**Online Poker Timer MVP**

**FUNCTIONAL SPEC**

* 1. Account registration - Sign-Up
  2. Login adds cookie so user is remembered. No logout for MVP.
  3. Single fixed blind schedule retrieved from database (per below)
  4. Each blind level 7 mins. Ignore the breaks and chip-ups for MVP.
  5. Game automatically ends (in browser and on server) at the end of the last blind. Game automatically ends if paused for more than 4 hours.
  6. Only one active game allowed (only the current game is saved)
  7. Timer must survive browser re-connect (meaning the timer must be sync'd with the server which is the master)
  8. Everything on one page after login
  9. One CSS design template (no themes)
  10. Two controls on the page: New Game, Play/Pause (ideally New throws a warning to confirm intention when a game is running)
  11. System Management console page that reports how many games are currently active.

**PROGRAM SPEC**

* 1. Handle thousands of simultaneous timers running (single server)
  2. Handle the three browser control events (New, Start, Pause)
  3. Only the browser counts down. The server always knows the current timer value based on the blind schedule and the time since the game started minus any pause events. Upon a browser re-connect, the browser retrieves the current timer position automatically and continues without user intervention.
  4. Track cumulative pause time for the game.
  5. Clicking Pause, pauses the JavaScript timer in the browser and sends the pause event to the server.
  6. Clicking Play, retrieves the current blind info and current time from the server and resets the JavaScript clock to that value and starts the clock. (This enforces the server as the master clock for MVP.)
  7. Server Values:
     1. userId
     2. userName
     3. password
     4. gameStartTime
     5. gameCurrentState - running/paused
     6. currentPausedSinceTime
     7. accumulatedPausedTime

|  |  |  |  |
| --- | --- | --- | --- |
| **Level** | **Small Blind** | **Big Blind** | **Ante** |
| 1 | 5 | 10 | 0 |
| 2 | 10 | 20 | 0 |
| 3 | 25 | 50 | 5 |
| 4 | 50 | 100 | 10 |
| 5 | 75 | 150 | 10 |
|  | Break - $5 Chip-up |  |  |
| 6 | 100 | 200 | 25 |
| 7 | 150 | 300 | 25 |
| 8 | 200 | 400 | 25 |
| 9 | 300 | 600 | 50 |
| 10 | 400 | 800 | 50 |
|  | Break - $25 Chip-up |  |  |
| 11 | 500 | 1000 | 100 |
| 12 | 700 | 1400 | 100 |
| 13 | 1000 | 2000 | 200 |
| 14 | 1500 | 3000 | 300 |
| 15 | 2000 | 4000 | 400 |
| 16 | 3000 | 6000 | 600 |

{"blindScheduleName" : "Office Turbo", "levels" : [

{"level" : 1, "smallBlind" : 5, "bigBlind" : 10, "ante" : 0, "duration" : 7},

{"level" : 2, "smallBlind" : 10, "bigBlind" : 20, "ante" : 0, "duration" : 7},

{"level" : 3, "smallBlind" : 25, "bigBlind" : 50, "ante" : 5, "duration" : 7},

{"level" : 4, "smallBlind" : 50, "bigBlind" : 100, "ante" : 10, "duration" : 7},

{"level" : 5, "smallBlind" : 75, "bigBlind" : 150, "ante" : 10, "duration" : 7},

{"level" : 6, "smallBlind" : 100, "bigBlind" : 200, "ante" : 25, "duration" : 7},

{"level" : 7, "smallBlind" : 150, "bigBlind" : 300, "ante" : 25, "duration" : 7},

{"level" : 8, "smallBlind" : 200, "bigBlind" : 400, "ante" : 25, "duration" : 7},

{"level" : 9, "smallBlind" : 300, "bigBlind" : 600, "ante" : 50, "duration" : 7},

{"level" : 10, "smallBlind" : 400, "bigBlind" : 800, "ante" : 50, "duration" : 7},

{"level" : 11, "smallBlind" : 500, "bigBlind" : 1000, "ante" : 100, "duration" : 7},

{"level" : 12, "smallBlind" : 700, "bigBlind" : 1400, "ante" : 100, "duration" : 7},

{"level" : 13, "smallBlind" : 1000, "bigBlind" : 2000, "ante" : 200, "duration" : 7},

{"level" : 14, "smallBlind" : 1500, "bigBlind" : 3000, "ante" : 300, "duration" : 7},

{"level" : 15, "smallBlind" : 2000, "bigBlind" : 4000, "ante" : 400, "duration" : 7},

{"level" : 16, "smallBlind" : 3000, "bigBlind" : 6000, "ante" : 600, "duration" : 7}

]}

Note: considering deriving level from index position in array.

