BLOCKCHAIN PROOF-OF-CONCEPT PROPOSAL

Blockchain Development individual assignment.

Booking Rooms

Prepared by: Besik Kristesiashvili

Dear Mr./Mrs. **Alexander Panayotov**,

We would like to present a smart contracts-based blockchain solution to the discussed **Booking Rooms**. Please get acquainted with the proposed prototype to understand the structure and process of our solution below.

SUMMARY

A "Booking Room" smart contract is a blockchain-based solution designed to streamline and automate the process of reserving and managing rooms. It leverages the benefits of blockchain technology to enhance transparency, security, and efficiency in room booking operations.

The Solution

Centralized systems problem:

Traditional room booking processes can be time-consuming, prone to errors, and lack transparency. Centralized systems may cause an argument, overbooking, and limited accountability.

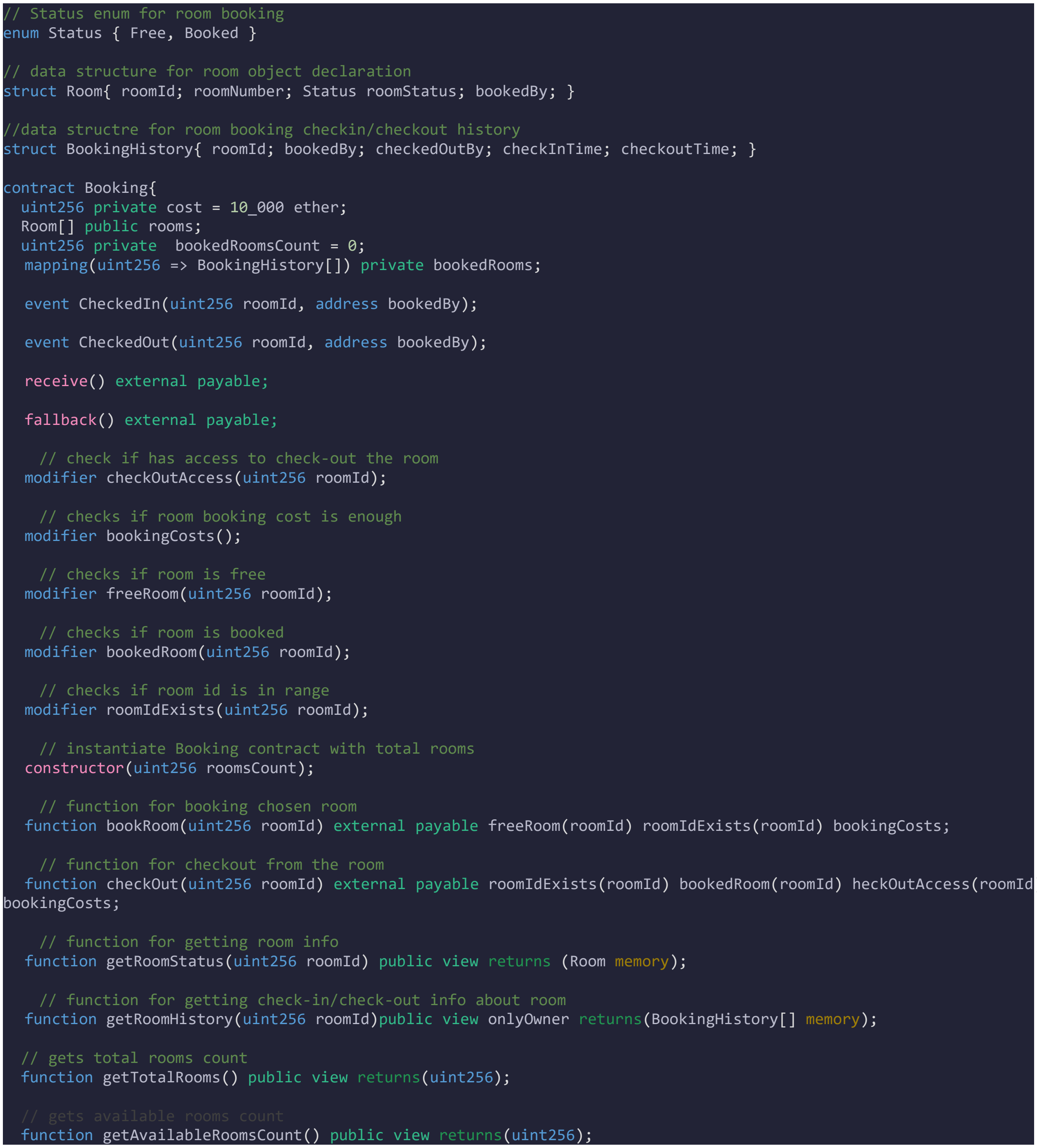
Blockchain can offer solution:

The "Booking Room" smart contract offers a decentralized and Interruption resistance platform for room reservations. It enables users to book rooms, check availability, and manage bookings through self-executing code.

- **Decentralization**: The smart contract operates on a blockchain network, eliminating the need for intermediaries and providing a trustless environment.
- **Room Listings**: Users can view available rooms.
- **Booking Process**: Users initiate bookings by sending a transaction to the smart contract. The contract validates availability and terms.
- **Confirmation**: Once a booking is made, the smart contract automatically confirms and records the reservation, preventing double bookings.
- **Payment Handling**: Users make payments in cryptocurrency or tokens, ensuring secure and borderless transactions.
- **Refunds**: The smart contract defines cancellation policies and processes refunds according to predefined rules.
- **Time-Stamped bookings**: All transactions and reservations are time-stamped and permanently recorded on the blockchain, enhancing transparency and accountability.
- **Notifications**: Users receive automated notifications about booking confirmations, payment status.
- **Unchangeable History**: The contract's unchangeable nature ensures an auditable history of all room bookings, reducing any misunderstandings.

THE BLOCKCHAIN SOLUTION

Smart contracts specification



```
// Status enum for room booking
enum Status { Free, Booked }

// data structure for room object declaration
struct Room{ roomId; roomNumber; Status roomStatus; bookedBy; }

//data structre for room booking checkin/checkout history
struct BookingHistory{ roomId; bookedBy; checkedOutBy; checkInTime; checkoutTime; }

contract Booking{
  uint256 private cost = 10_000 ether;
  Room[] public rooms;
  uint256 private  bookedRoomsCount = 0;
  mapping(uint256 => BookingHistory[]) private bookedRooms;

  event CheckedIn(uint256 roomId, address bookedBy);

  event CheckedOut(uint256 roomId, address bookedBy);

  receive() external payable;

  fallback() external payable;

    // check if has access to check-out the room
  modifier checkOutAccess(uint256 roomId);

    // checks if room booking cost is enough
  modifier bookingCosts();

    // checks if room is free
  modifier freeRoom(uint256 roomId);

    // checks if room is booked
  modifier bookedRoom(uint256 roomId);

    // checks if room id is in range
  modifier roomIdExists(uint256 roomId);

    // instantiate Booking contract with total rooms
  constructor(uint256 roomsCount);

    // function for booking chosen room
  function bookRoom(uint256 roomId) external payable freeRoom(roomId) roomIdExists(roomId) bookingCosts;

    // function for checkout from the room
  function checkOut(uint256 roomId) external payable roomIdExists(roomId) bookedRoom(roomId) heckOutAccess(roomId)
bookingCosts;

    // function for getting room info
  function getRoomStatus(uint256 roomId) public view returns (Room memory);

    // function for getting check-in/check-out info about room
  function getRoomHistory(uint256 roomId)public view onlyOwner returns(BookingHistory[] memory);

  // gets total rooms count
  function getTotalRooms() public view returns(uint256);

  // gets available rooms count
  function getAvailableRoomsCount() public view returns(uint256);
```

Smart contracts explanation

State Variables:

- **cost**: Private variable representing the cost of booking a room (10,000 Ether).
- **rooms**: Public array of Room structures to store room information.
- **bookedRoomsCount**: Private variable tracking the total number of booked rooms.
- **bookedRooms**: Private mapping that associates room IDs with an array of BookingHistory structures.

Events:

- **CheckedIn**: Event emitted when a user checks into a room.
- **CheckedOut**: Event emitted when a user checks out of a room.

Modifiers:

- **checkOutAccess**: Modifier to check if a user has access to check out from a room.
- **bookingCosts**: Modifier to check if the sent Ether matches the booking cost.
- **freeRoom**: Modifier to check if a room is available for booking.
- **bookedRoom**: Modifier to check if a room is already booked.
- **roomIdExists**: Modifier to check if a room ID is within the valid range.

Constructor:

- Initializes the contract with the total number of rooms.

Functions:

- **bookRoom**: Allows users to book a room by sending the booking cost in Ether. It checks room availability and the provided Ether value.
- **checkOut**: Allows users to check out from a room by sending the booking cost in Ether. It checks room availability, user access, and the provided Ether value.
- **getRoomStatus**: Retrieves information about a specific room's status (e.g., availability, booking status).
- **getRoomHistory**: Retrieves the check-in and check-out history of a specific room. Only accessible by the contract owner.
- **getTotalRooms**: Returns the total count of rooms.
- **getAvailableRoomsCount**: Returns the count of available rooms for booking.

Smart contracts process flow

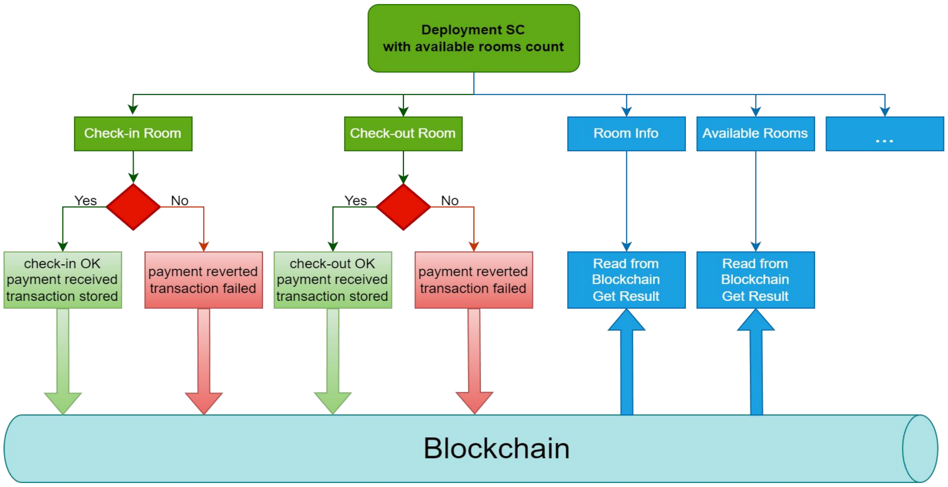


Image 1. Booking Rooms FlowChart.

EXPECTED BUSINESS RESULTS

Implementing a "Booking Room" smart contract can lead to several expected business results and benefits. Here are some potential outcomes that the smart contract could achieve for a room booking system:'s requirements to provide the following results:

Technical Benefits

- Increased transparency in the booking process as all transactions are recorded on the blockchain.
- Enhanced trust between users and the platform due to the immutable and auditable nature of blockchain records.
- Automated and streamlined booking process, reducing manual intervention and administrative tasks.
- Faster booking confirmations and room availability checks through self-executing smart contracts.
- Elimination of intermediaries such as booking agencies or third-party platforms, reducing associated fees.
- Cost savings by removing the need for manual verification and reconciliation of bookings.
- Secure and borderless payments using cryptocurrency or tokens, reducing the risk of fraud or chargebacks.
- Instant payment verification and confirmation through blockchain transactions.
- Accurate and tamper-proof recording of room bookings, preventing disputes related to overbooking or reservation changes.
- Empowerment of users to directly interact with the booking system without relying on intermediaries.
- Access to real-time room availability information and booking status.
- Simplified and user-friendly booking process, contributing to a positive user experience.
- Automated notifications and reminders for booking confirmations, check-ins, and check-outs.
- Improved data privacy as sensitive information can be stored off-chain while utilizing blockchain for transactional records.
- Enhanced security against data breaches or unauthorized access.
- Scalable solution that can handle a large number of bookings without sacrificing performance.
- Global access for users from different regions, enabling international bookings.
- Facilitated auditability and compliance with regulations by providing a transparent and traceable record of transactions.
- Simplified reporting and record-keeping for regulatory purposes.
- Flexibility to define and customize booking rules, cancellation policies, and pricing structures through smart contract logic.
- Differentiation from competitors by offering a modern and innovative room booking solution.
- Attraction of tech-savvy customers who value blockchain-powered services.

CONCLUSION

In conclusion, the "Booking Room" smart contract offers a transformative solution for the room booking industry, leveraging the power of blockchain technology to revolutionize the way accommodations are reserved and managed. By addressing the inefficiencies, lack of transparency, and trust issues inherent in traditional booking systems, this smart contract introduces a new era of efficiency, security, and user empowerment.

Through a seamless integration of smart contract automation, the "Booking Room" solution streamlines the entire booking lifecycle. Users benefit from a simplified and user-friendly process, instant confirmation, and real-time access to room availability. Automation ensures that bookings are accurately recorded, eliminating the risk of overbooking and reducing the need for manual intervention.

Thank you for your consideration,

Besik Kristesiashvili