Adhithyakrishna Kovai Srinivasan

Person Number: 50317495

Email: akovaisr@buffalo.edu

Major Area: Online Betting

Title: Cricket betting lite

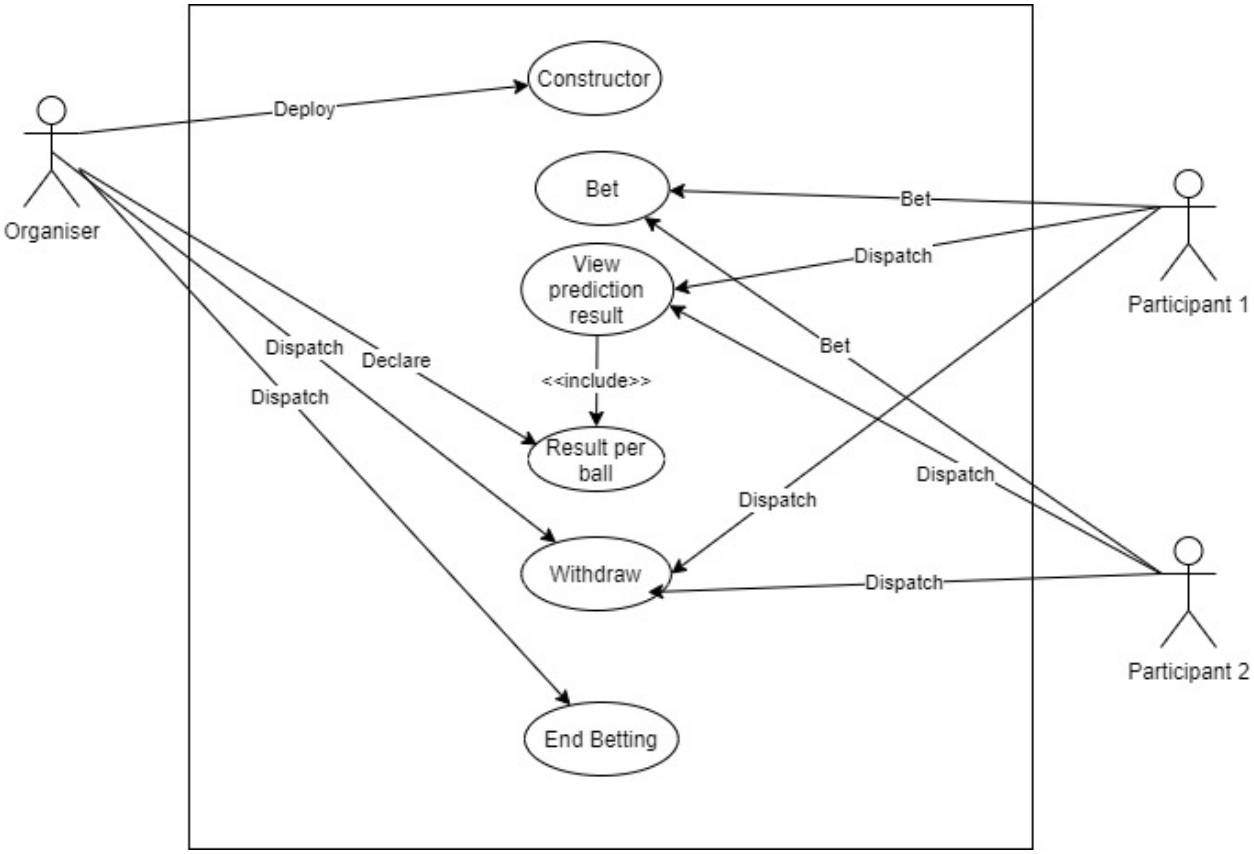
Dapp Name: BettingLite-Dapp

Clients: Participant

Abstract:

- The idea of this project is to create a decentralized betting platform that keeps the participants anonymous while betting. The idea of the stems from “per-over” betting concepts in the game of cricket that happens in local online groups or forums.
- The participant has to pay a nominal amount for participating in the online betting.
- An “over” consists of 6 balls. Before the start of the over, a participant can participate in betting. He is then allowed to bet against a real time event i.e. a player hitting a ball and scoring a particular run. The player can choose from 8 options. Runs from 0 to 6 or ‘W’. The probability of a player scoring 6 and 4 is high. So, we assign a higher value like 2. The probability of the batsman getting out is even more high, so we assign twice the value of 6 and 4 i.e. 4. For a score of 0 to 6 we assign a score of 1. For every wrong guess we reduce the corresponding values from the total score.
- The participant is declared a winner if the cumulative score is in positives and he is declared a looser if the cumulative sum is in negatives and transaction occurs during the end of over.
- The percentage of winning is calculated from the nominal amount paid by the user. The higher he bets, the higher he wins.
- The DAPP will establish a direct connection between the organizer and participant. The transaction would be real time and take place by the end of on over and only between the organizer and participant.
- The shorter format of betting will attract more users for the application and since it is based off of the block chain the participants can trust and invest amount for online betting.
- The block chain will ensure a trust layer that will verify, that the participant would definitely win or lose money even if he quits mid-way and the deserving party gets the right amount.

2. Use case diagram



In the use case diagram, the roles of the actors include:

1) Organizer
   a) Authorizing participants for betting.
   b) Declaring result as per the score.
   c) Withdraw amount if he wins a series.
   d) End betting.

2) Participant
   a) Participate in betting
   b) View prediction result
   c) Withdraw the winning amount

3. Contract Diagram

BettingLite-Dapp

//data
struct participants {uint256 bettingVal; uint256 points;}
mapping (address => participants) result;
enum Score {ZERO, ONE, TWO, THREE, FOUR, FIVE, SIX, SEVEN, EIGHT}
enum Phase {INIT, BETTING, REVEAL, DONE}

//modifier
modifier _ownerOnly();
modifier _playerOnly();

//function
function incrementPhase() public _playerOnly
function createPassword (string memory password) _playerOnly
function placeBet (string memory password) _playerOnly;
function buyTokens (uint256 amount);
function getArbitarBalance() view public returns (uint256);
function getPlayerBalance() view public returns (uint256);
function predictWinning(uint256 prediction, uint256 actualValue) _playerOnly;
function declareWinner() public view returns (bool);
function withdraw (string memory password) _playerOnly;

4. State Diagram