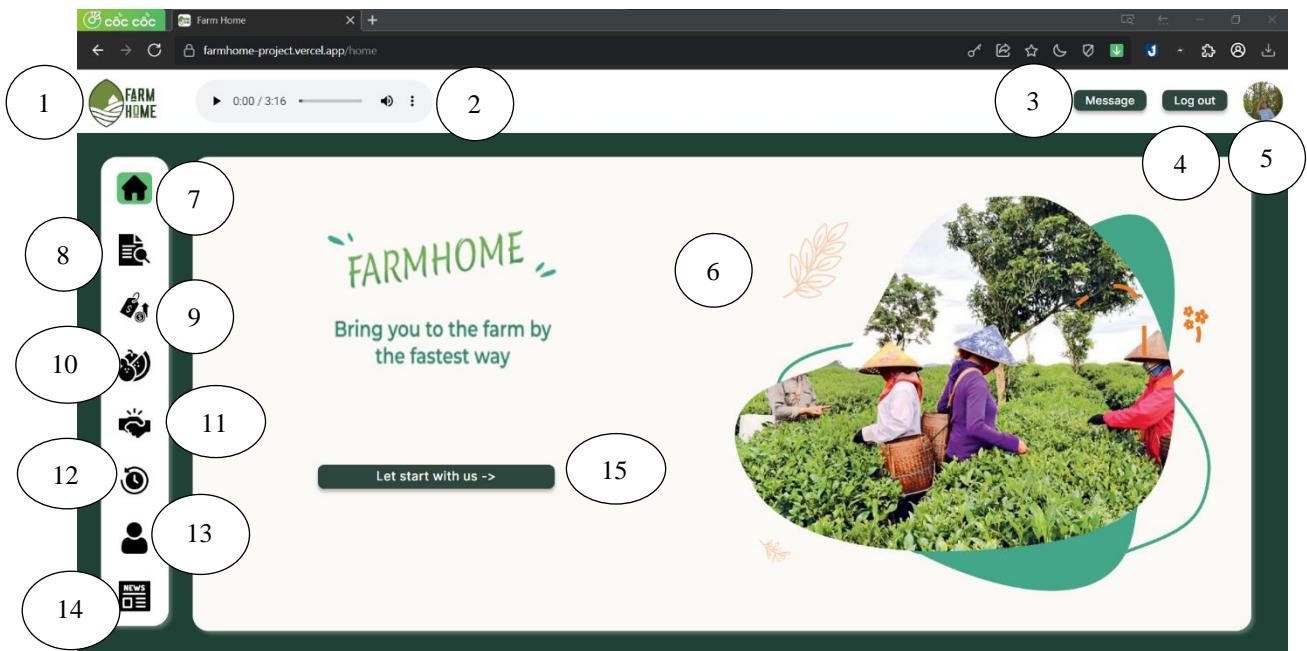


Figure 2. 37. Home page

No	Name	Type	Event
1	Farmhome logo	Image	Click to go to home page
2	Music player	Mp3	Play music to chill
3	Message button	Button	Click to go to message page
4	Log out button	Button	Click to go log out
5	User avatar	Image	Show user avatar
6	Home banner	Image	Show slogan and image intro

7	Home	Button	Click to go to home page
8	Overviews	Button	Click to go to overviews page
9	Prices	Button	Click to go to price page
10	Products	Button	Click to go to products page
11	Orders	Button	Click to go to orders page
12	History	Button	Click to go to history page
13	Profile	Button	Click to go to profile page
14	News	Button	Click to go to news page
15	Let start button	Button	Click to go to product page

Table 2. 52. Home page event

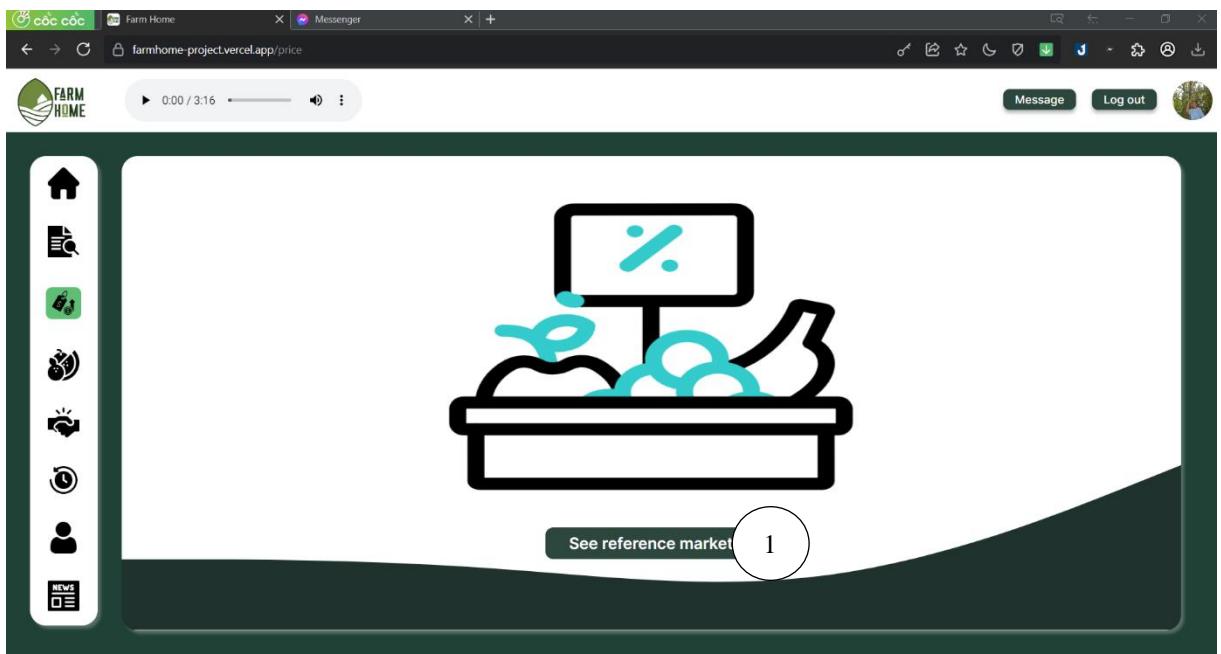


Figure 2. 38. Reference page

## CHAPTER 4: SYSTEM DESIGN

The screenshot shows a web browser with a table titled "Cập nhật giá nông sản chợ đầu mối Thủ Đức". The table lists various vegetables with their names in Vietnamese and their prices per kilogram. To the right of the table is a sidebar with several food-related articles, each with a thumbnail, title, and a timestamp indicating when it was published.

STT	Tên sản phẩm	Giá cả thị trường (VNĐ/kg)
1	Rau lá, củ quả	
1	Cải thảo	12.000
2	Xà lách búp	10.000
3	Cải bắp tròn	16.000
4	Cải ngọt	9.000
5	Cải bẹ xanh	17.000
6	Rau muống nước	22.000
7	Rau muống hạt	11.000
8	Cải thia	8.000
9	Rau quế	12.000
10	Bầu	6.000
11	Su su	7.000
12	Khoai lang bí	10.000
13	Cà chua	6.000
14	Bông cải xanh	18.000

Figure 2. 39. Reference price website

No	Name	Type	Event
1	See reference market button	Button	Click to go to reference market

Table 2. 53. Prices page event

The screenshot shows a products page with a sidebar menu on the left. The main area displays four fruit items (#1 Măng cụt, #2 Chôm chôm, #3 Ói nữ hoàng, #4 Bòn bon thái) with their details: name, having weight, remaining weight, category, season, date, and description. Each item has a "Market price" field, some of which have "no data". There are "Click to Update" and "Click to Delete" buttons for each item. A "Refresh" button and a "+ Add new product" button are also visible.

Figure 2. 40. Products page

No	Name	Type	Event
1	Refresh button	Button	Click to refresh the list of products
2	Add new product button	Button	Open the dialog to add new product
3	Market price	Text	Show reference price if available
4	Product image	Image	Show product image
5	Product information	Text	Show product information
6	Expansion panel	Panel	Expand to update/delete product

Table 2. 54. Products page event

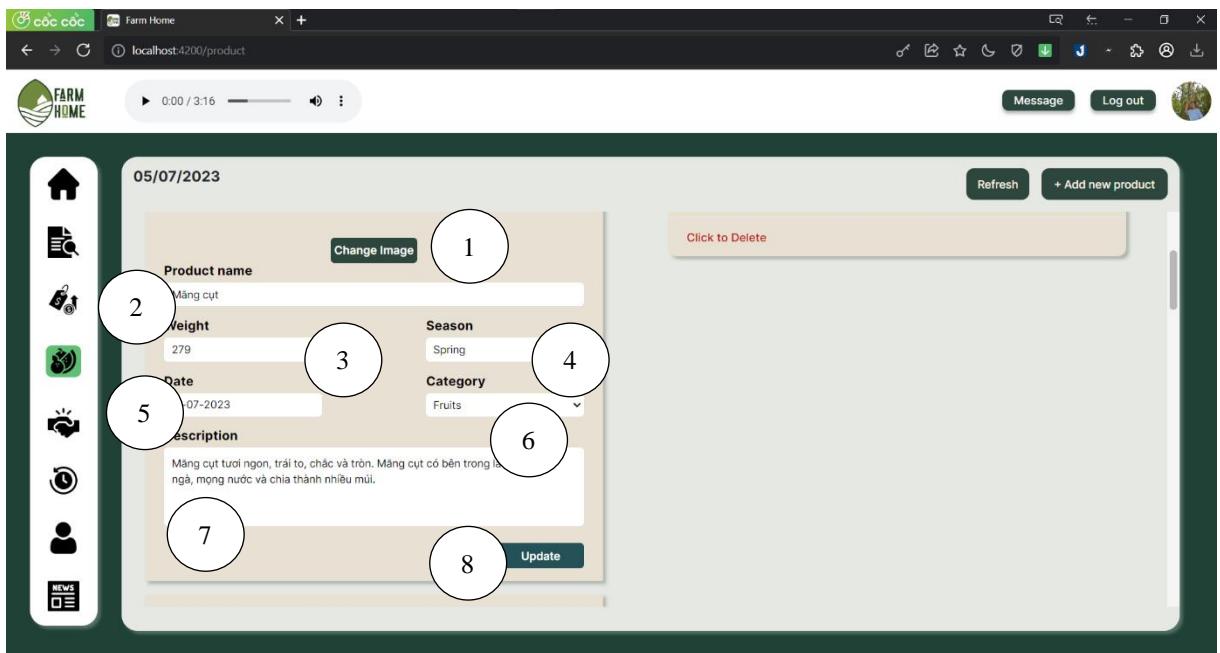


Figure 2. 41. Update product

No	Name	Type	Event
1	Change image button	Button	Click to get new image for product
2	Input product name	Textbox	For user input product name

## CHAPTER 4: SYSTEM DESIGN

3	Input product weight	Textbox	For user input product weight
4	Input product season	Textbox	For user input product season
5	Input product date	Textbox	For user input product date
6	Input product category	Textbox	For user input product category
7	Input product description	Textarea	For user input product description
8	Update button	Button	Click to update new information

Table 2. 55. Update product event

Figure 2. 42. Delete product

No	Name	Type	Event
1	Delete button	Button	Click delete the product

Table 2. 56. Delete product event

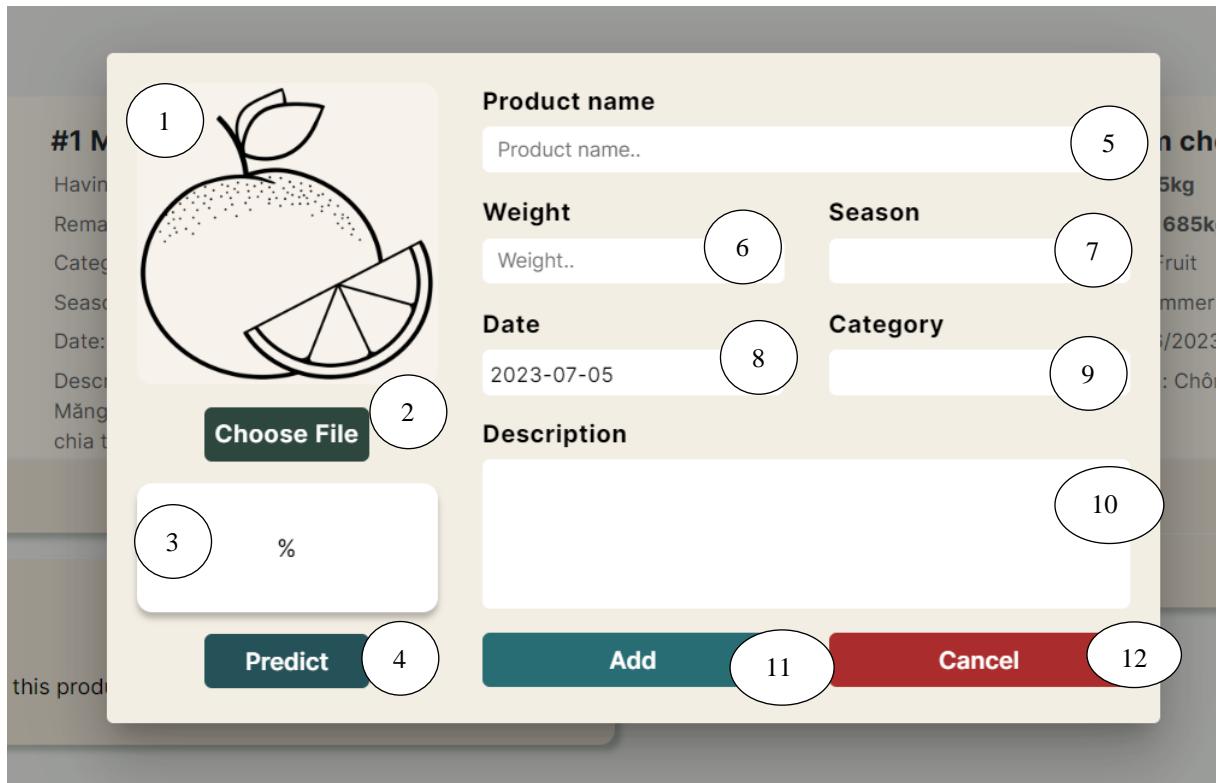


Figure 2. 43. Add product dialog

No	Name	Type	Event
1	Product image	Image	Show image after got
2	Choose image	Button	Click to add image for product
3	Detect result	Text area	Show the confidence and name after detected
4	Predict button	Button	Click to detect fruit
5	Input product name	Textbox	For user input product name
6	Input product weight	Textbox	For user input product weight
7	Input product season	Textbox	For user input product season
8	Input product date	Textbox	For user input product date
9	Input product category	Textbox	For user input product category

## CHAPTER 4: SYSTEM DESIGN

10	Input product description	Textarea	For user input product description
11	Add button	Button	Click to add new product with all filled information
12	Cancel	Button	Close the dialog

Table 2. 57. Add product dialog event

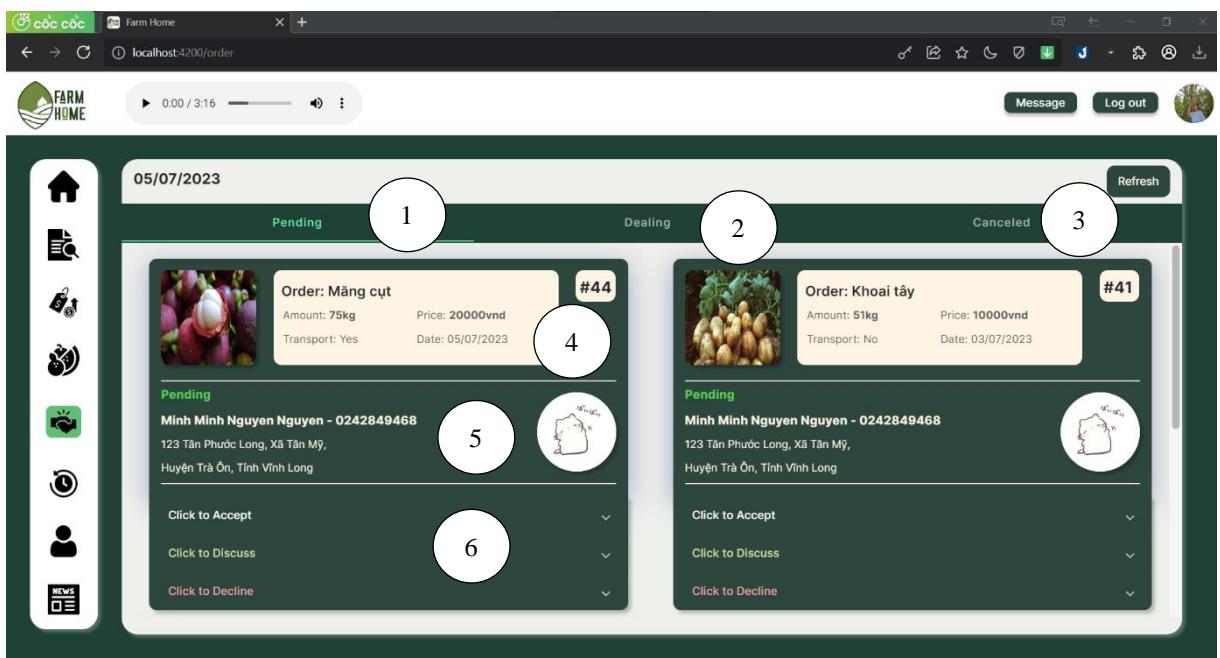


Figure 2. 44. Order page

No	Name	Type	Event
1	Pending tab	Tab	After clicking, switch to pending panel
2	Dealing tab	Tab	After clicking, switch to dealing panel
3	Canceled tab	Tab	After clicking, switch to canceled panel
4	Order information	Text	Show all information of the order

5	Merchant information	Text	Show all information of the merchant that ordered
6	Expansion panel	Panel	Expand to accept/discuss/decline order

Table 2. 58. Order page event

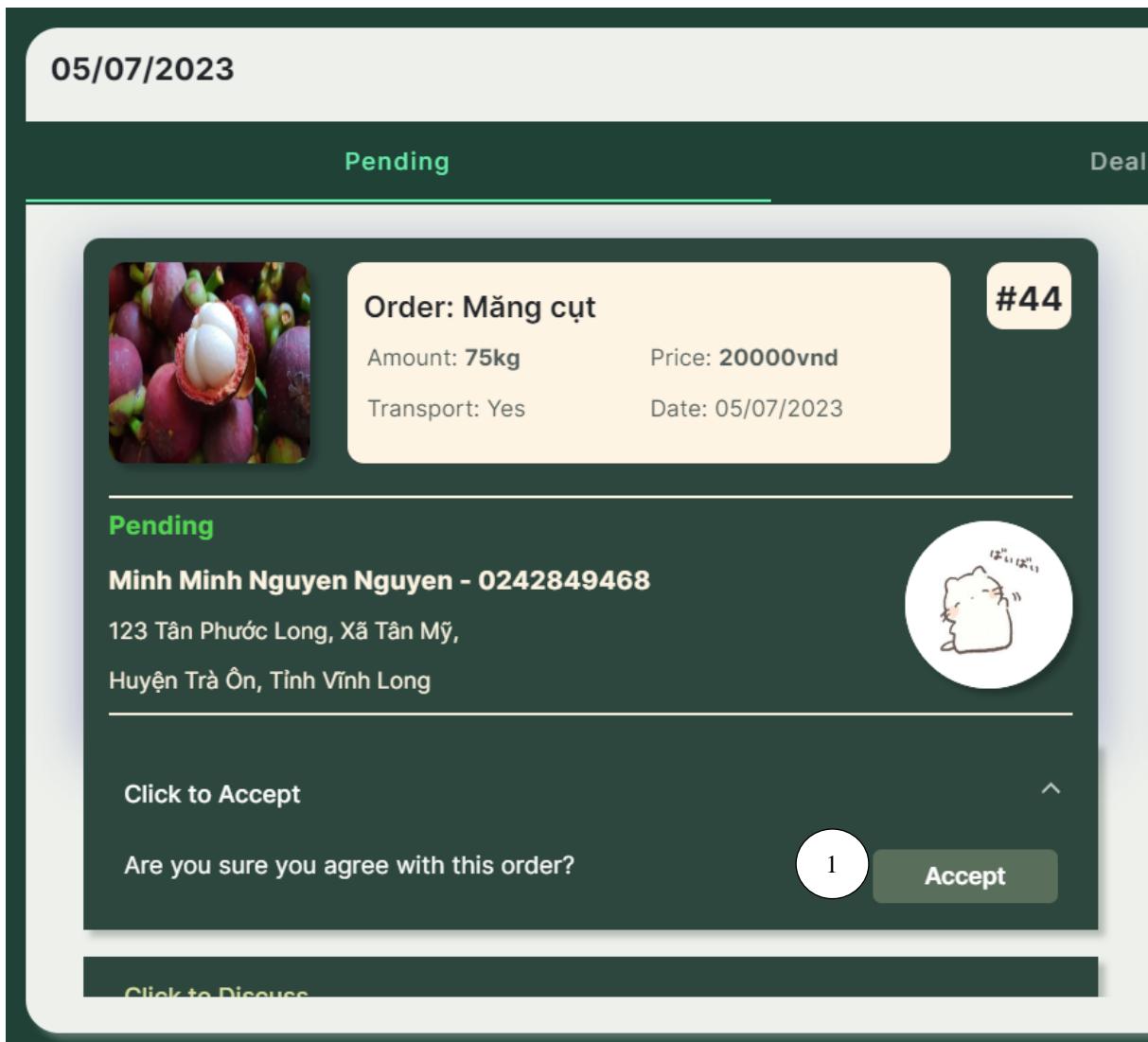


Figure 2. 45. Accept order

No	Name	Type	Event
1	Accept button	Button	After clicking, the order will be placed successfully

Table 2. 59. Accept order event

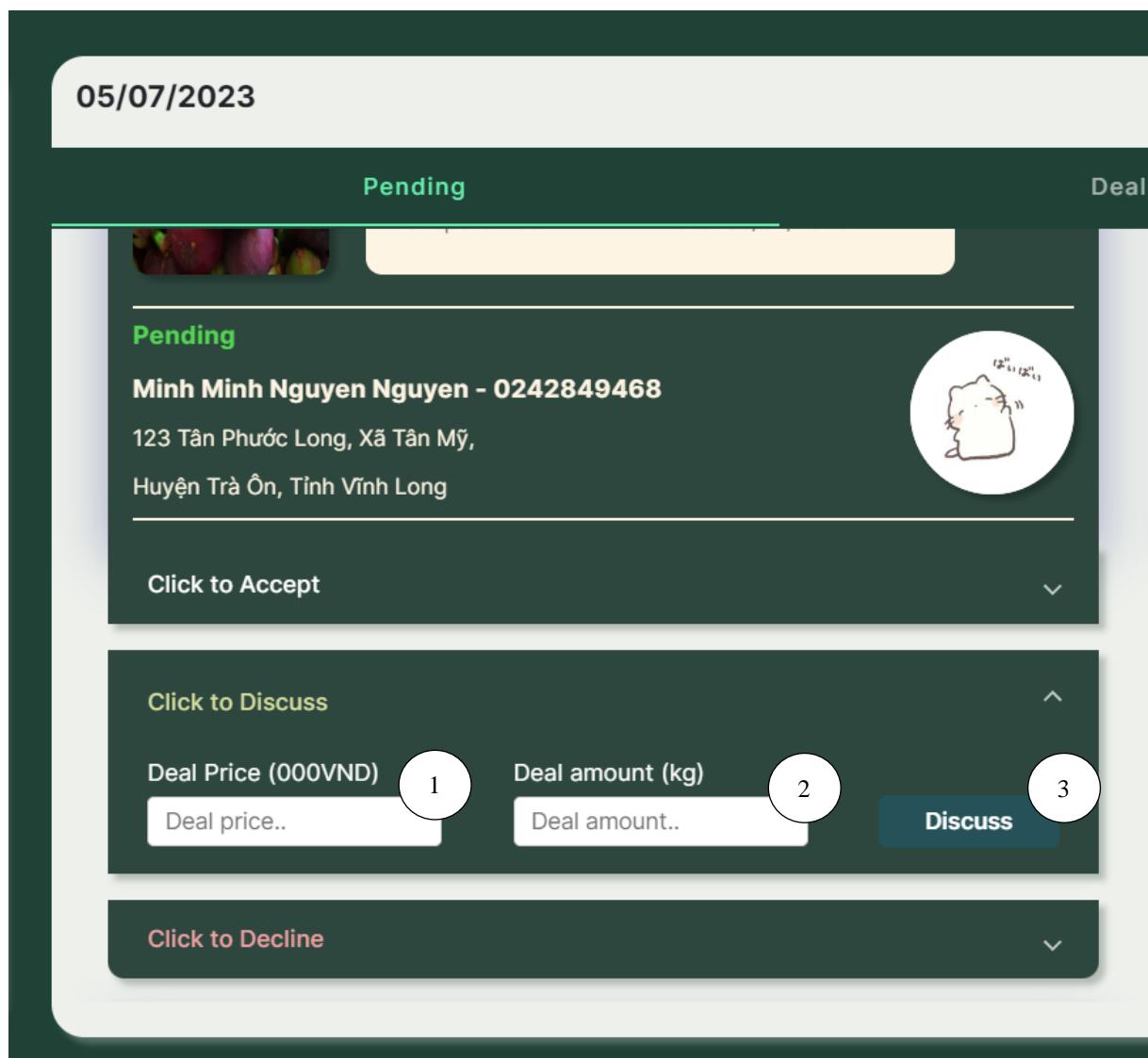


Figure 2. 46. Discuss price

No	Name	Type	Event
1	Input deal price	Input	For user input new deal price
2	Input deal amount	Input	For user input new deal amount
3	Discuss button	Button	Click to confirm new order deal

Table 2. 60. Discuss price event

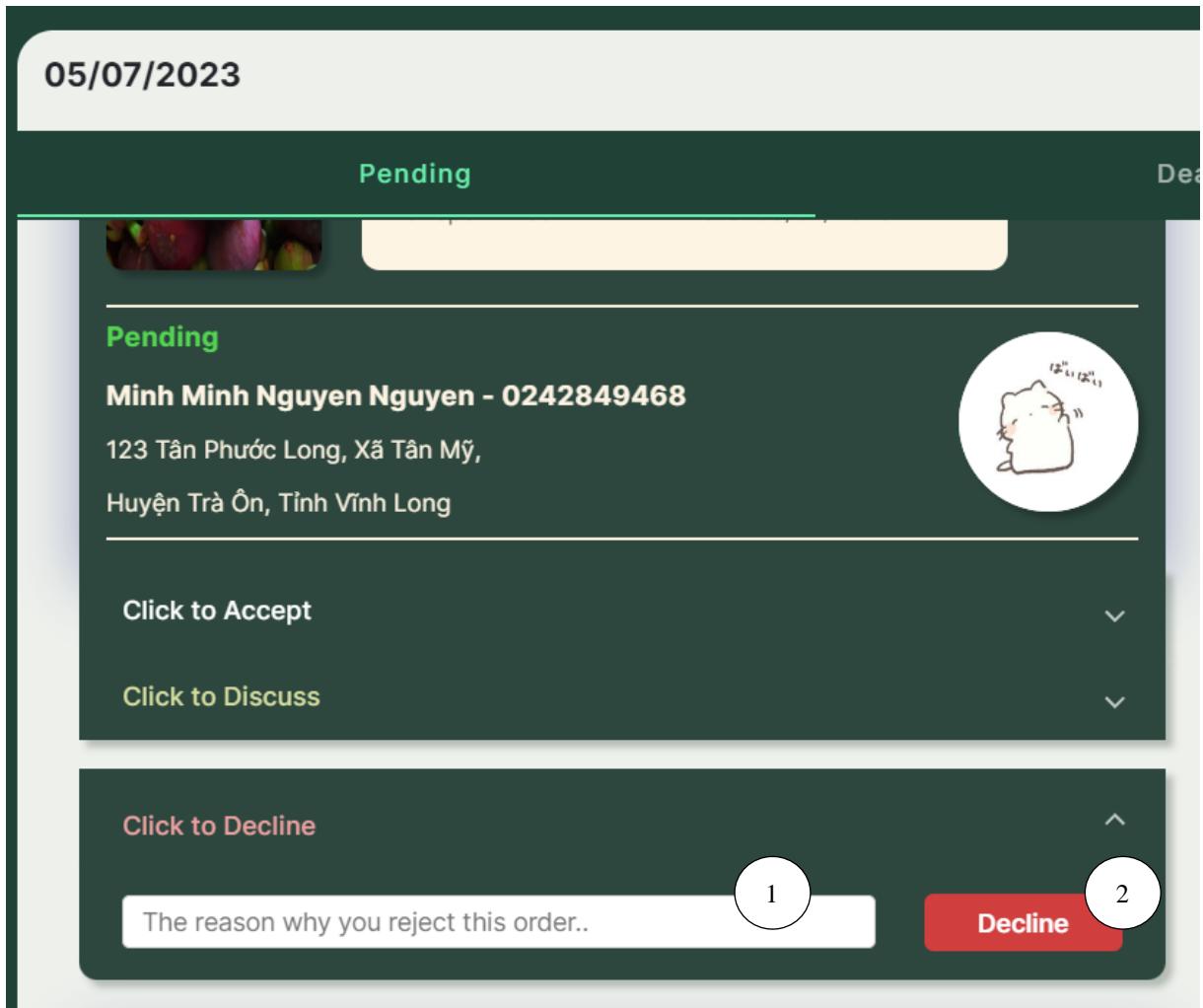


Figure 2. 47. Decline order

No	Name	Type	Event
1	Input reason	Input	For user input the reason why user reject the order
2	Decline button	Button	Click to decline the order

Table 2. 61. Decline order event

## CHAPTER 4: SYSTEM DESIGN

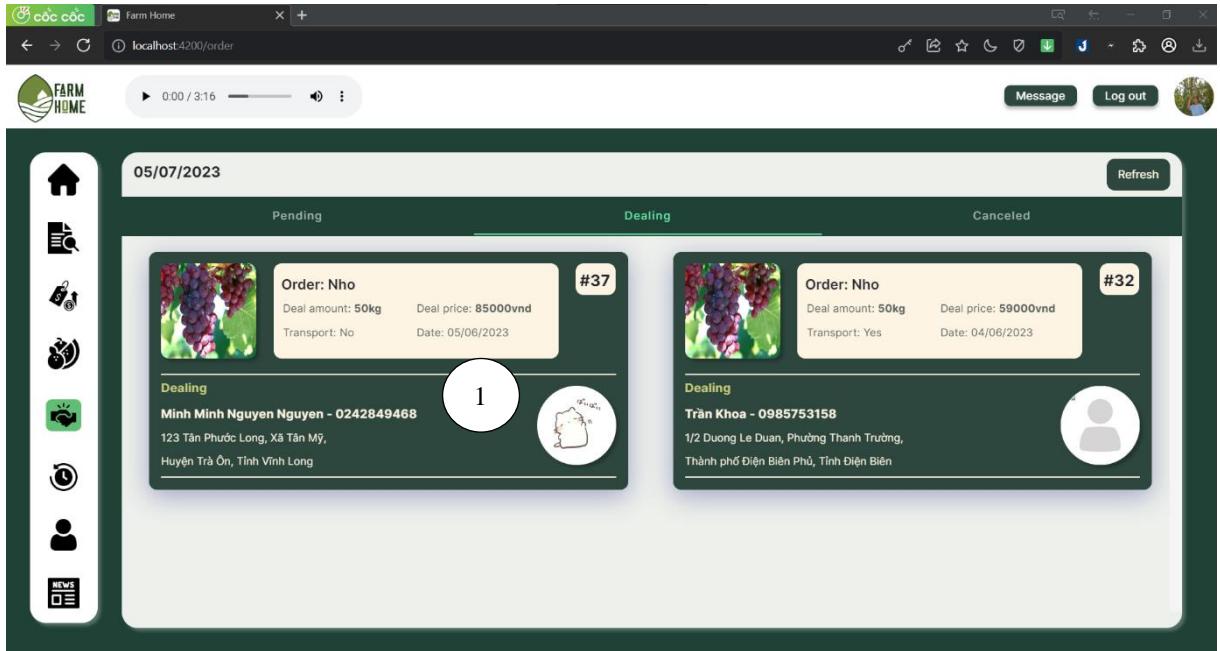


Figure 2. 48. Order with dealing status

No	Name	Type	Event
1	Dealing card	Card	Display all information of order after a deal

Table 2. 62. Order with dealing status event

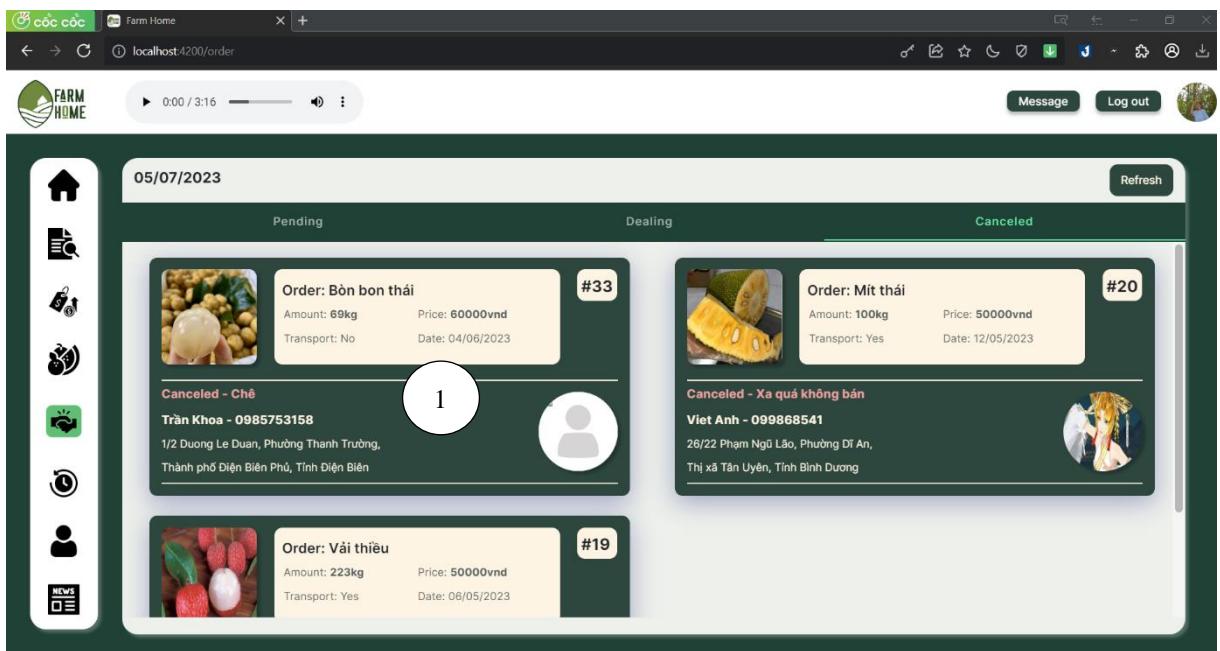


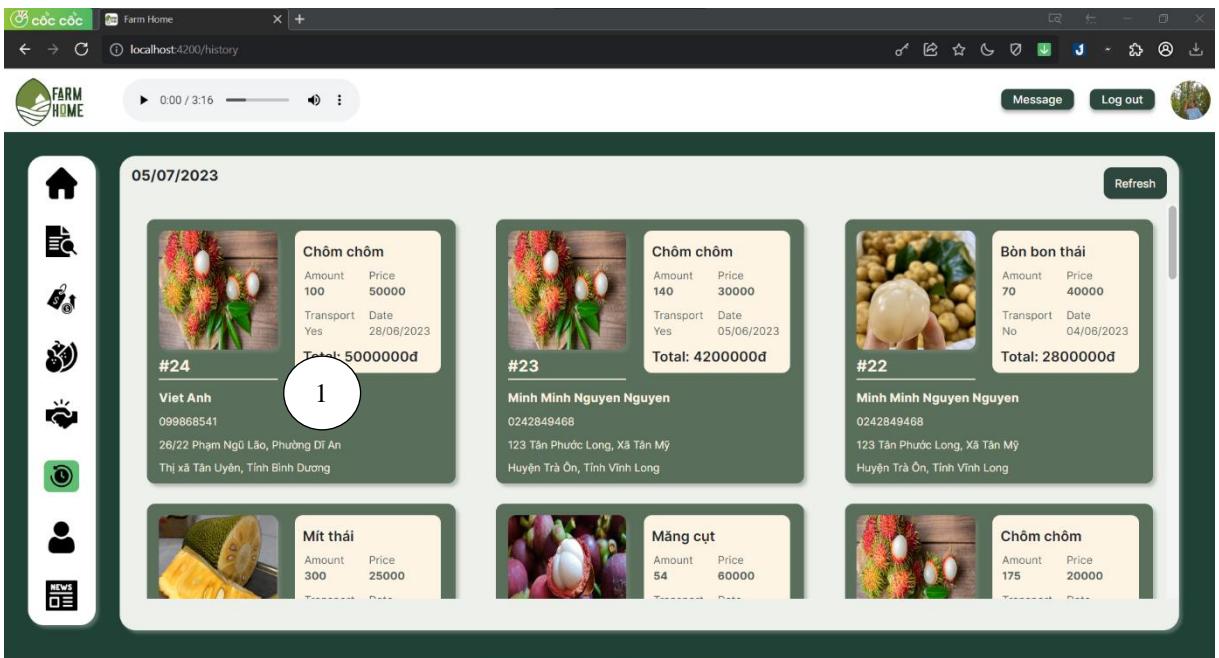
Figure 2. 49. Order with canceled status

## CHAPTER 4: SYSTEM DESIGN

---

No	Name	Type	Event
1	Decline card	Card	Display all information of order after a decline

Table 2. 63. Order with canceled status event



The screenshot shows the 'Farm Home' application interface. On the left, there is a vertical sidebar with various icons: a house, a magnifying glass, a key, a hand, a clock, a person, and a news icon. The main content area is titled 'History' and shows a grid of completed orders. Each order card includes a thumbnail image of the product, the order number, the customer's name, contact information, and delivery details. For example, order #24 is for Chôm chôm from Viet Anh, delivered to 26/22 Phạm Ngũ Lão, Phường Dĩ An, Thị xã Tân Uyên, Tỉnh Bình Dương. The total amount for this order is 5000000đ.

Figure 2. 50. History page

No	Name	Type	Event
1	History card	Card	Display information of completed orders

Table 2. 64. History page event

## CHAPTER 4: SYSTEM DESIGN

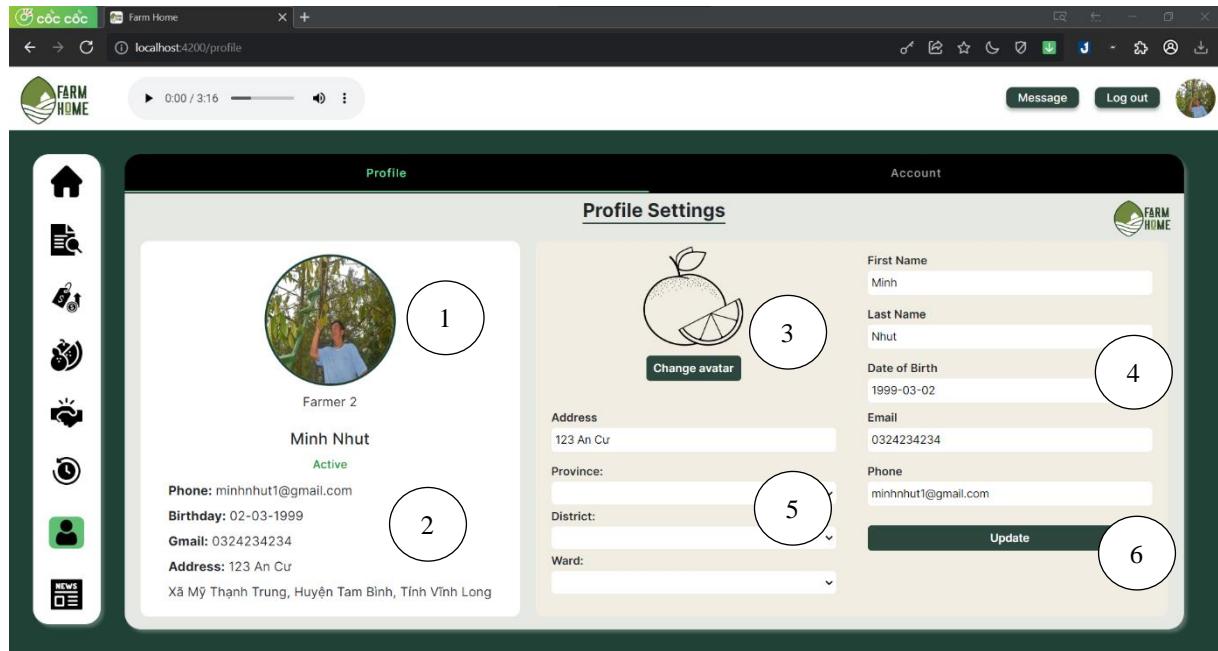


Figure 2. 51. Profile page

No	Name	Type	Event
1	User avatar	Image	Display the avatar of user
2	Personal information	Text	Display all information of user
3	Upload image	Button	Choose new image to update
4	Input personal information	Texbox	For user to input all information to update
5	Input location information	Textbox	For user to input location
6	Update button	Button	Click to update new information

Table 2. 65. Profile page event

## CHAPTER 4: SYSTEM DESIGN

---

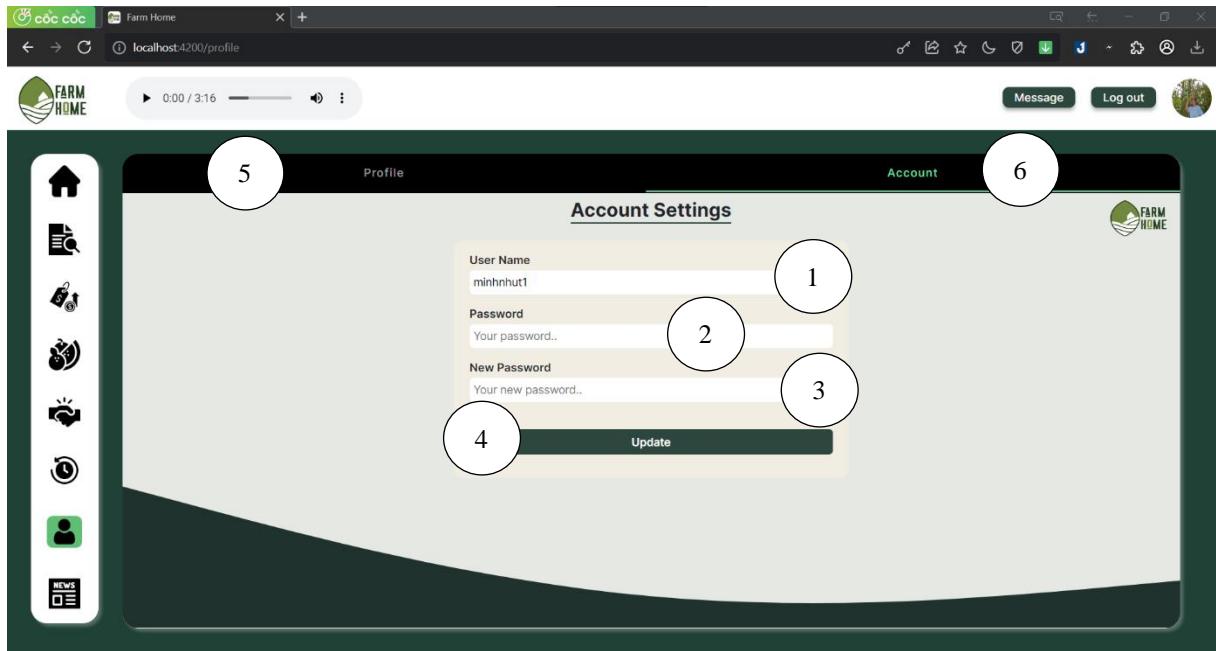


Figure 2. 52. Change password

No	Name	Type	Event
1	Input username	Textbox	For user see username (readonly)
2	Input old password	Textbox	For user input old password
3	Input new password	Textbox	For user input new password
4	Update button	Button	Confirm the action
5	Profile tab	Tab	After clicking, switch to profile panel
6	Account tab	Tab	After clicking, switch to account panel

Table 2. 66. Change password event

### 4.5.3. Website for admin

No.	Screen name	Description
1	Login	Login for admins

<b>2</b>	Statistics	Display statistical information
<b>3</b>	Merchants	Display merchants information
<b>4</b>	Farmers	Display farmers information
<b>5</b>	News	Display news information
<b>6</b>	Category	Display categories information
<b>7</b>	Reports	Display reports information
<b>8</b>	Logout	Leave the admin page

*Table 2. 67. List of website page for admin*

The current administration's website serves multiple purposes, primarily focusing on the display of statistical information and providing relevant details about users and products. The website plays a crucial role in presenting a wide range of statistical data related to the administration's performance, enabling transparency and accountability in its actions.

The website serves as a comprehensive platform for sharing information about users and products. It provides a valuable resource by showcasing details about registered user.

## CHAPTER 4: SYSTEM DESIGN

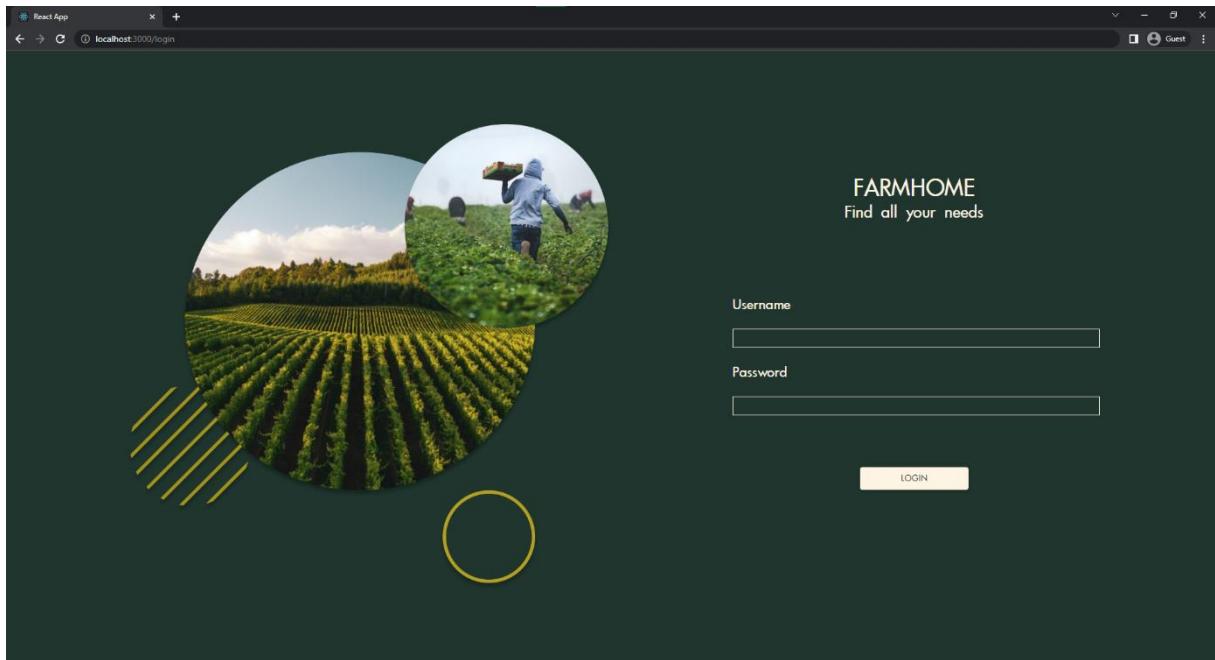


Figure 2. 53. Login Page

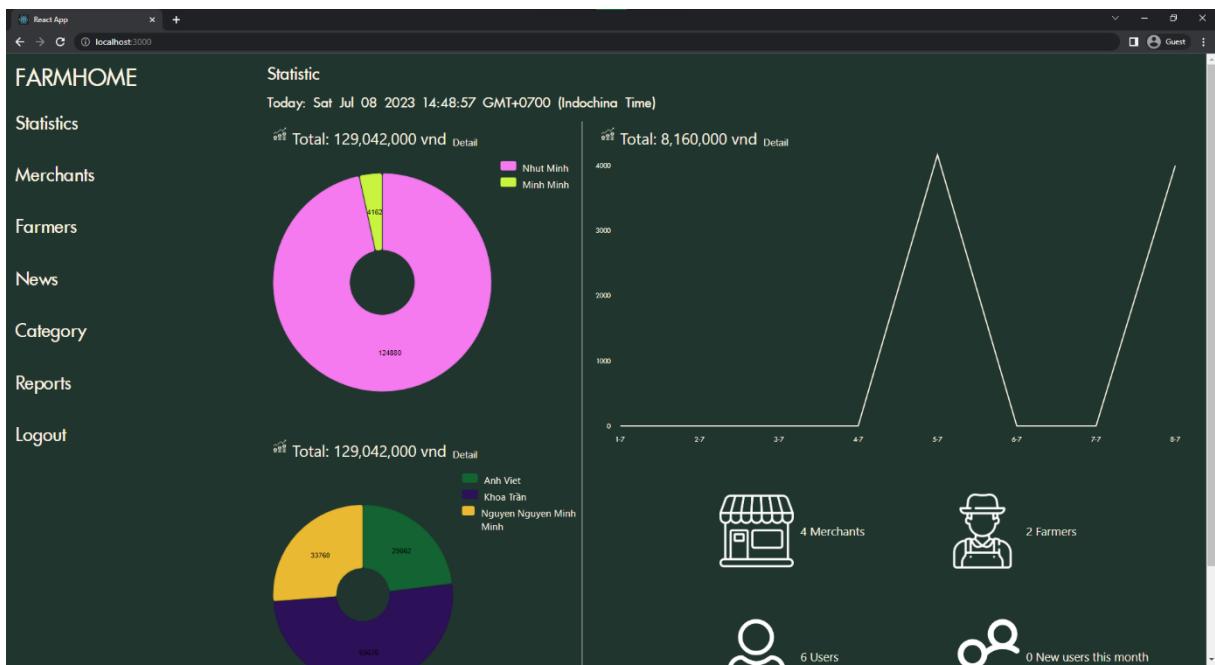
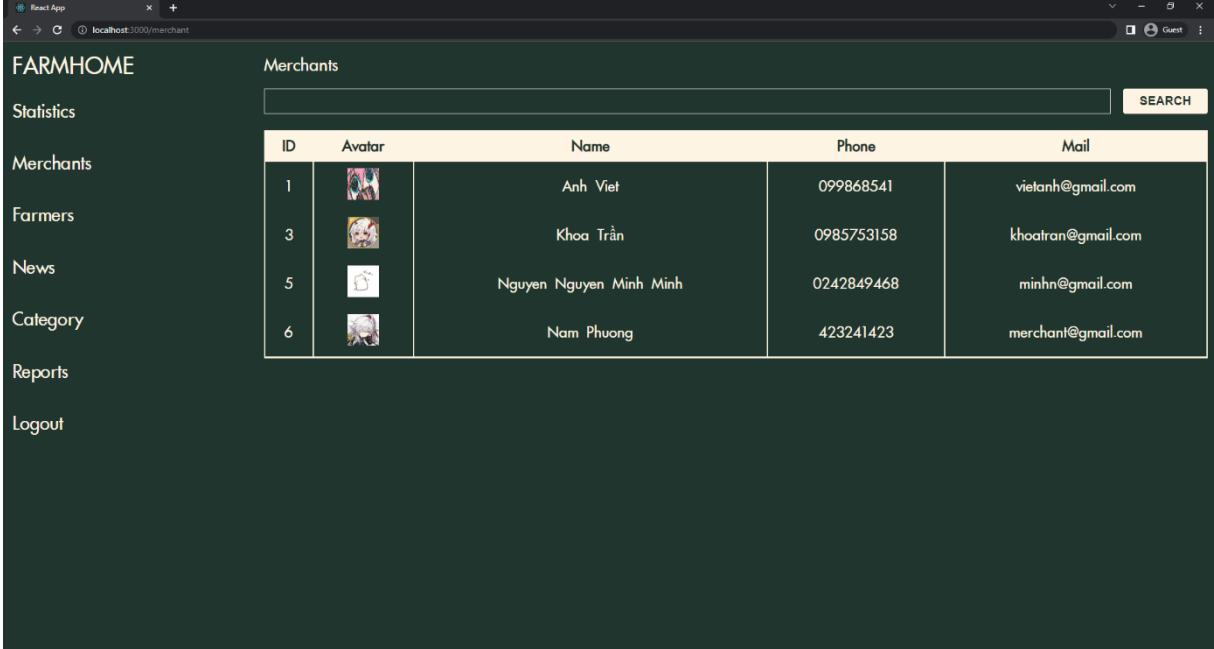


Figure 2. 54. Statistic Page

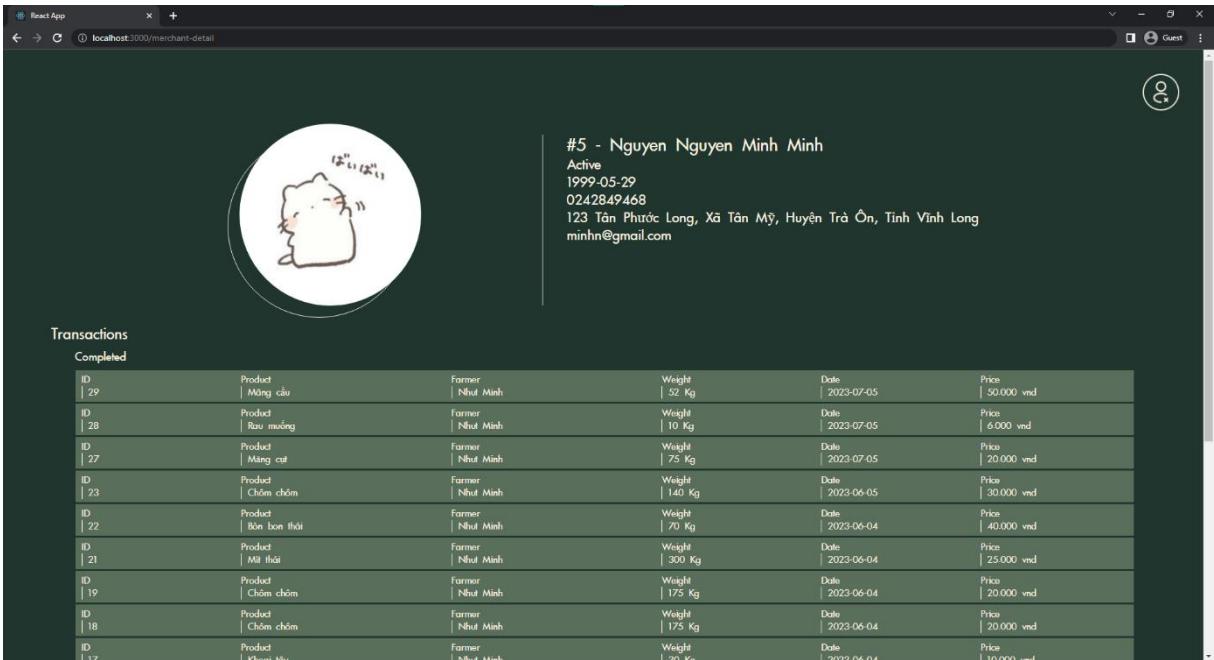
## CHAPTER 4: SYSTEM DESIGN



The screenshot shows a dark-themed web application interface. On the left, there is a sidebar with navigation links: 'Statistics', 'Merchants' (selected), 'Farmers', 'News', 'Category', and 'Reports'. Below these are 'Logout' and a guest account indicator ('Guest'). The main content area is titled 'Merchants' and contains a table with the following data:

	ID	Avatar	Name	Phone	Mail
Merchants	1		Anh Việt	099868541	vietanh@gmail.com
Farmers	3		Khoa Trần	0985753158	khoatran@gmail.com
News	5		Nguyen Nguyen Minh Minh	0242849468	minhn@gmail.com
Category	6		Nam Phuong	423241423	merchant@gmail.com

Figure 2. 55. Merchant List Page



The screenshot shows a merchant detail page for 'Nguyen Nguyen Minh Minh'. The profile picture is a cartoon character. The right side displays the merchant's information: Active, 1999-05-29, 0242849468, 123 Tân Phước Long, Xã Tân Mỹ, Huyện Trà Ôn, Tỉnh Vĩnh Long, minhn@gmail.com.

Below this, a section titled 'Transactions Completed' lists 10 completed transactions:

ID	Product	Former	Weight	Date	Price
29	Măng cụt	Nhật Minh	52 Kg	2023-07-05	50.000 vnd
28	Rau muống	Nhật Minh	10 Kg	2023-07-05	6.000 vnd
27	Măng cụt	Nhật Minh	75 Kg	2023-07-05	20.000 vnd
23	Chảm chảm	Nhật Minh	140 Kg	2023-06-05	30.000 vnd
22	Bún bò thái	Nhật Minh	70 Kg	2023-06-04	40.000 vnd
21	Mít thái	Nhật Minh	300 Kg	2023-06-04	25.000 vnd
19	Chảm chảm	Nhật Minh	175 Kg	2023-06-04	20.000 vnd
18	Chảm chảm	Nhật Minh	175 Kg	2023-06-04	20.000 vnd
17	Khoai lang	Nhật Minh	90 Kg	2023-06-04	10.000 vnd

Figure 2. 56. Merchant detail page

## CHAPTER 4: SYSTEM DESIGN

The figure consists of two screenshots of a web application interface, likely a React application, showing farmer details.

**Screenshot 1: Merchant Detail Page (localhost:3000/merchant-detail)**

This screenshot shows a merchant profile for Nguyen Ngugen Minh Minh. The profile picture is a circular drawing of a person's head. The contact information listed is:

#5 - Nguyen Ngugen Minh Minh  
Active  
1999-05-29  
0242849468  
123 Tân Phước Long, Xã Tân Mỹ, Huyện Trà Ôn, Tỉnh Vĩnh Long  
minhn@gmail.com

A "Transactions" button is visible at the bottom left.

**Screenshot 2: Farmer Detail Page (localhost:3000/farmer-detail)**

This screenshot shows a farmer profile for Nhut Minh. The profile picture is a circular photo of a person in a blue shirt standing next to a banana tree. The contact information listed is:

#2 - Nhut Minh  
Active  
1999-03-02  
minhnht@gmail.com  
123 An Cử, Xã Mỹ Thành Trung, Huyện Tam Bình, Tỉnh Vĩnh Long  
0324234234

A "Products" button is visible at the bottom left.

**Product Listings**

The page displays several product cards:

- Mango cut**: 279 kg - 2023-06-18. Description: Mango cut tươi ngon, trái to, chắc và tròn. Mango cut có bên trong là ruột trắng ngọt, mọng nước và chia thành nhiều múi.
- Chôm chôm**: 985 kg - 2023-06-18. Description: Chôm chôm vườn siêu ngọt.
- Ôi nǚ hoàng**: 296 kg - 2023-06-18. Description: Ôi từ lâu luôn nằm trong danh sách những loại trái cây được nhiều người yêu thích bởi hương vị dễ ăn, nhiều lợi ích cho sức khỏe.
- Bòn bon thái**: 630 kg - 2023-06-18. Description: Bòn bon có tác dụng duy trì lòn da sáng mịn, khỏe mạnh, thúc đẩy tiêu hóa.
- Vải thiều**: 780 kg - 2023-06-18. Description: Vải thiều là một loại trái cây rất quen thuộc với chúng ta, bền,甜美, ngọt ngào và phái thường.
- Nho**: 400 kg - 2023-06-18. Description: Nho là loại trái cây quả mọng, được thu hoạch từ cây thân leo. Nho mọc thành từng chùm, kích thước trung bình.

Figure 2. 57. Farmer detail page

## CHAPTER 4: SYSTEM DESIGN

**Transactions**

Completed

ID	Product	Merchant	Weight	Date	Price
29	Măng cụt	Nguyen Nguyen Minh Minh	52 Kg	2023-07-05	50.000 vnd
28	Khoai mì	Nguyen Nguyen Minh Minh	10 Kg	2023-07-05	6.000 vnd
27	Măng cụt	Nguyen Nguyen Minh Minh	75 Kg	2023-07-05	20.000 vnd
24	Chôm chôm	Ash Viel	100 Kg	2023-06-28	50.000 vnd
23	Chôm chôm	Nguyen Nguyen Minh Minh	140 Kg	2023-06-05	30.000 vnd
22	Bòn bon thái	Nguyen Nguyen Minh Minh	70 Kg	2023-06-04	40.000 vnd
21	Mít thái	Nguyen Nguyen Minh Minh	300 Kg	2023-06-04	25.000 vnd
20	Măng cụt	Khoa Trần	54 Kg	2023-06-04	60.000 vnd
19	Chôm chôm	Nguyen Nguyen Minh Minh	175 Kg	2023-06-04	20.000 vnd
18	Chôm chôm	Nguyen Nguyen Minh Minh	175 Kg	2023-06-04	20.000 vnd
17	Khoai idy	Nguyen Nguyen Minh Minh	30 Kg	2023-06-04	10.000 vnd
16	Khoai idy	Nguyen Nguyen Minh Minh	30 Kg	2023-06-04	10.000 vnd
15	Măng cụt	Khoa Trần	95 Kg	2023-06-03	50.000 vnd
14	Vải thiều	Khoa Trần	10 Kg	2023-06-03	70.000 vnd

#2 - Nhung Minh

Active

1999-03-02  
minhnhu1@gmail.com  
123 An Cử, Xã Mỹ Thành Trung, Huyện Tam Bình, Tỉnh Vĩnh Long  
0324234234

**Products**

**Transactions**

**Statistics**

## CHAPTER 4: SYSTEM DESIGN

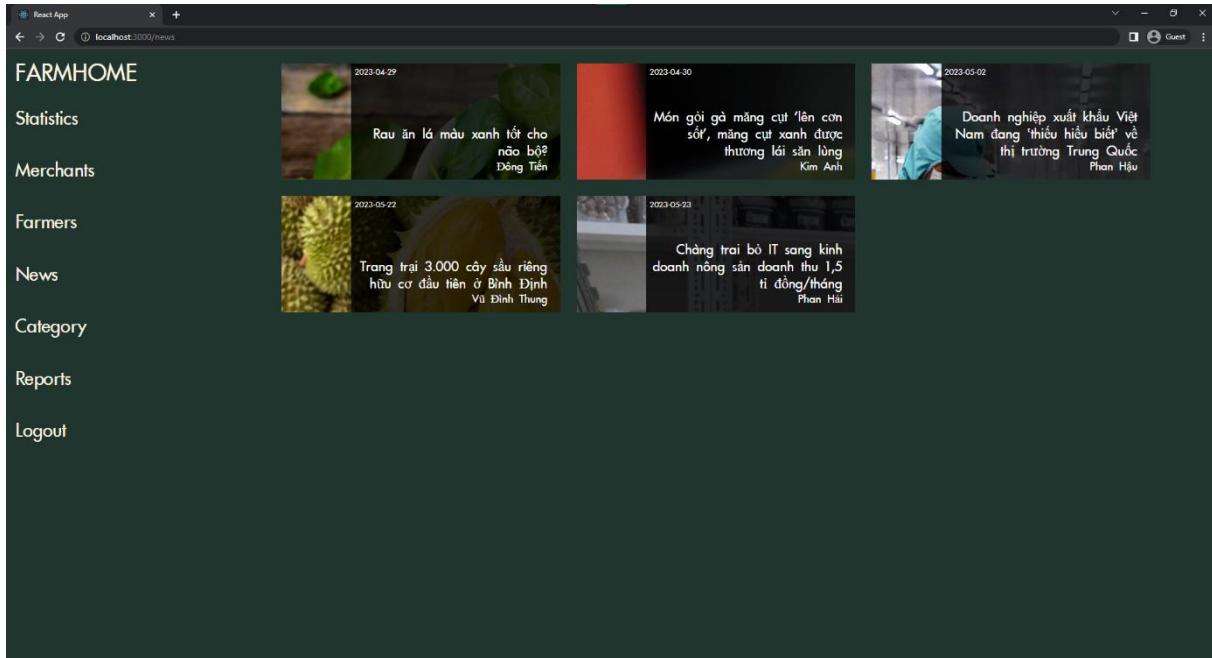


Figure 2. 58. News list page

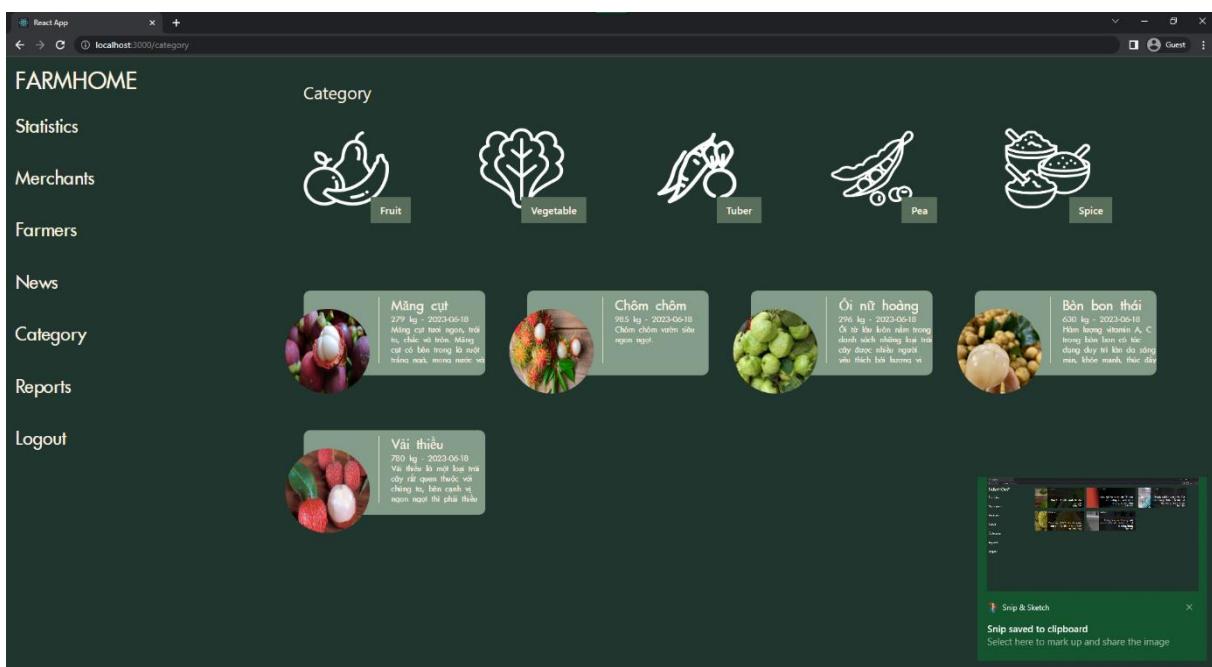


Figure 2. 59. Category list page

## CHAPTER 4: SYSTEM DESIGN

The image consists of two screenshots of a web application named "FARMHOME".

**Top Screenshot (Category Page):**

- Header: "React App" and "localhost:3000/category".
- Left sidebar menu:
  - Statistics
  - Merchants
  - Farmers
  - News
  - Category
  - Reports
  - Logout
- Main content area:
  - Category icons: Fruit (apple), Vegetable (lettuce), Tuber (potato), Pea (pea pod), Spice (rice bowls).
  - A modal window showing a thumbnail of green onions and the text: "Rau muống 100g - 2023-07-08 Rau xanh ngọt, thơm, không sâu nấm."

**Bottom Screenshot (Reports Page):**

- Header: "React App" and "localhost:3000/reports".
- Left sidebar menu:
  - Statistics
  - Merchants
  - Farmers
  - News
  - Category
  - Reports
  - Logout
- Main content area:

	Reports	CHOOSE DATE
Merchants	Delivery late	2023-07-08
Farmers	Scam	2023-07-08

Figure 2. 60. Report list page

## CHAPTER 4: SYSTEM DESIGN

The figure consists of two screenshots of a web application named FARMHOME.

**Screenshot 1: Reports**

This screenshot shows two reports listed on the right side of the dashboard:

- Delivery late** (2023-07-08):
  - From: Viet Anh
  - Username: vietanh
  - To: Minh Nhut
  - Username: minhnhat1

Content: Minh nhut fooled me to buy at higher price and delivery late
- Scam** (2023-07-08):
  - From: Viet Anh
  - Username: vietanh
  - To: Minh Minh
  - Username: minhminh

Content: Minh Minh is a scam

**Screenshot 2: Statistic**

This screenshot shows a statistic chart and a table:

- Statistic**: A donut chart showing values for two categories:
  - Nhut Minh: 9160
  - Minh Minh: 4162
- Table**:

Name	Value
Nhut Minh	9160
Minh Minh	4162
- User Metrics**:
  - 6 Users
  - 0 New users this month

Figure 2. 61. Statistic detail

## CHAPTER 4: SYSTEM DESIGN

React App | localhost:3000 | Guest

### FARMHOME

Statistic  
Today: Sat Jul 08 2023 14:52:29 GMT+0700 (Indochina Time)

From: July 1st, 2023 To: July 8th, 2023

Statistics Merchants Farmers News Category Reports Logout

**CHOOSE DATE**

**July 2023**

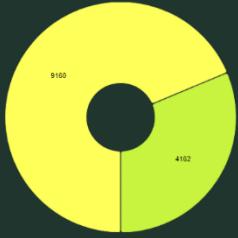
SU	MO	TU	WE	TH	FR	SA
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

**FIND** **APPLY**

**Name Value**

Name	Value
Nhut Minh	9160
Minh Minh	4162

**6 Users** **0 New users this month**



2023

0-7

React App | localhost:3000 | Guest

### FARMHOME

Statistic  
Today: Sat Jul 08 2023 14:52:29 GMT+0700 (Indochina Time)

From: June 24th, 2023 To: July 8th, 2023

Statistics Merchants Farmers News Category Reports Logout

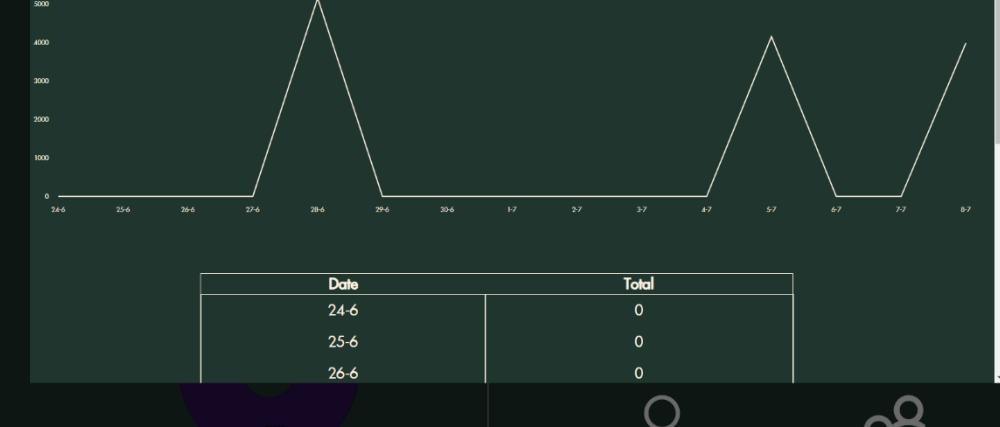
**CHOOSE DATE**

**Find**

**Date Total**

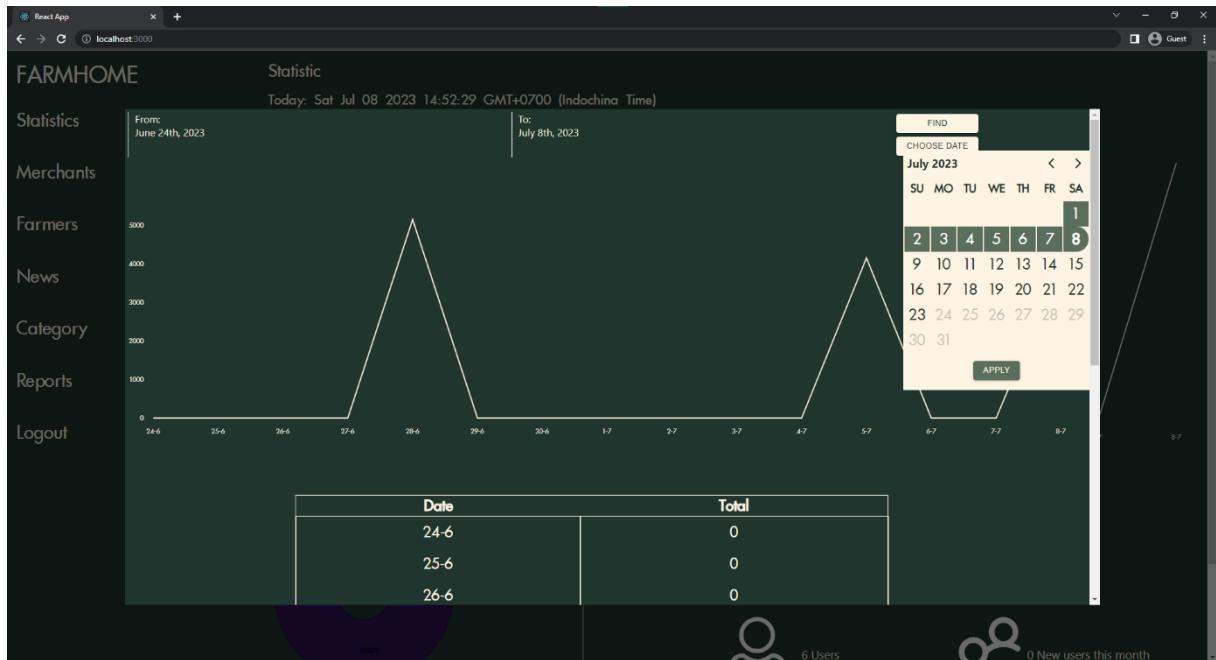
Date	Total
24-6	0
25-6	0
26-6	0

**6 Users** **0 New users this month**



## CHAPTER 4: SYSTEM DESIGN

---



## CHAPTER 5: IMPLEMENTATION AND TESTING

### 5.1. Overall architecture of the system

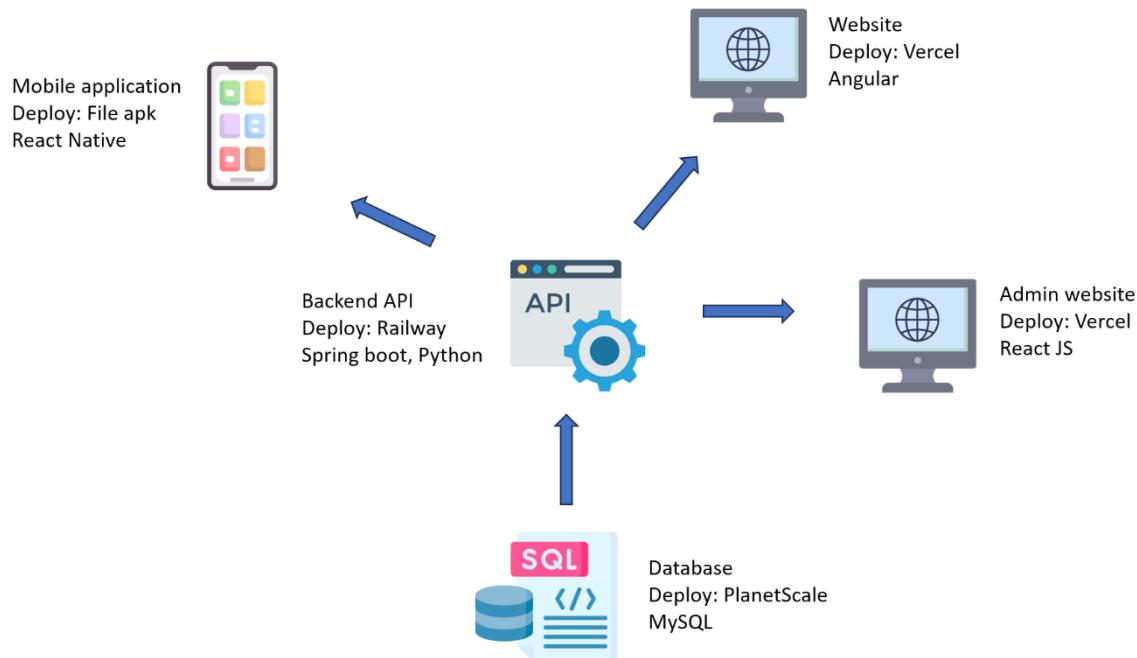


Figure 3. 1. Architecture of the system

Firstly, the system will consist of a website dedicated to farmers to post information about agricultural products, carry out product transactions, and manage their orders. They can add their products to their store to be displayed on the merchant's application.

Meanwhile, on the merchant's mobile application, they can log in (or register) to access the marketplace and view the products posted by farmers and engage in buying and selling transactions if needed. Additionally, the application also supports some features to enhance the user experience, such as language switching, reading news, or receiving notifications related to their orders.

Furthermore, the system includes an admin management page. The main purpose of the website is to monitor, evaluate, and track user data, orders, and products on the marketplace by displaying statistical charts and information tables. The admin also has the authority to block users if any suspicious or invalid actions or transactions are detected. The admin page can also view reports from users to identify difficulties or disputes they may be facing.

Moreover, to ensure effective communication between all components of the system, an API management system has been developed. This system provides routes to perform various functionalities and utilizes connections to write or retrieve information from the database. As for the database, it is provided by a specialized website for online database rentals, ensuring data backup synchronization and access from anywhere.

## 5.2. Tensorflow

Dataset source: Kaggle – Fruits and Vegetables Image Recognition Dataset

<https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition>

Use Shallow Neural Networks

Number of pictures: 3193

Number of different labels: 36

Labels: ['grapes' 'pomegranate' 'banana' 'carrot' 'garlic' 'onion' 'pineapple' 'orange' 'capsicum' 'jalepeno' 'paprika' 'watermelon' 'raddish' 'lettuce' 'spinach' 'tomato' 'sweetpotato' 'cauliflower' 'bell pepper' 'peas' 'lemon' 'kiwi' 'chilli pepper' 'cabbage' 'turnip' 'eggplant' 'potato' 'soy beans' 'pear' 'mango' 'beetroot' 'sweetcorn' 'cucumber' 'corn' 'apple' 'ginger']

Data processing:

- Use tf.keras preprocessing.image.ImageDataGenerator with config:

```
dataframe=train_df,  
x_col='Filepath',  
y_col='Label',  
target_size=(224, 224),  
color_mode='rgb',  
class_mode='categorical',  
batch_size=32,  
shuffle=True,  
seed=0,  
rotation_range=30,  
zoom_range=0.15,  
width_shift_range=0.2,  
height_shift_range=0.2,  
shear_range=0.15,  
horizontal_flip=True,  
fill_mode="nearest"
```

Model training:

- 2 layers of Dense with 128 units and Rectified Linear Unit activation

- output layer: 36 units and softmax activation

- model compile:

- optimizer='adam',

- loss='categorical\_crossentropy',

- metrics=['accuracy']

- model train:

- Batch\_size = 32

- Epochs = 5

- EarlyStopping(monitor="val\_loss", patience=2, restore\_best\_weights=True)

- Accuracy: 0.9417

- loss: 0.1790

- val\_accuracy: 0.9521

- val\_loss: 0.1094

### 5.3. Testing

We will start with the website for farmers, login if you already have an account, or proceed to register if you don't.

## CHAPTER 5: IMPLEMENTATION AND TESTING

---

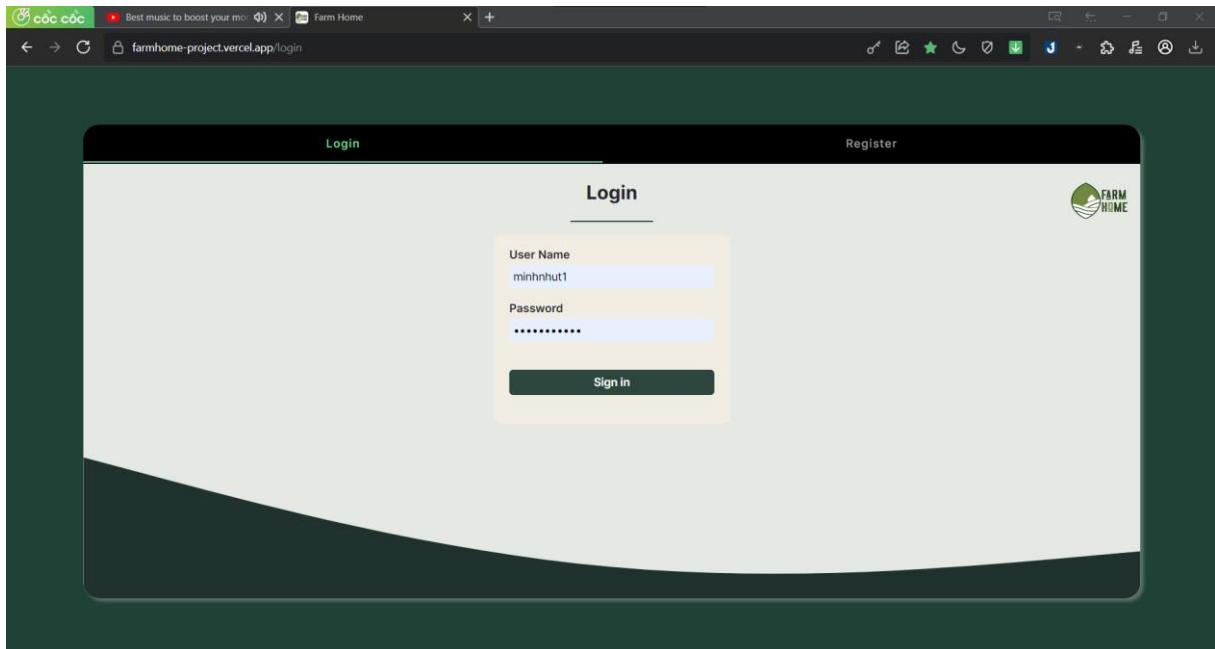


Figure 3. 2. Login page

After logging in, you will be on the home screen, where you can access the product page by clicking the "Let start" button or using the icon in the sidebar to navigate to the products.

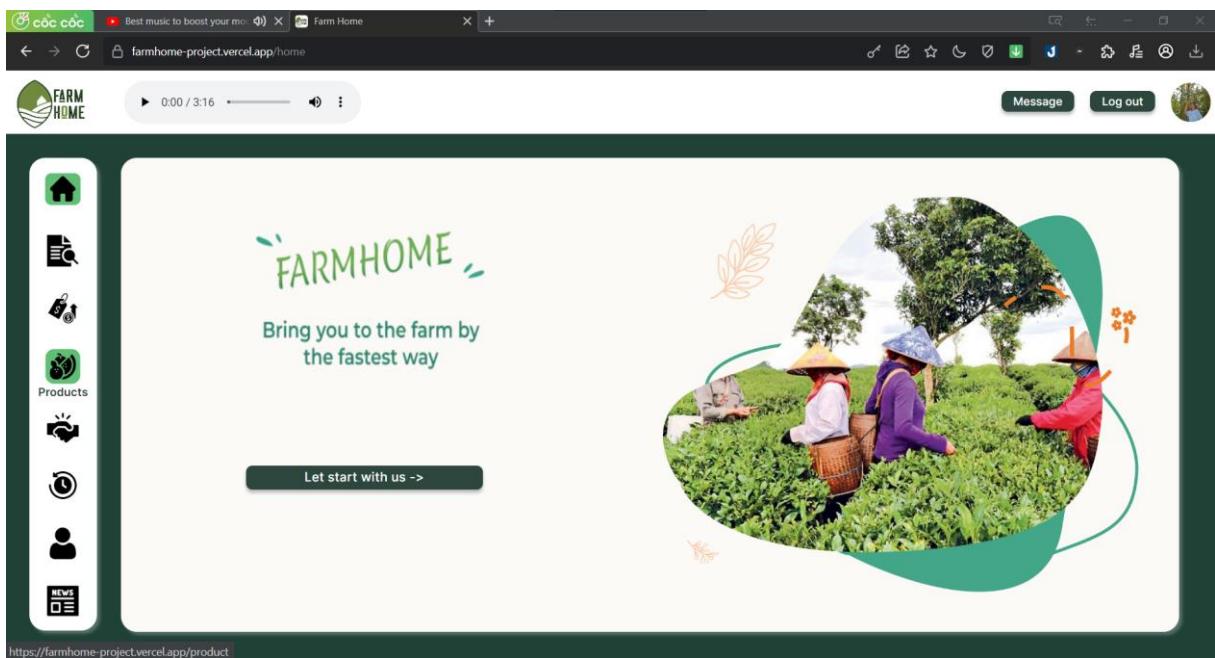


Figure 3. 3. Home page

## CHAPTER 5: IMPLEMENTATION AND TESTING

Here, we can perform functions such as adding, editing, and deleting products.

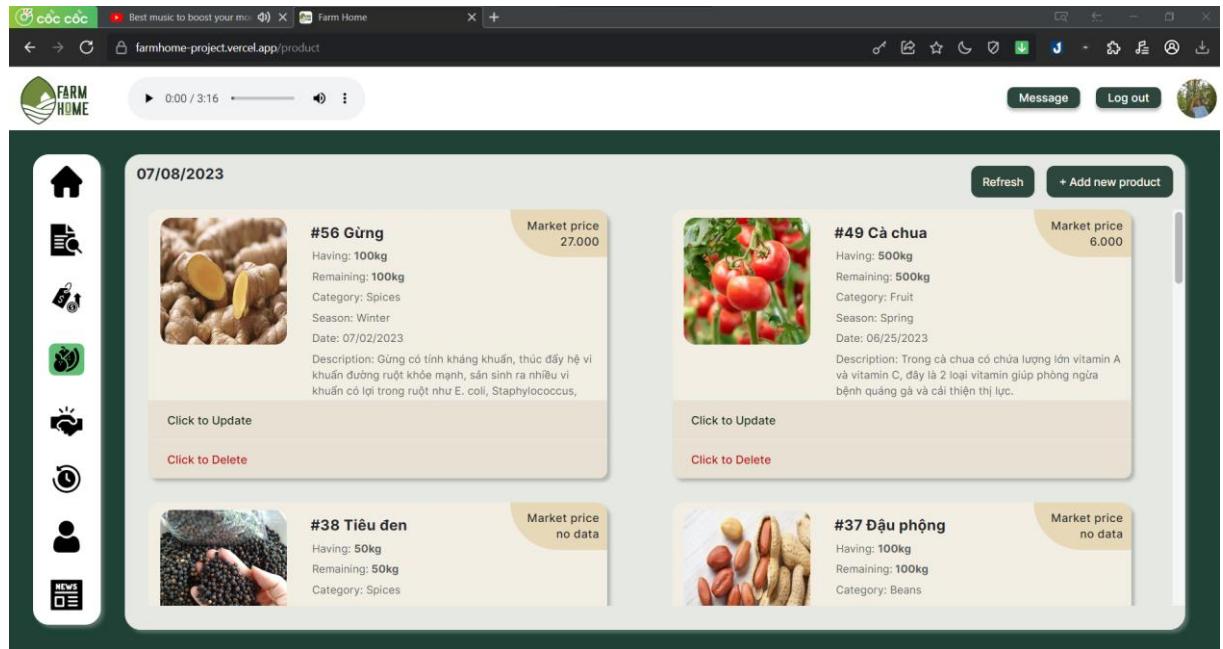


Figure 3. 4. Product page

Add product.

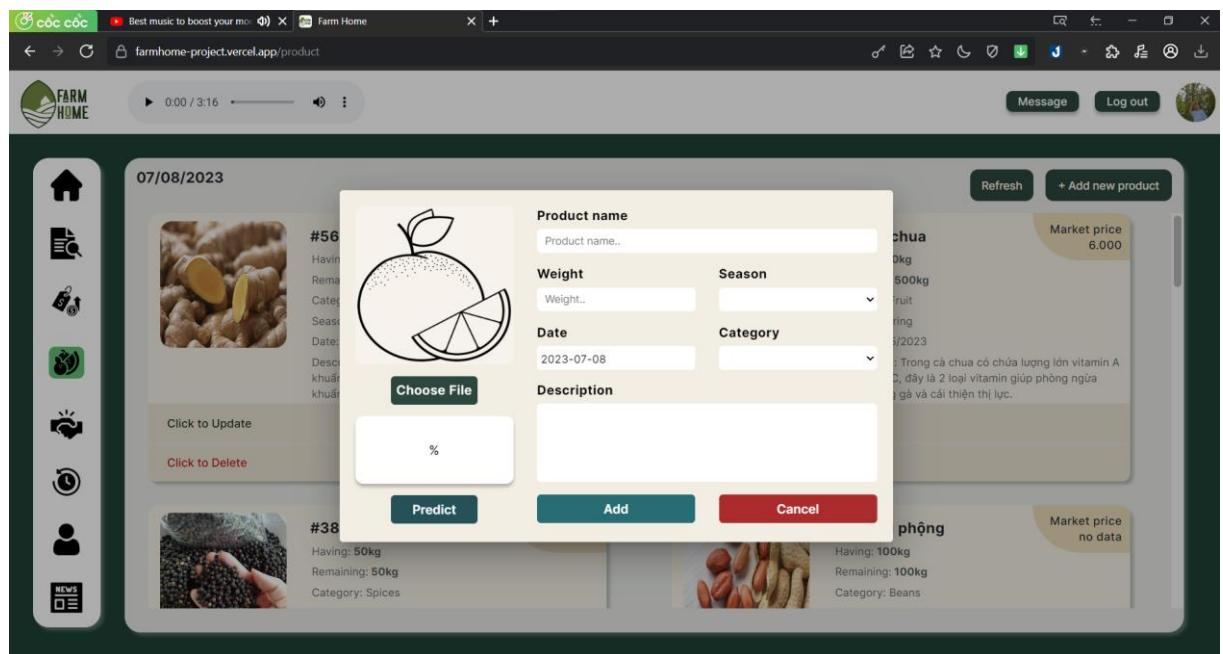


Figure 3. 5. Add new product

After the dialog appears, we can add a new product by uploading an image and filling in all the required fields. Additionally, the system supports user assistance in

## CHAPTER 5: IMPLEMENTATION AND TESTING

detecting the product using a pre-trained model, which automatically fills in the fields such as name, season, and type with a determined level of accuracy.

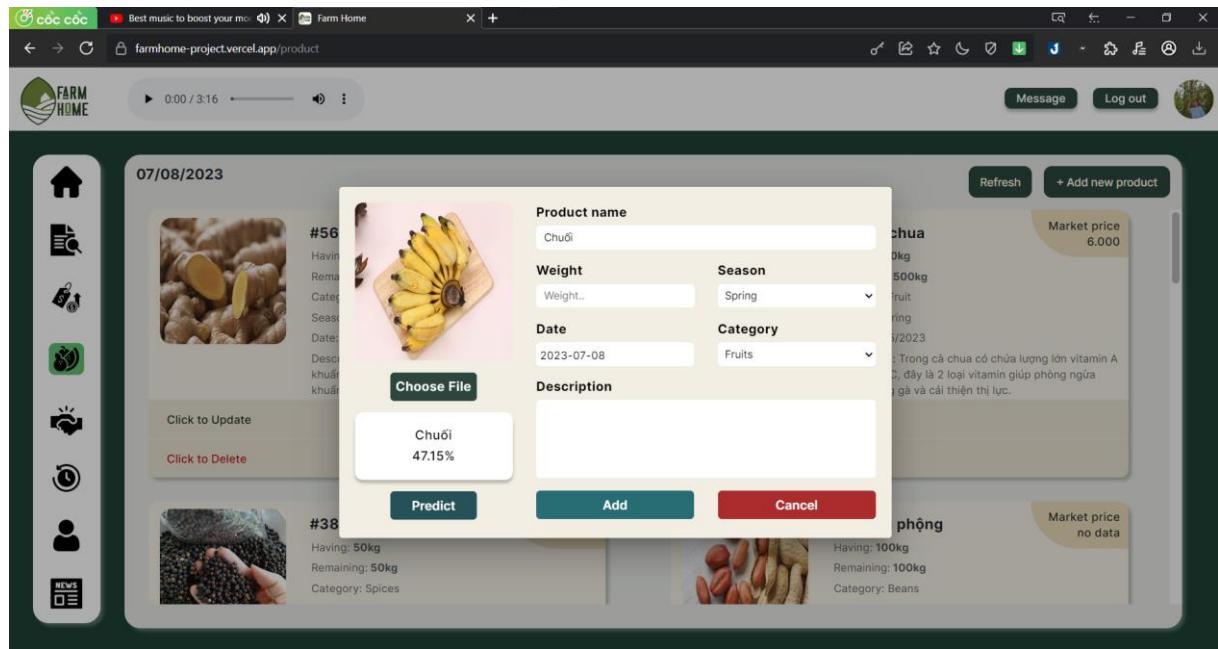


Figure 3. 6. Detect product

New product “Chuối xiêm” is added.

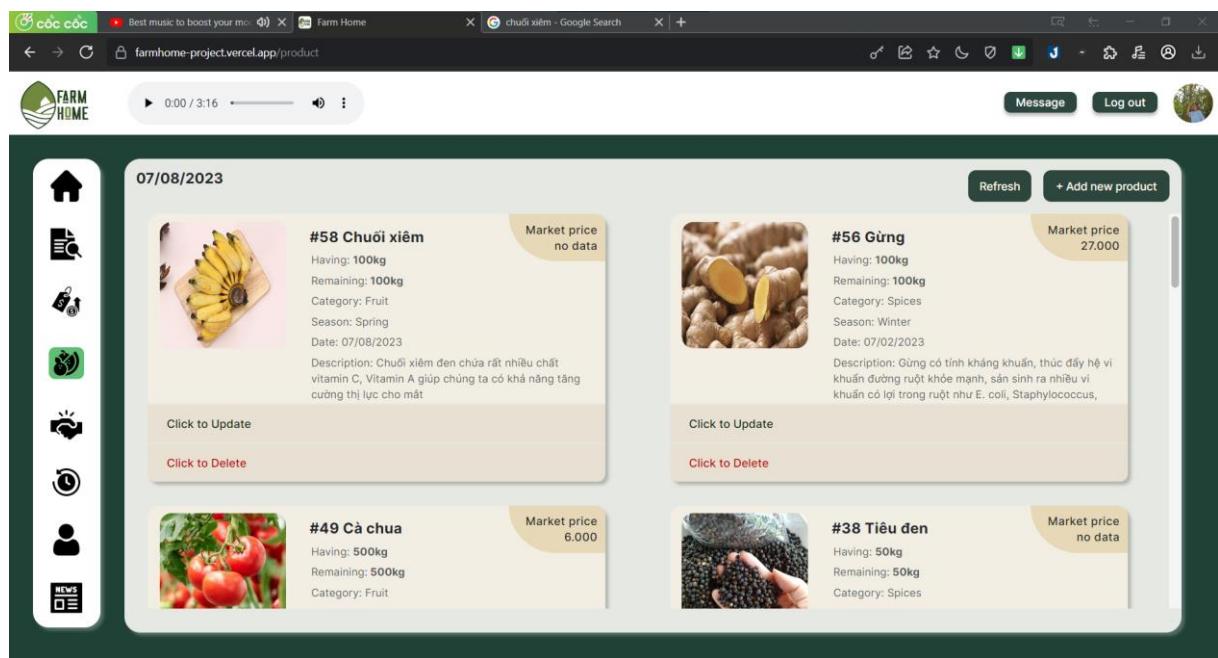


Figure 3. 7. New product is added

Edit.

Click to Update

**Change Image**

**Product name**

Chuối xiêm

**Weight**

600

**Season**

Spring

**Date**

08-07-2023

**Category**

Fruits

**Description**

Chuối xiêm đen chứa rất nhiều chất vitamin C, Vitamin A.

G

**Update**

Figure 3. 8. Edit product

Delete product.

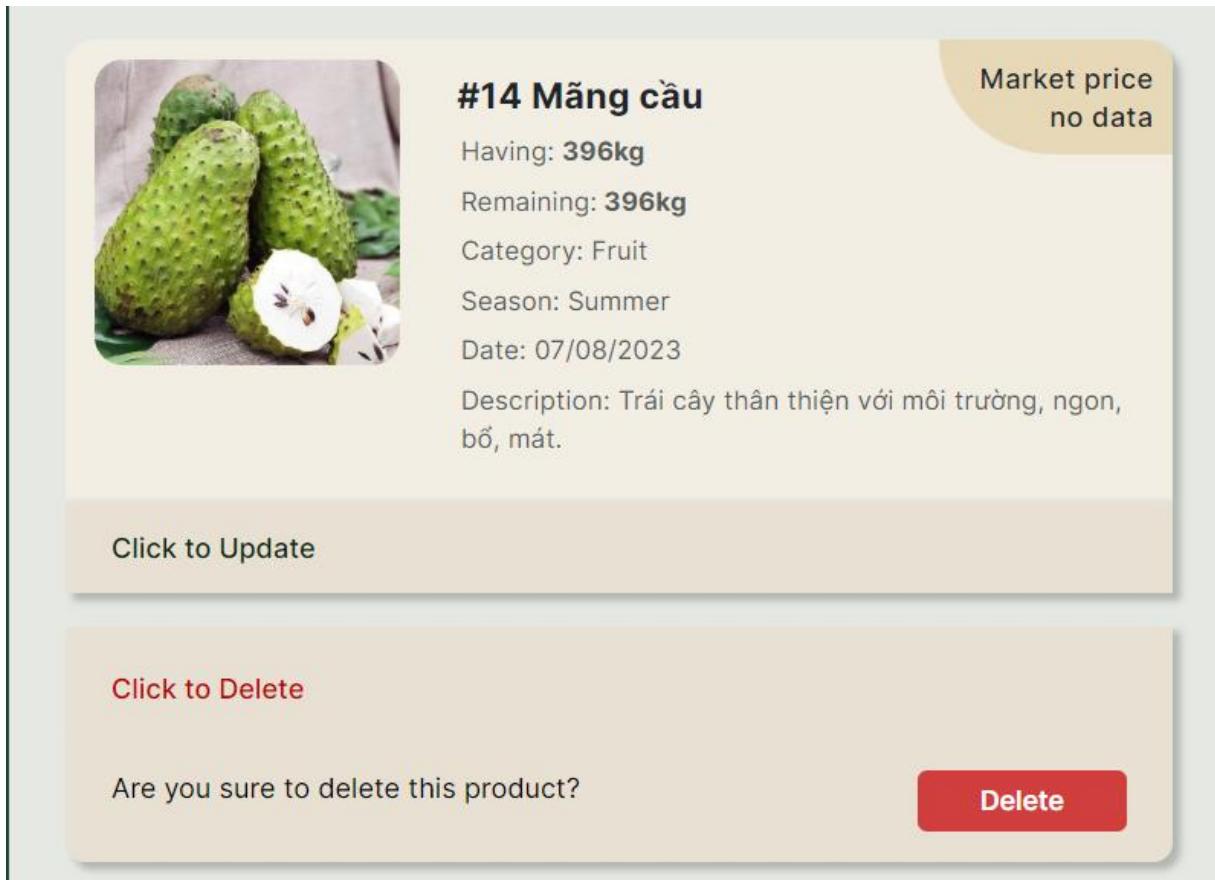


Figure 3. 9. Delete product

Now, let's switch to the mobile app in the role of a merchant.

Login.



Figure 3. 10. Login screen

Arriving at the product display screen.



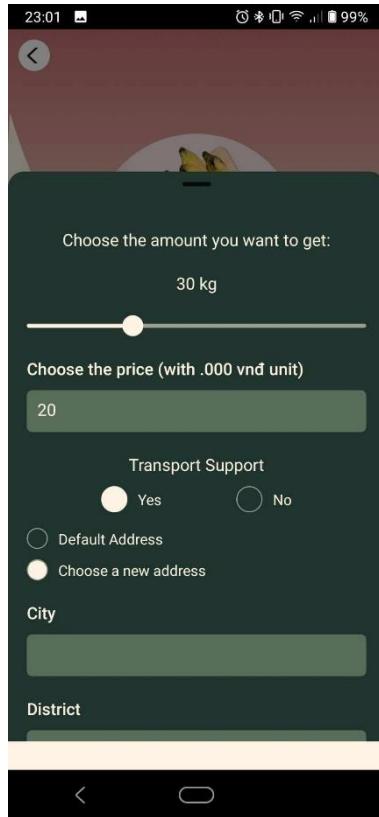
Figure 3. 11. Explore screen

Choose product.



Figure 3. 12. Product details screen

Order.



*Figure 3. 13. Order*

If you need transport assistance, there will be an option to select the address.

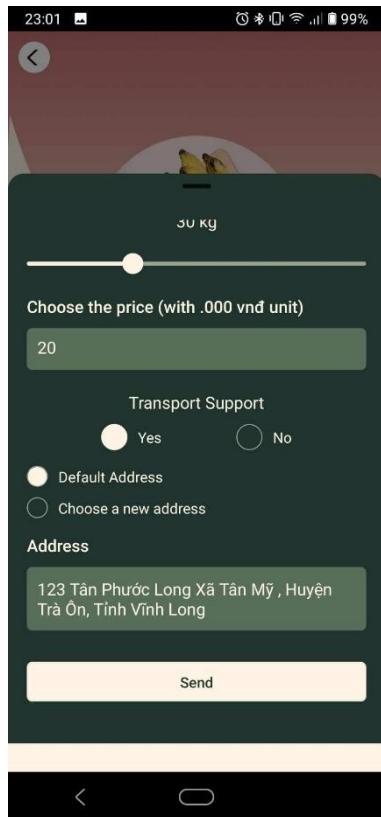


Figure 3. 14. Choose location

Navigate to the screen to view the order.

## CHAPTER 5: IMPLEMENTATION AND TESTING

---

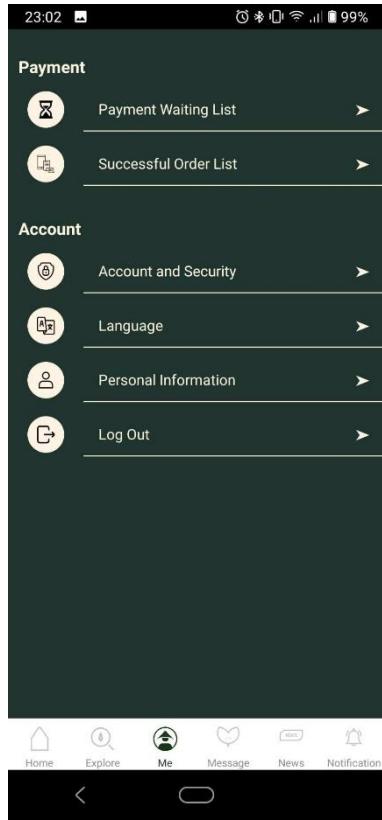


Figure 3. 15. Go to payment waiting list



Figure 3. 16. Payment waiting list

## CHAPTER 5: IMPLEMENTATION AND TESTING

---

Returning to the role of a farmer, let's navigate to the orders page.

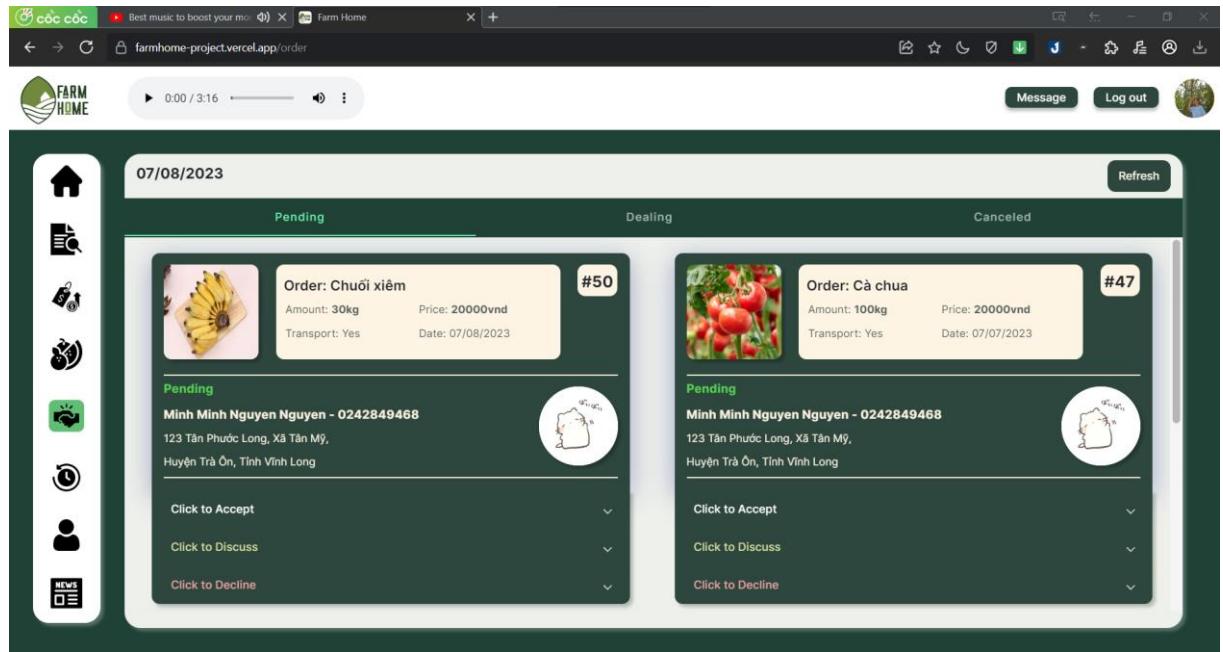


Figure 3. 17. Order page

Accept order.

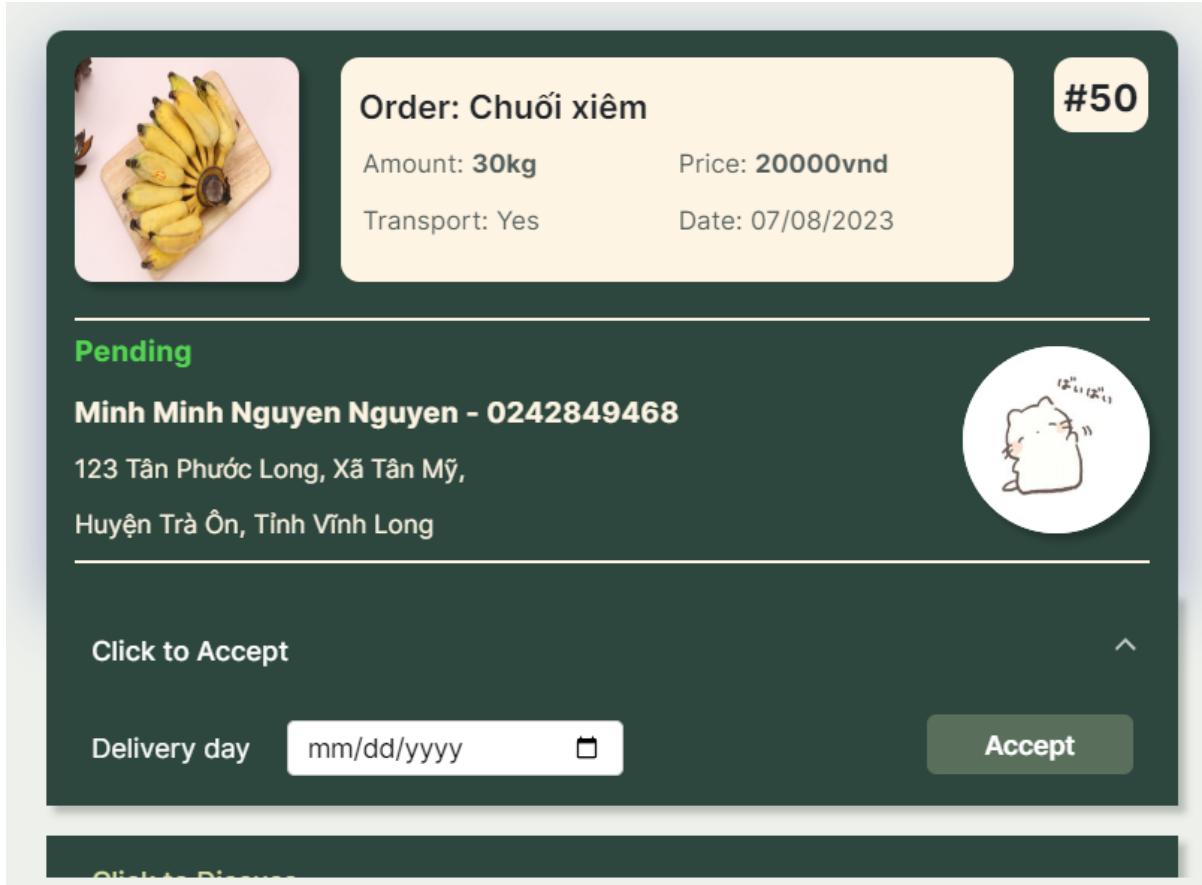


Figure 3. 18. Accept order

Negotiating about the new price.

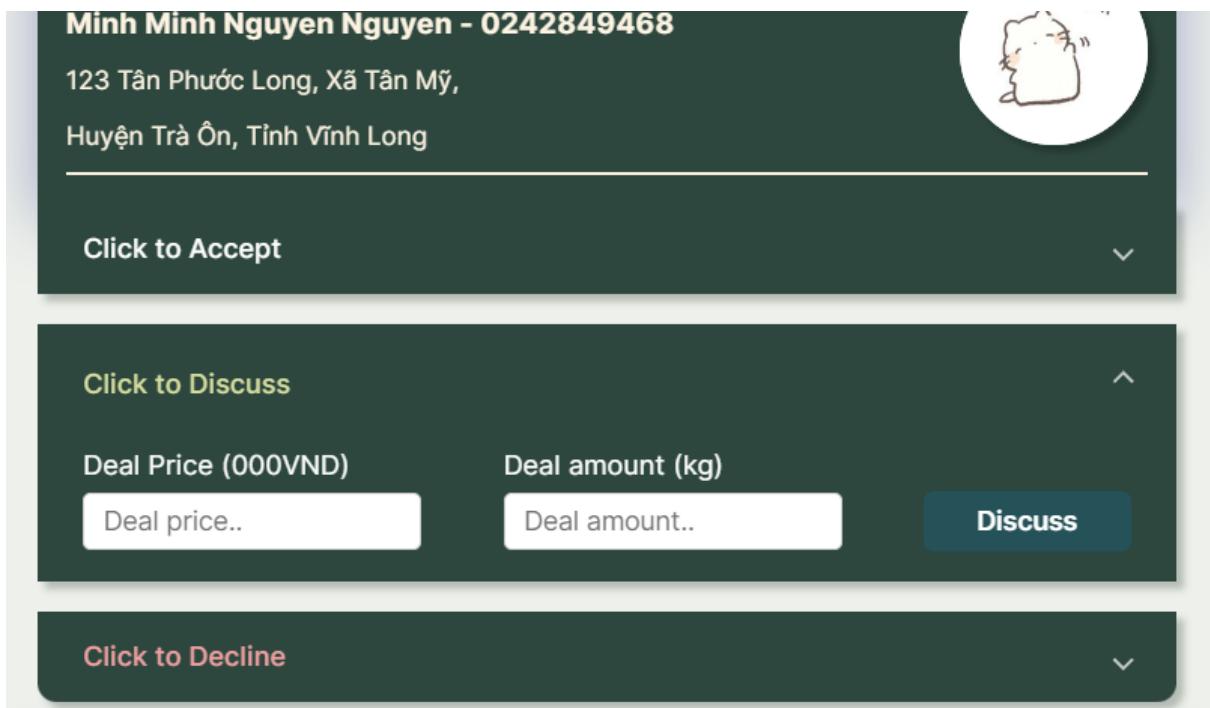


Figure 3. 19. Negotiating about the new price

Decline this order.

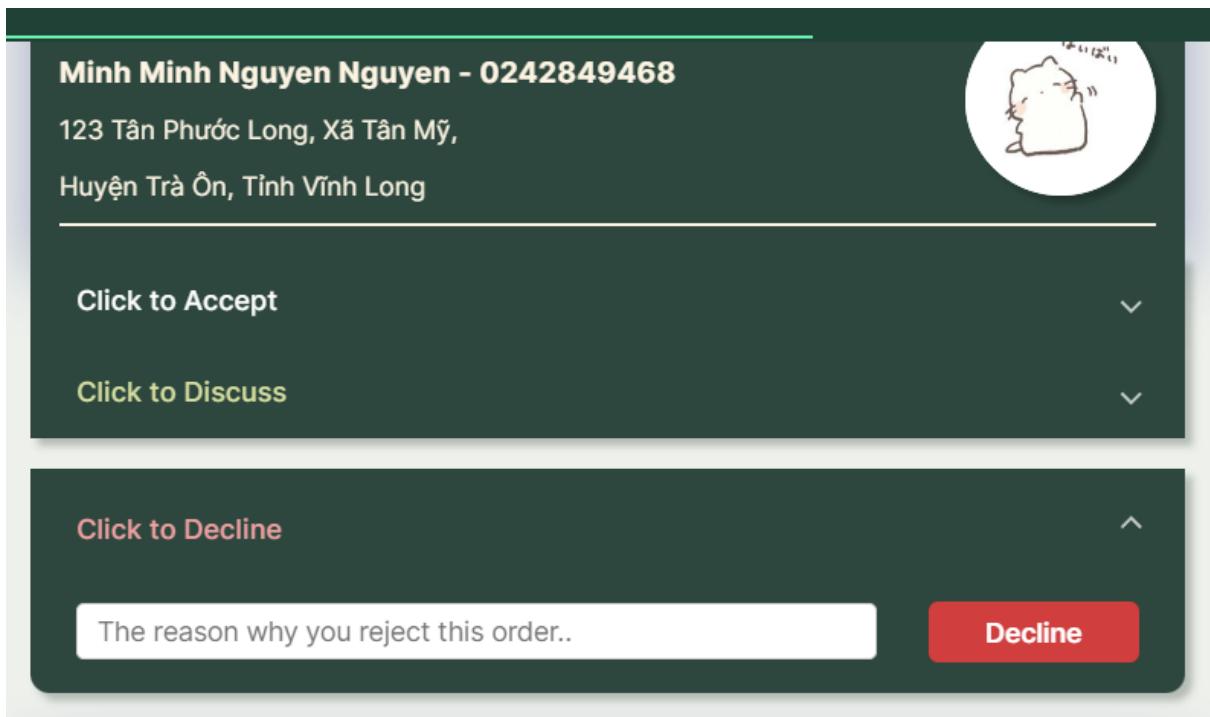


Figure 3. 20. Decline order

Let's try to negotiate a new price..

## CHAPTER 5: IMPLEMENTATION AND TESTING

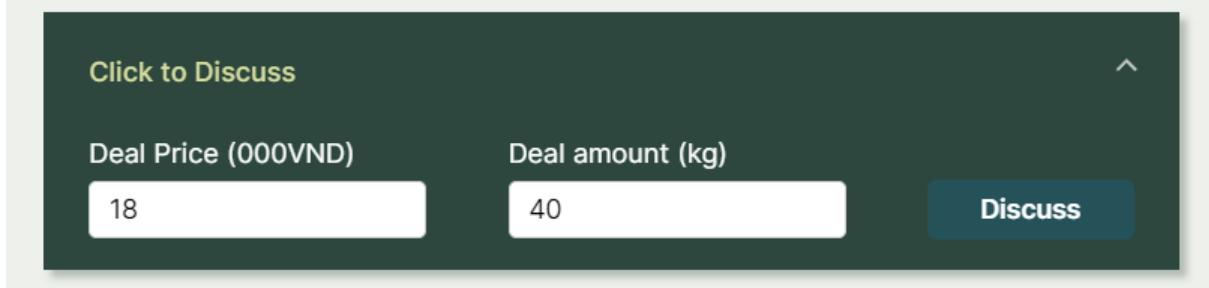


Figure 3. 21. Discuss new price and amount

The order will be moved from "Pending" to "Dealing" status with the new price and quantity.

Order	Deal amount (kg)	Deal price (vnd)	Order #
Chuối xiêm	40	18000	50
Nho	50	85000	37
Nho	50	59000	32

Figure 3. 22. Dealing tab

The phone will receive a notification.

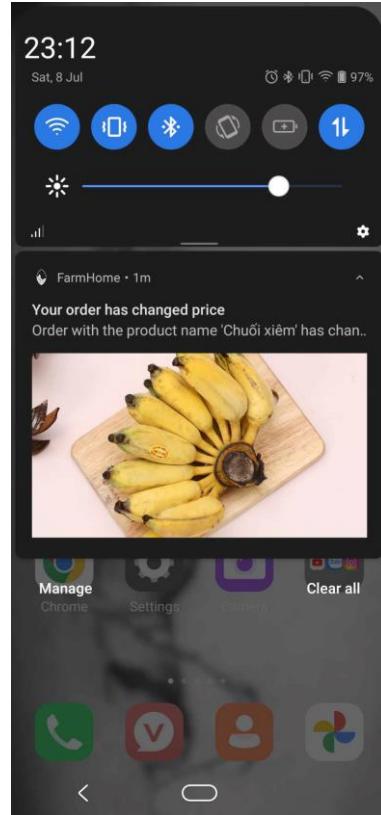
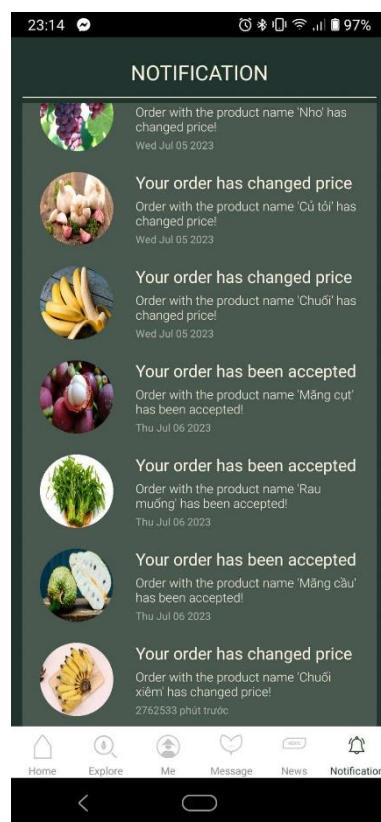


Figure 3. 23. Phone received notification



Going into the order details, you will see a section to confirm the deal price. Agreeing to it.



Figure 3. 24. Order details

After the merchant agrees to the new price, the order will be moved back to the "Pending" status.

## CHAPTER 5: IMPLEMENTATION AND TESTING

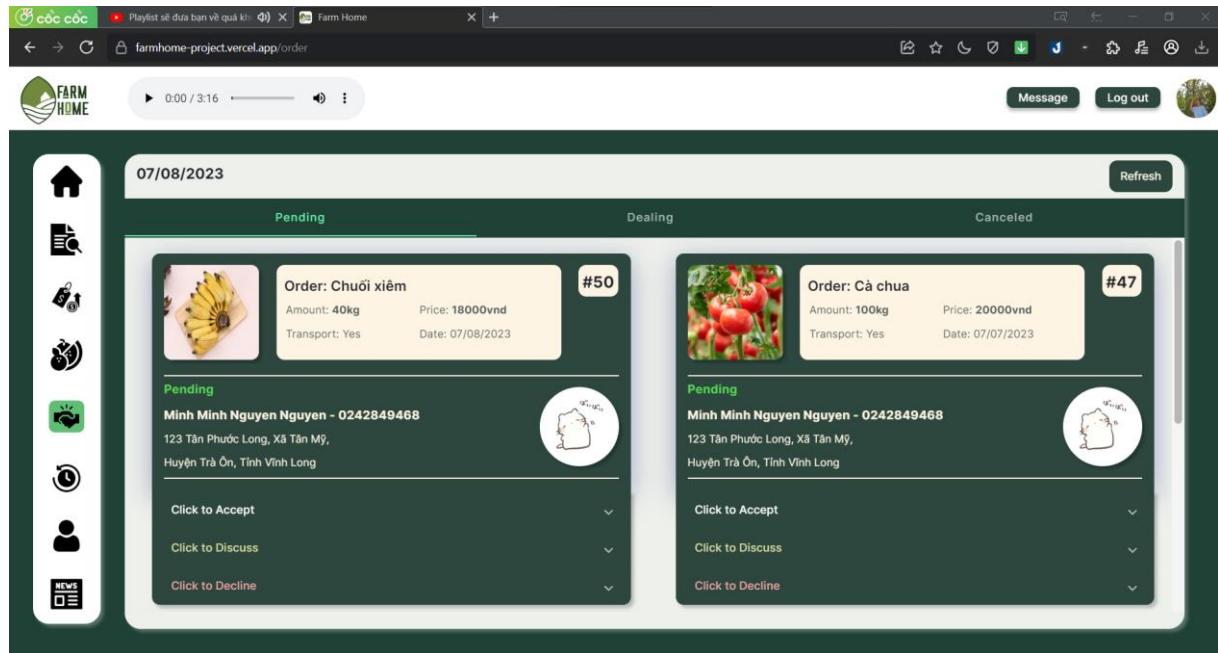


Figure 3. 25. Pending tab

Agree to the order and select the delivery date (if there is a delivery option available, otherwise this section may not be present).

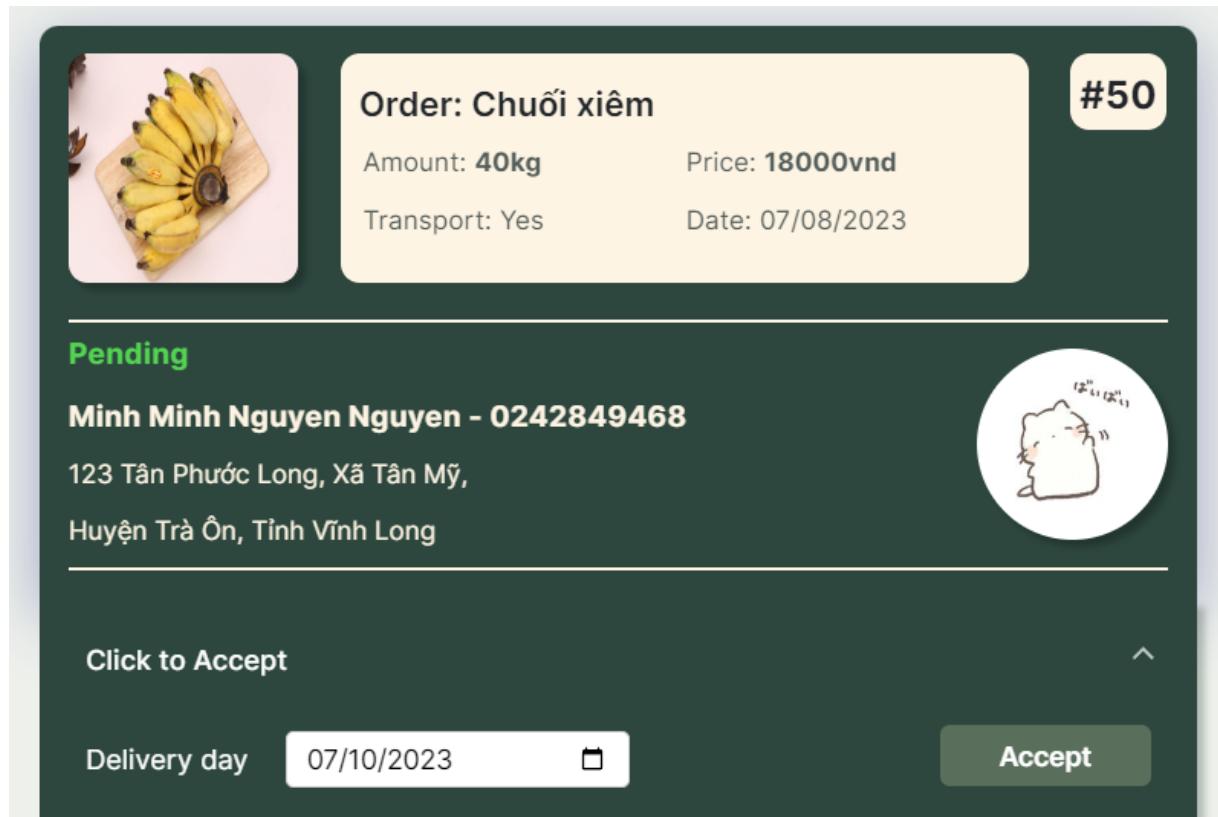


Figure 3. 26. Accept order

## CHAPTER 5: IMPLEMENTATION AND TESTING

After the order is successfully confirmed.

The screenshot shows the Farm Home history page for July 8, 2023. It displays three completed orders:

- #31 Chuối xiêm**: Amount 40, Price 18000, Transport Yes, Date 07/08/2023, Delivery date 07/10/2023. Total: 720,000đ. Shipped by Minh Minh Nguyen Nguyen, 0242849468, 123 Tân Phước Long, Xã Tân Mỹ, Huyện Trà Ôn, Tỉnh Vĩnh Long.
- #29 Măng cầu**: Amount 52, Price 50000, Transport No, Date 07/05/2023, Delivery date 07/06/2023. Total: 2,600,000đ. Shipped by Minh Minh Nguyen Nguyen, 0242849468, 123 Tân Phước Long, Xã Tân Mỹ, Huyện Trà Ôn, Tỉnh Vĩnh Long.
- #28 Rau muống**: Amount 10, Price 6000, Transport No, Date 07/05/2023, Delivery date 07/09/2023. Total: 60,000đ. Shipped by Minh Minh Nguyen Nguyen, 0242849468, 123 Tân Phước Long, Xã Tân Mỹ, Huyện Trà Ôn, Tỉnh Vĩnh Long.

A sidebar on the left contains icons for Home, Search, Categories, Products, Orders, and News.

Figure 3. 27. History page



Figure 3. 28. History screen

In addition, there are other features to explore, such as profile management, search with image in mobile, etc ...

## CHAPTER 6. CONCLUSION

### 6.1. Achieved result

#### 6.1.1. About knowledge and skills:

We see the use of Java with the Spring Boot framework and a small portion of Python for backend development. This enables the creation of APIs to support frontend and mobile applications efficiently. Students are equipped with fundamental knowledge of Java with the Spring Boot framework and Python, including object-oriented programming, data structures, and algorithms. They also learn about web development, including handling HTTP requests and responses, creating RESTful APIs, and implementing web security.

Additionally, Spring Boot facilitates database connection, allowing students to learn about object-relational mapping and writing efficient queries. It is crucial for students to understand how to deploy Spring Boot applications to different environments and platforms, and they can explore other technologies through cloud platforms like Google Cloud Console or Railway to enhance continuous integration and delivery pipelines. Moreover, working with TensorFlow makes the exploration of machine learning and AI applications more practical.

Bugging plays a vital role in the development process as students encounter issues and errors, and they enhance their troubleshooting and problem analysis skills using debugging tools.

The system highlights the use of JavaScript, Typescript, SASS/SCSS with the React JS, React Native library as well as Angular with its material for developing the admin page, website for farmers, mobile app for merchants enabling dynamic styling, animations, and interactivity for the website.

Students gain fundamental knowledge of JavaScript, Typescript, SASS/SCSS and the React library, Angular material, including syntax, data types, variables, control flow, and manipulation of HTML elements through the Document Object Model

(DOM). They also learn about handling asynchronous operations and user interactions, such as making API requests, managing callbacks, and responding to events.

The JavaScript ecosystem offers a variety of third-party libraries and frameworks, like Axios for HTTP requests and Redux for managing application state, providing additional features and functionalities for specific purposes. Debugging is also emphasized as an essential skill for troubleshooting and analyzing problems encountered during the development process.

### **6.1.2. About the topic:**

The backend program is built to provide comprehensive support and ensure the required APIs of the frontend and mobile applications are met.

The admin website provides features for user monitoring, user management, product management, category management, and sales data analytics.

A mobile app used for “Merchant” role to view and order products, also support for other support feature like chat, change information and read news, etc... When there is an event that they will have a notification sent back to their phone.

The website is designed to assist farmers in managing agricultural products and handling orders. Users can log in to their accounts and manage information about various agricultural products, including detailed descriptions, images, descriptions, and prices. They can add, modify, or delete products as needed. It also provides features to manage orders placed by customers. Farmers can view a list of pending orders, confirm, discuss or decline orders, etc...

### **6.2. Strengths**

Using a third-party file storage provider ensures seamless and efficient uploading of images and files without interruptions or excessive load.

Database and applications are hosted on the internet, users do not need to install third-party applications to be able to use them.

Integrated a agricultural detection model that can be used to classify those products for further automation process in the project such as classification, searching by image.

The errors have been thoroughly tested by all members who have been learning through the software testing course. Understand the procedures for finding errors and making corrections.

The interface undergoes continuous improvements based on feedback from instructors, resulting in enhanced user suitability. Careful sourcing and consultation with experts ensure the accuracy and relevance of statistical indicators. By optimizing simple functions, the platform achieves higher efficiency and effectiveness.

The platform caters to both "Merchant" and "Farmer" sides of agricultural product trading, offering essential functions for trading purposes. Additional support functions are available to facilitate the trading process.

The platform collects data to provide users with real-time market prices, enabling them to stay updated with current pricing trends. Notifications are sent directly to users's mobile phones, ensuring prompt awareness of any events or updates related to their orders.

With the assistance of fruit detection technology, users can easily identify different types of fruits by capturing images. This streamlined process simplifies data recording and information retrieval, thanks to the utilization of a pre-trained model.

### 6.3. Drawbacks

Although some functions within the application have not been fully optimized, they may not provide optimal access speed. As the application has not been specifically optimized for high frequency and large-scale usage, there is a possibility of encountering errors related to speed or storage capacity.

While some errors can be returned to the frontend for self-adjustment, they are not completely resolved autonomously. It's important to note that the machine learning model used for classification purposes may misclassify some new agricultural products.

In real-world scenarios, there may be instances where certain features are missing or insufficient, which can present challenges for users. For example, the absence of comprehensive statistical data can create difficulties for individuals in financial fields who rely on detailed statistics for documentation and analysis purposes.

Another potential challenge arises from the limited availability of advanced permission management features, which can make it challenging to effectively manage user roles and access levels, potentially compromising system security and control.

Furthermore, the absence of a dedicated reference price management feature for farmers can pose difficulties. Farmers often rely on reference prices to make informed decisions about pricing their products, and the lack of this functionality can complicate the process of setting competitive prices in the market.

The current website for farmers lacks important functions such as messaging, notifications, and language switching. Additionally, mobile does not support IOS devices at the moment. Furthermore, there is a lack of detailed delivery statuses for both farmers and merchants, which can impact tracking and managing orders effectively. The performance of the chat service is not up to the desired standard. Moreover, the website does not offer support for social media login, as well as features like rating and comments for farmers, limiting engagement and feedback from users.

### **6.4. Future developments**

Improve the limitations that have not been reached.

To accommodate future development, potential APIs can be integrated to introduce functionalities like social media login, rating, commenting, and more. Clearer user categorization can be achieved by implementing additional roles such as a super admin or external service providers for agricultural products. Optimizing coding logic and algorithms will enhance access speed. Furthermore, the development of a new self-learning model can enable the system to learn and detect a wider range of agricultural products with improved accuracy.

Introducing a super admin role to facilitate systematic management of other admins would be a valuable improvement. Additionally, incorporating a feature that enables admins to make decisions and adjust reference prices for farmers' products would enhance functionality. Furthermore, implementing programs to identify and retain potential users, along with incentives for popular products, would contribute to the platform's growth and engagement.

Creating a better business model for the circular economy entails providing solutions for farmers to address issues related to bad harvests and damaged products. By implementing strategies such as crop diversification, climate-resilient farming practices, and effective post-harvest management techniques, farmers can mitigate the impact of unfavorable harvests. Additionally, introducing innovative packaging and storage methods can help reduce product damage and waste, promoting sustainable agricultural practices and resource optimization.

To enhance the delivery process, it is essential to offer more options to users. This can be achieved by providing merchants with access to warehouses where they can store their products conveniently and securely. Moreover, integrating with third-party delivery services can offer greater flexibility and choice for both farmers and merchants, ensuring efficient and reliable delivery of agricultural products to customers. These initiatives contribute to the overall improvement of the supply chain and facilitate smoother transactions within the agricultural ecosystem.

---

## REFERENCES

[1]. Amigoscode, Spring Boot and Spring Security with JWT including Access and Refresh Tokens (July 25th 2021)

Retrieved from:

[https://www.youtube.com/watch?v=VVn9OG9nfH0&ab\\_channel=Amigoscode](https://www.youtube.com/watch?v=VVn9OG9nfH0&ab_channel=Amigoscode)

[2]. loda.me, Hướng dẫn Spring Security + JWT (Json Web Token) + Hibernate (May 31st, 2019)

Retrieved from: <https://viblo.asia/p/huong-dan-spring-security-jwt-json-web-token-hibernate-oVlYGmoK8W>

[3]. baeldung, Spring Data JPA Delete and Relationships (September 5th, 2022)

Retrieved from: <https://www.baeldung.com/spring-data-jpa-delete>

[4]. Michał Pierzchała - Head of Technology at Callstack, React Native Core Contributor Summit 2022

Retrieved from: <https://reactnative.dev/blog>

---

## WORKFLOW

ID	Student Name	Work
19110151	Pham Viet Anh	Spring boot, Tensorflow, React JS
19110145	Tran Dang Khoa	React Native, React JS
19110100	Nguyen Le Minh Nhut	Angular