Code review Isina Token contract

Inheritance: The contract is inherited from standard contracts from the Open-Zeppelin repository - ERC20, ERC20Detailed, Ownable. These contracts are standard and unchanged.

Variables:
1. UNLOCK_TIME - containing the unix time to start unlocking tokens.
2. STOP_ALL_LOCKS_TIME - containing unix-time of completion of all locks.
3. STOP_SELL_TIME - unix-time to complete the sale of unlocked tokens.
4. team_tokens - the number of team tokens.
5. marketing_tokens - the number of tokens for marketing.
6. reserve_tokens - the number of tokens for the reserve.
7. seed_tokens - the number of tokens for sale in the seed round.
8. private_tokens - the number of tokens to sell in the private round.
9. launchpad_tokens - the number of tokens to sell on the launchpad round.
10. tokens_holder is the address of the token holder's wallet for sale.
11. team - team wallet address.
12. marketing - the address of the wallet for marketing.
13. reserve - wallet address for the reserve.
14. bank is a copy of a bank contract.
15. seed_balance - match addresses and their investments during the seed round.
16. private_balance - matching addresses and their investments during the private round.
17. seed_bonus - match addresses and their investment bonuses during the seed round.
18. private_bonus - match addresses and their investment bonuses during the seed round.
19. seed_locked - matching of addresses and blocking states of their tokens purchased during the seed round.
20. private_locked - matching addresses and blocking states of their tokens purchased during the private round.
21. tokens_for_sell - match addresses and quantities of tokens for sale.

Functions:
1. Modifier only_tokens_holder - checks that the sender's address matches the variable tokens_holder.
2. Modifier after_unlock_time - checks that the block creation time was more than the UNLOCK_TIME variable.
3. The before_stop_sell_time modifier - checks that the block creation time was less than the STOP_SELL_TIME variable.
4. constructor (address _tokens_holder, address _team, address _marketing, address _reserve) Assigns the transferred addresses to tokens_holder, team, marketing and reserve. Assigns seed_tokens, private_tokens, launchpad_tokens, team_tokens, marketing_tokens, reserve_tokens to the corresponding number of tokens, and also releases tokens to the

corresponding addresses in the number:
102203100 for tokens_holder (seed_tokens)
163524960 for tokens_holder (private_tokens)
81762480 for tokens_holder (launchpad_tokens)
28616868 for team (team_tokens)
4088124 for marketing (marketing_tokens)
28616868 for reserve (reserve_tokens)

5. trasnfer_from_bank (address recipient, uint256 amount)
   Can only be called with the address owner (Ownable contract).
   Translates tokens from bank to recipient in amount.
   [The function is safe]
6. unlock_seed ()
   Can only be called after the start time for unlocking tokens.
   It can only be called before the token unlock completion time.
   It can only call an address whose seed_locked is set to true, i.e. Tokens purchased in the seed round are blocked.
   seed_locked is set to false.
   transfer_for_sell is increased by the result of subtracting from the total seed balance of the investor of its available seed balance.
   The investor's seed_balance is set to 0.
   transfer_for_sell is increased by the result of subtracting the available seed bonuses from the total seed bonus of the investor.
   seed_bonus is set to 0.
   [The function is safe]
7. unlock_private ()
   Can only be called after the start time for unlocking tokens.
   It can only be called before the token unlock completion time.
   It can call only the address for which private_locked is set to true, i.e. Tokens purchased in the private round are blocked.
   private_locked is set to false. transfer_for_sell is increased by the result of subtracting the available private balance from an investor’s total private balance sheet.
   investor's private_balance is set to 0.
   transfer_for_sell is increased by the result of subtracting from the total private investor bonuses of his available private bonuses.
   private_bonus is set to 0.
   [The function is safe]
8. burn_seed_tokens ()
   Can only be called with the address tokens_holder. Burns tokens from the tokens_holder in the amount of seed_tokens. seed_tokens sets to 0.
   [The function is safe]
9. burn_private_tokens ()
   Can only be called with the address tokens_holder.
   It burns tokens_holder tokens in the amount of private_tokens. private_tokens sets to 0.
   [The function is safe]
10. burn_launchpad_tokens ()
   Can only be called with the address tokens_holder.

Burns tokens from tokens_holder in the amount of launchpad_tokens.
launchpad_tokens sets to 0.
[The function is safe]

11. set_bank (address payable bank_address)
Only owner can be called.
Changes the value of the bank address to bank_address.
[The function is safe]

12. transfer (address recipient, uint256 amount)
Override the standard ERC20 function.
[The function is safe]

13. transferFrom (address sender, address recipient, uint256 amount)
Override the standard ERC20 function.
[The function is safe]

14. send_seed_tokens (address recipient, uint256 amount)
May only be called with the address tokens_holder.
It can only be called when seed_tokens> 0.
seed_balance [recipient] is incremented by amount.
The variable bonus_amount is created and assigned the value of 14.96% of the amount. seed_bonus [recipient] increases by bonus_amount.
seed_locked [recipient] is set to true.
seed_tokens [recipient] decreases by amount.
seed_tokens [recipient] decreases by bonus_amount.
Tokens are transferred from the address tokens_holder to the recipient in the amount amount + bonus_amount.
[The function is safe]

15. send_private_tokens (address recipient, uint256 amount)
Can only be called with the address tokens_holder.
It can only be called when private_tokens> 0.
private_balance [recipient] is incremented by amount.
The variable bonus_amount is created and assigned the value of 14.96% of the amount. private_bonus [recipient] increases by bonus_amount.
private_locked [recipient] is set to true.
private_tokens [recipient] decreases by amount.
private_tokens [recipient] is reduced by bonus_amount.
Tokens are transferred from the address tokens_holder to the recipient in the amount amount + bonus_amount.
[The function is safe]

16. send_launchpad_tokens (address recipient, uint256 amount)
May only be called with the address tokens_holder. It can only be called when launchpad_tokens> 0.
launchpad_tokens [recipient] is reduced by amount.
The transfer of tokens from the address tokens_holder to the recipient in the amount amount is performed.
[The function is safe]

17. get_available_balance (address sender)
The available_balance local variable is created and assigned the value of the sender address token balance.
If sender is a team, then the difference between team_tokens and

get_available_team_balance () is subtracted from available_balance.
If sender is marketing, then the difference between marketing_tokens and get_a
is subtracted from available_balance
[The function is safe]

18. get_available_marketing_balance ()
    Depending on the current time, the available number of tokens for marketing is returned.
    [The function is safe]
19. get_available_reserve_balance ()
    Depending on the current time, the available number of tokens for the reserve is returned.
    [The function is safe]
20. get_available_seed_balance (address sender)
    Depending on the current time, the available number of sender address tokens purchased in the seed round is returned.
    [The function is safe]
21. get_available_private_balance (address sender)
    Depending on the current time, the available for use number of sender address tokens purchased in a private round is returned.
    [The function is safe]
22. get_available_seed_bonus (address sender)
    Depending on the current time, the available number of bonus tokens of the sender address tokens is returned for the seeds purchased in the seed round.
    [The function is safe]
23. get_available_private_bonus (address sender)
    Depending on the current time, the available number of bonus tokens of the sender address is returned for those purchased in a private round.
    [The function is safe]
24. _transfer (address sender, address recipient, uint256 amount)
    Can only be called within a contract.
    If the current time is less than the completion time of all locks and if the token receiver is the bank address and if tokens_for_sell [sender]> = amount and if the current time is less than the completion of the sale of tokens to the bank contract and if the bank function was able to transfer sender airs, thenkens_for_sell [sender] decreases by amount.
    Otherwise, if the current time is less than the time of completion of all the locks, then amount <= get_available_balance (sender) is checked.
    Tokens are transferred from sender to recipient in amount amount.
    [The function is safe]