**Gamblers-Anonymous High-Level Design Security – Dakota List**

User login/registration should be designed and implemented with the following considerations:

**User Login/Registration:**

1. **Account Creation:**
   * **Invite Code Verification:** Require users to enter a valid invite code to create an account. This helps control the number of accounts and prevents unauthorized access.
   * **Unique Username and Email:** Enforce unique usernames and email addresses for each account. This ensures user identity and prevents account duplication.
   * **Strong Password Requirements:** Implement strong password requirements, including minimum length, complexity, and password history. This protects user accounts from brute-force attacks and weak passwords.
2. **User Authentication:**
   * **Secure Login:** Implement secure login mechanisms, such as salted and hashed passwords, to protect user credentials.
   * **Multi-Factor Authentication (MFA):** Offer MFA options, such as email verification or SMS codes, to add an extra layer of security.
   * **Login Throttling:** Implement login throttling limits to prevent brute-force attacks.
3. **Session Management:**
   * **Session Timeouts:** Set appropriate session timeouts to automatically log out inactive users.
   * **Secure Session Tokens:** Use secure session tokens and invalidate tokens upon logout or session expiration.
   * **Prevent Session Hijacking:** Implement measures to prevent session hijacking, such as session cookies with secure flags and HTTP Strict Transport Security (HSTS).

**Password Resets:**

1. **Password Reset Requests:**
   * **Verify User Identity:** Verify user identity before initiating a password reset. This can be done by sending a verification code to the user's registered email address.
   * **Time-Sensitive Reset Links:** Generate time-sensitive reset links to prevent unauthorized password changes.
   * **Prevent Password Reuse:** Discourage password reuse by requiring users to create a new password that is different from their previous passwords.
2. **Password Storage:**
   * **Never Store Plaintext Passwords:** Never store passwords in plaintext. Always store passwords in a securely hashed format.
   * **Use Strong Hashing Algorithms:** Use strong hashing algorithms, such as bcrypt or Argon2, to protect password hashes.
   * **Regular Password Hashing Updates:** Consider regularly updating password hashes with newer, more secure hashing algorithms.

**Invite Code System:**

1. **Invite Code Generation:**
   * **Unique Invite Codes:** Generate unique and unpredictable invite codes to prevent unauthorized account creation.
   * **Invite Code Expiration:** Set expiration dates for invite codes to limit their usage and prevent code sharing.
   * **Invite Code Revocation:** Allow administrators to revoke invite codes if they are compromised or no longer needed.
2. **Invite Code Usage:**
   * **Invite Code Validation:** Validate invite codes before allowing account creation.
   * **Invite Code Consumption:** Mark invite codes as used after successful account creation to prevent multiple accounts per code.
   * **Invite Code Tracking:** Keep track of invite code usage to monitor account creation and identify potential misuse.