Testudo Bank Utilization Score Feature

Owner: Casey Oppenheim (casey.oppenheim@yahoo.com)

Problem Statement

Customers of Testudo Bank are currently rewarded the same, regardless of their loyalty to the bank. Therefore, this feature would enable the customers to receive monetary rewards (in the way of a larger interest percentage) for have a good score. The score will consider the number of transactions made, any history of overdraft, and involvement in crypto. This will be beneficial for the bank as it will incentivize users to put more money into the bank, and use the bank more in general.

Solution Requirements

- Customers should be able to be scored based on their history of deposits into the bank, their involvement in cryptocurrency, and their history of overdraft.
- Customers should be able to view their score on the front end. The score should be out of a maximum value, so that the user can compare themselves to others in the bank.
- Customers should be able to view which factors are contributing to their score in a breakdown format. For example, any history of overdraft will decrease their score and the number of deposits made should improve their score.
- Customers should have a place on the Testudo Bank application front-end to see their score history and evaluate why it's gone up/down compared to previous scores.
- Customers scores should be updated every 5 deposits of values larger than $20. Their

score should impact their interest rate application such that as the scores improve, the interest rate improves as well (and vice versa).

Solutions Considered

Interest Rate by Score Range

In this approach, customer's interest rates will only improve once they have moved into a new bracket of scores. Consequently, the interest rate will not change each time.

Interest Rate by Score Only

In this approach, the interest rate applied to the user's account will change with respect to the individual score, as opposed to the range that the score is in. Therefore, unless the score remains exactly the same between the 5 deposits needed to apply interest again, the interest rate will change each time, and the changes will be proportional to the change in score.

Pro/Con of all approaches considered:

Interest Rate by Score Range

Pros:

- Increases incentive to put more money in the bank because users have to make significant changes in order to move to the next bracket and receive the reward.
- Easier for customer to understand the interest rate they should expect to receive based on their score.
- Much easier to implement.
    - A formula for transforming the score into a comparable interest rate doesn't need to be developed.
    - A new interest rate doesn't need to be calculated and applied each time the score changes.
- More power in the banks hands to decide what values they would like to represent each level of the interest rate scale.

    - Easy to change the score ranges and interest rate values whenever the company wants.

Cons:

- Hard to determine appropriate ranges for scores and related interest rates because there are only arbitrary ways to calculate those values.
- It's important that breaking into a new score bracket isn't easily done in one 5-deposit session (or this solution is essentially the same as the alternative).

Interest Rate by Score Only

Pros:

- Interest rate will only change by a small amount each time (in proportion to the change in score) which could potentially cost less money for the bank
- Simplifies the process such that there are no levels/brackets to understand. The interest rate is simply a direct function of the score.
- Reduces the need for abitrary ranges and interest rates to be determined by the bank

Cons:

- The customer will not be able to determine which interest rate they should expect based on their score
    - The algorithm used to calculate the interest rate as a function of the score will not be publicly available on the front end
- Harder to implement.
    - Testudo Bank will have to develop an algorithm to calculate an interest rate as a function of the score.
        - It's important that the algorithm never produces an interest rate that is unrealistically high (or low)
    - The interest rate will have to be calculated each time the score changes which will be costly in terms of time.
    - The front end will also be more complicated because the interest rate history will be much longer

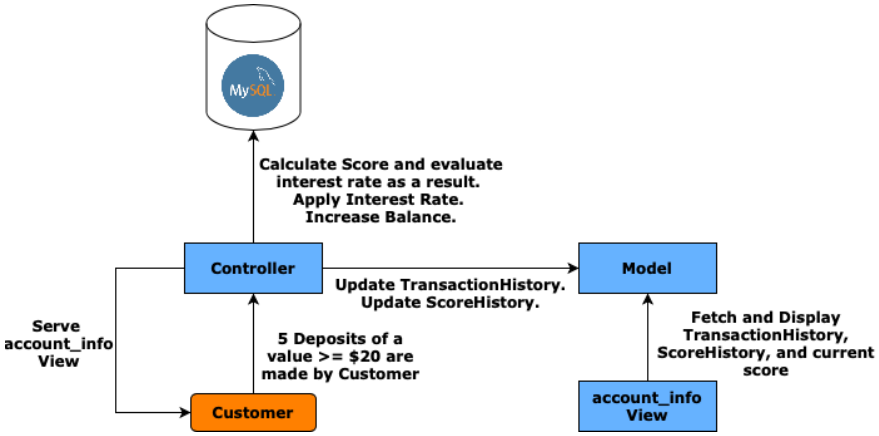
Proposed Solution

The proposed solution is the **Interest Rate by Score Range** because it enables Testudo Bank engineers to develop a Minimal Viable Product (MVP) that can be developed quickly due to the simplicity of the design. With this approach, only one algorithm has to be developed (the **utilization score**), as opposed to the alternative solution where two algorithms would have to be developed. Considering that the algorithm developed to determine the score in the **Interest Rate by Score Range** approach can be used in the **Interest Rate by Score Only** approach, it makes the most logistic sense to develop the former approach first and potentially add on to it later to satisfy the **Interest Rate by Score Only** approach.

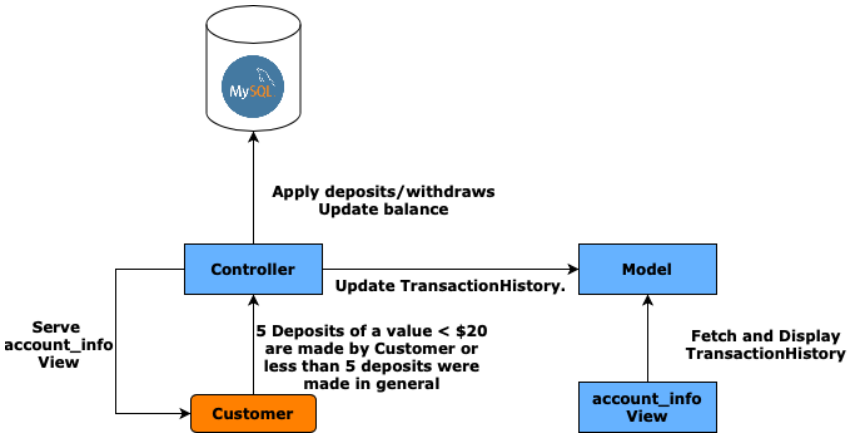
Technical Architecture

MVC Logic Diagrams

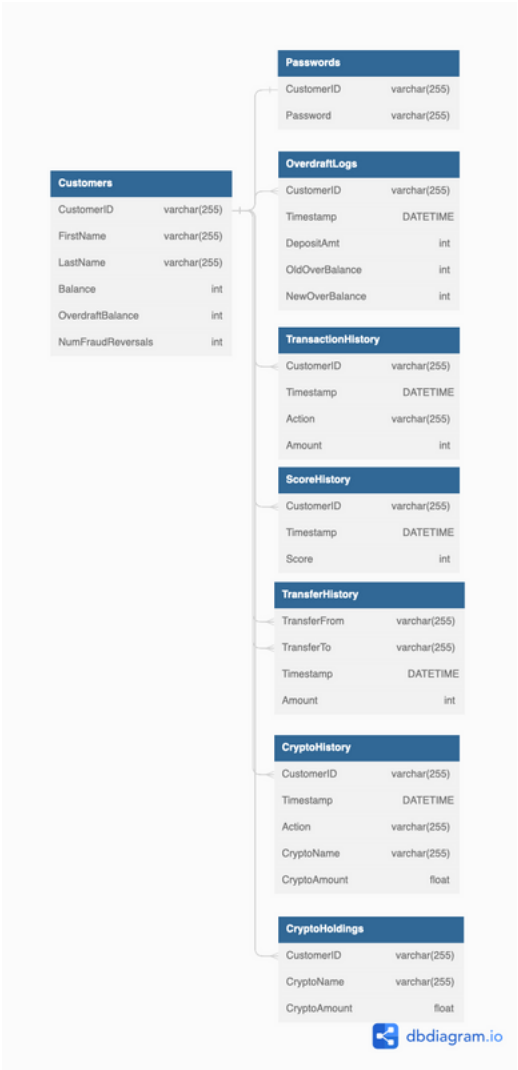
Score is Calculated



Score is Not Calculated



MySQL DB Schema



DB Schema Notes

- `CustomerID` from `Customers` table has a **one-to-one** relationship with `CustomerID` in the ScoreHistory table. This is because a single customer can only have one score at a time, and consequently can only have one list of the individuals previous scores.
- `Amount` in ScoreHistory is limited to integer values so that the score cannot be a decimal as this will interfere with the ranges established for each level in the interest rate scale.
- Transaction History `Action`s remain the same because, while the score effects the interest rate value, it doesn't handle any deposits or withdraws on it's own. The score is processed using the transaction history information, and then interest is applied as it usually is.