Overview

PEARL (Personalized Experience Adaptive Relationship Liaison)

Purpose: To revolutionize the way individuals network and interact at various events by leveraging artificial intelligence. PEARL aims to reduce the discomfort and inefficiency of traditional networking by facilitating more meaningful and personalized connections among attendees.

Vision: Transform every event into a dynamic, engaging, and inclusive environment where every guest feels valued and finds genuine connections. PEARL will use AI to understand the needs and preferences of each attendee, making introductions and recommendations that enhance their event experience.

Objectives:

- To provide a seamless, intuitive interaction platform for event attendees via voice and text.
- To utilize AI to analyze and understand the personal interests, professional goals, and social styles of each participant.
- To create personalized networking opportunities that are perceived as natural and valuable.
- To ensure privacy and security in all interactions and data handling.
- To offer scalable solutions that can be adapted for different types of events and sizes.

Scope:

- The system will be designed to cater to a wide range of events, from small private gatherings to large professional conferences.
- Features will include AI-driven introductions, event-specific customization, privacy-focused follow-ups, and interactive elements like gamified awards and photo sharing.

User Experience Flow

1. Initial Contact

- **New Users:**
    - Upon connecting via voice, PEARL initiates a conversation to welcome the user.
    - PEARL asks if the user would prefer to continue the conversation via text. If the user agrees and provides a phone number, PEARL sends a text message to continue the interaction. A cookie is set with a unique session ID linked to this interaction.
    - If the user decides to continue via voice or does not provide a phone number, PEARL proceeds with the conversation in voice mode. The session ID is still saved in a cookie for future interactions.
- **Returning Users:**
    - Identified by the session cookie, PEARL greets them with familiarity, referencing previous interactions stored in MongoDB.
    - The conversation continues based on the user's history and preferences, enhancing the personalized experience.

2. During the Event

- **Text Interaction:**
    - If the user has opted for text communication, PEARL interacts through messages, asking for photos or updates about their experience.
    - Users can text photos to PEARL, which she stores and may later share as part of a party highlight reel.
- **Voice Interaction:**
    - For users continuing via voice, PEARL maintains engaging and interactive communication, offering information, facilitating introductions, and encouraging participation in event activities.
- **Dynamic Interaction Switching:**
    - If a user initially interacting via voice later texts PEARL from the registered number, she recognizes the number and seamlessly continues the earlier conversation, ensuring a cohesive experience.

3. End of the Event

- **Photo Sharing and Final Messages:**
    - At the event's conclusion, PEARL shares highlights from the event, including guest messages and photos collected via text.
    - She sends out personalized follow-up messages based on the interactions and connections made during the event.

4. Post-Event Engagement

- **Follow-Up:**
    - PEARL sends follow-up texts or voice messages to encourage further interaction among guests who connected during the event, fostering long-term relationships.

5. Continuous Interaction

- **Ongoing Communication:**
    - PEARL remains available for post-event questions or feedback, maintaining a channel for guests to reach out or receive updates about future events for 24 hours following the event. She sends one final message out asking if there is anything else they need before she goes offline.

Specific Dialogues and Interactions Based on User Flow

Here's how specific interactions with PEARL might play out, based on the revised user experience flow:

For New Users:

1. Voice Initial Contact:
    - PEARL: "Hello and welcome! I'm PEARL, your guide today. Would you prefer to keep chatting here or switch to text for a more discreet interaction? If text, what's your phone number?"
    - User: "Let's switch to text, my number is [phone number]."
    - PEARL: "Great, I'll send you a message right now. See you there!"
2. Text Interaction Begins:
    - PEARL via Text: "Hi again! To get started, can you share what you're most looking forward to at this event?"
    - User: "I'm looking to meet other entrepreneurs."
    - PEARL: "Perfect! There's a networking session for entrepreneurs starting in 30 minutes. I can introduce you to some folks."

For Returning Users:

1. Identifying Returning User:
    - PEARL: "Welcome back! Last time you were interested in entrepreneurship events. Are you looking for something similar today, or is there something new on your mind?"
    - User: "I'm exploring investment opportunities now."

- PEARL: "Excellent! There's a panel discussion on investment trends today. Would you like to join?"

During the Event:

1. Encouraging Participation and Photo Sharing:
    - PEARL via Text: "I hope you're enjoying the panel! If you're having a great time, why not share a photo? I'd love to include it in our event highlights."
    - User sends a photo.
    - PEARL: "Thanks for sharing! This looks fantastic. I'll make sure it gets featured."

End of the Event:

1. Photo Sharing and Final Messages:
    - PEARL via Text: "As we wrap up, here's a look back at some great moments from today [link to photo highlights]. It was wonderful having you!"
2. Follow-Up Post-Event:
    - PEARL: "I hope you made some valuable connections today. Would you like a reminder about our next event or perhaps a follow-up with someone you met today?"

Post-Event Engagement:

1. Continued Interaction:
    - PEARL: "Just checking in to see how you are doing after our event. Any thoughts or feedback you'd like to share?"
    - PEARL: "I'm about to go offline, any last messages you want to share with me or host?"

Feature Details

AI-Powered Socialization

- **Personalized Introductions:** Utilizing OpenAI's natural language processing capabilities, PEARL will analyze attendees' registration information to match them based on shared interests and compatible social styles. This will facilitate introductions both digitally (via text or voice chat) and in-person at events.

- **Event Customization:** PEARL can be tailored to specific event themes and goals, adapting its interaction strategies and conversation topics accordingly. For business conferences, it might focus on professional interests, while at social gatherings, more personal interests could be highlighted.

Data Storage and Management

- **Guest Information:** PEARL will collect data during the registration phase and through ongoing interactions. This includes basic demographics, contact information, interests, and preferences for socializing.
- **Interaction Logs:** All interactions facilitated by PEARL will be logged to refine future event strategies and improve AI responses. This data will be crucial for understanding the effectiveness of introductions and for ongoing system training.
- **Event-Specific Data:** Details about each event (e.g., theme, size, venue) will be stored to customize the AI's approach and ensure relevance in its interactions.

Privacy and Safety

- **Secure Data** Handling: Implement robust security protocols to protect personal data. All data exchanges between the front-end and back-end will be encrypted, and access to sensitive data will be restricted to authorized personnel only.
- **User Consent and Preferences:** Users will have control over the information they share and can adjust their privacy settings at any time. PEARL will respect these preferences in all interactions and data storage.

User Interaction Enhancements

- **Gamified Engagement:** Introduce game-like elements (e.g., challenges, leaderboards) based on event participation, encouraging more interaction and making networking fun.
- **Photo and Message Sharing:** Allow guests to share photos and messages through the platform, enhancing the social experience and creating shared memories.

Architecture Overview

System Architecture

- **Front-end:** The user interface will be accessible via web browsers, optimized for both mobile and desktop use. It will provide interfaces for text and voice interactions, supported by a responsive design to ensure a seamless user experience across devices.
- **Back-end:** The server-side will handle processing, AI interactions, data management, and integration with external services like MongoDB for data storage and Twilio for communication.
- **APIs:** The system will interact with OpenAI for natural language processing and understanding, and MongoDB APIs for database interactions. These APIs will allow the system to retrieve and store data efficiently and securely.

Technology Stack

- **JavaScript/Node.js:** The primary programming language and runtime environment, used for both front-end and back-end development.
- **Express.js:** This web application framework for Node.js will be used to handle server-side logic, routing, and interaction with APIs.
- **MongoDB:** Chosen for its flexibility with dynamic schemas, making it suitable for the varied and potentially unstructured data generated by event interactions.
- **Twilio:** For handling real-time communication needs, including sending and receiving text messages and managing voice interactions.
- **OpenAI:** Leveraged for its powerful language understanding capabilities, enabling PEARL to interact naturally with users and provide personalized event experiences.
- **VapiAI:** Used to manage voice to voice interactions and connect with the Twilio messaging system seamlessly.
- **Playht:** Used to dynamically generate the AI voice to have natural emotional resonance and conversational pauses.

Data Storage

Database System: MongoDB, a NoSQL database, is chosen for its flexibility in handling unstructured data and its ability to scale efficiently with our application needs.

Data Schema

- **Guest Profiles:** Each guest's profile will contain fields for name, contact information, social preferences, interests, and a log of interactions at different events. This is structured like:

```
{
  "name": String,
  "phone_number": String,
  "email": String,
  "interests": [String],
  "social_style": String,
  "event_history": [{
    "event_id": ObjectId,
    "interactions": [ObjectId]
  }]
}
```

Event Data: Information about each event, such as location, date, theme, and specific AI customization options.

```
{
  "event_id": ObjectId,
  "name": String,
  "date": Date,
  "location": String,
  "theme": String,
  "guests": [ObjectId]
}
```

Interaction Logs: Records of interactions facilitated by PEARL, including

conversation logs and outcomes.

```
{
  "interaction_id": ObjectId,
  "guests_involved": [ObjectId],
  "timestamp": Date,
  "conversation": [String],
  "outcome": String
}
```

Data Flow

- **Data Collection:** Data will be collected through Pearl using natural conversational chat and ongoing interactions at events. This includes both automated data collection through AI interactions and manual inputs by guests.
- **Data Usage:** Data is primarily used to improve guest experiences at current and future events by making more accurate and personalized introductions.
- **Data Updates:** Guests can update their profiles to reflect changes in their interests or social preferences, ensuring that the AI's recommendations remain relevant and effective.

Security Measures

- **Encryption:** All sensitive data will be encrypted both in transit and at rest.
- **Access Control:** Strict access controls will be implemented to ensure that only authorized personnel can access sensitive data. Regular audits will be conducted to maintain security standards.

Technical Implementation Overview

Backend Setup

1. Session Management:
    - Implement middleware in Node.js using Express to handle session cookies. This middleware will create, manage, and validate session cookies for each user interaction.

    - Use the `cookie-parser` package to simplify handling cookies, allowing the server to read the values stored in cookies and determine whether a user is new or returning.
2. Database Interactions:
    - Use MongoDB to store user data, interaction logs, and event details. Set up schemas as previously defined to include user profiles, event data, and interaction records.
    - Implement functions to retrieve user data based on session IDs stored in cookies, and to update user data dynamically based on interactions during an event.
3. API Endpoints:
    - Develop RESTful API endpoints to handle requests such as user information updates, event details retrieval, and photo uploads.
    - Ensure these endpoints are secure and can handle high volumes of requests efficiently, particularly during large events.

Frontend Interaction

1. Voice and Text Integration:
    - Implement a frontend system that can initiate and manage both voice and text interactions. Use WebSockets for real-time communication to ensure that messages are sent and received without delay.
    - Integrate with speech-to-text services if voice interactions are not directly handled by the backend AI system.
2. Dynamic Content Loading:
    - Use JavaScript to dynamically update the content based on user interactions and backend responses. Ensure that the user interface reflects changes in real-time, enhancing the responsiveness of the application.

Security Considerations

1. Data Encryption:
    - Encrypt sensitive data both in transit and at rest. Use HTTPS to secure all communications between the client and the server.
    - Implement encryption at the database level for stored data, especially personal information and interaction logs.
2. Authentication and Authorization:
    - Secure API endpoints using authentication mechanisms such as OAuth or API keys.

- Implement role-based access control to ensure that only authorized personnel can access sensitive data or administrative functions.

Testing and Quality Assurance

1. Unit Testing:
    - Write unit tests for both frontend and backend components to ensure that each part of the application functions as expected.
    - Use testing frameworks like Mocha and Chai for backend testing and Jest for frontend testing.
2. Integration Testing:
    - Conduct integration tests to ensure that different parts of the system work together seamlessly. This includes testing API endpoints, database interactions, and real-time communication channels.
3. Load Testing:
    - Perform load testing to simulate the expected number of users during events to ensure the system can handle high traffic without performance degradation.