**BMIS2003 Blockchain Application Development**

**Task Description**

Name : Lim Weng Ni ID : 2309341

Programme : RSD Group : 5

Instruction: Answer **ALL** the questions.

1. Please briefly describe the module(s)/function(s) you engaged in the assignment.

**Durian Purchase Module**

* The system manages durians based on their status:

- Locked: The purchase button is disabled, preventing consumers from buying.

- Cancelled: The durian is hidden from the list.

- Sold: The button is disabled, displaying "Sold" instead of a purchase option.

* Once a durian is purchased, the retailer can see the new owner's details.

**Add Durian Module**

* Implemented validations for harvest and ripen dates:

- Only fresh durians in specific stages are allowed:

- Unripe: Days 1-3

- Ripen: Days 4-5

- Overripe: Days 6-7 (not allowed for sale).

* Automatically assigns a unique durian ID to reduce human error.

**Modify TimeLock Module**

* Durians with canceled TimeLocks are marked as "cancelled."

**Retailer Authorization**

* Admin authorization for retailers has been successfully implemented.
* The system checks access to the retailer page, ensuring that unauthorized users cannot access it. If a user is not authorized as a retailer, they are redirected away from the page.

1. What are the strengths of the modules/functions created by you?
2. User-Friendly Workflow: The modules streamline processes for retailers and consumers, making it easy to manage durian records and purchases without unnecessary complexity.
3. Robust Validation: The inclusion of thorough validations ensures that only fresh and properly harvested durians are added, minimizing the risk of selling subpar products.
4. Dynamic Status Management: The system effectively manages durian statuses (locked, available, cancelled, sold), providing clear guidance to consumers on what they can purchase.
5. Automatic ID Generation: By automatically generating durian IDs, you reduce the potential for human error during data entry, enhancing data integrity.
6. Enhanced Authorization: The fix to the authorization process ensures that once retailers are authorized, they can add records without incurring repeated fees, encouraging more active participation.
7. Clear Ownership Tracking: After a durian is purchased, the ability for retailers to view the new owner improves transparency and trust within the system.
8. Cancelation Feature: The ability to cancel TimeLocks adds flexibility for retailers, allowing them to manage their inventory dynamically.
9. Scalability: The modular design allows for easy future enhancements, enabling the addition of new features or adjustments as needed.
10. What are the weaknesses of the modules/functions created by you?
11. Dependency on Time Constraints: The reliance on time-based conditions (like ripening stages) can lead to complications if there are delays in harvesting or unexpected changes in durian quality.
12. Authorization Delays: Although the authorization fix improves the process, there might still be delays in granting access, which could hinder retailers from quickly adding durians.
13. Complexity in Status Management: The different statuses (locked, available, cancelled, sold) can complicate user interactions, particularly for consumers who may find it challenging to navigate these changes.
14. What have you learned in doing this assignment?
15. Modular Design: I appreciated the benefits of a modular approach, which allows for easier maintenance and scalability of the application as features and functionalities expand.
16. Problem-Solving Skills: The assignment challenged me to troubleshoot and resolve issues, such as fixing the double payment problem and ensuring robust error handling.
17. Authorization Mechanisms: I discovered the nuances of implementing effective authorization processes, ensuring that actions like adding records are seamless for authorized users.
18. User Experience Considerations: I recognized the significance of creating intuitive user interactions, particularly regarding status management and error feedback, to enhance usability.
19. Smart Contract Development: I've gained a deeper understanding of how to write and implement smart contracts, including handling various states and transitions for assets like durians.
20. What are the challenges, if any, faced by you while working on this assignment?
21. Complexity of Smart Contracts: Understanding the intricacies of Solidity and smart contract development was initially challenging. Ensuring that the contracts functioned as intended required careful consideration of various states and transitions.
22. Debugging Issues: I faced difficulties debugging issues like the double payment problem. Identifying the root cause took time and required a thorough review of the code and logic.
23. User Experience: Balancing functionality with user experience was challenging. I had to consider how to present information clearly and ensure intuitive interactions, especially around status changes.

Signature:  Date: 22/09/2024