Functional Requirements

1. **User Management:**
    a. Users can create an account by providing necessary details (e.g., username, email, password).
    b. Users can log in and log out of the system.
    c. Users can view and update their personal information on the "About Me" page.
2. **Role Management:**
    a. The system must differentiate between users with "user" and "admin" roles.
    b. Only admins can create, update, or delete tournaments and categories.
3. **Tournament Management:**
    a. Users can view a list of all tournaments on the home page.
    b. Users can click on a tournament to view detailed information.
    c. Users can register for a tournament.
    d. Users can withdraw their registration from a tournament.
4. **Category Management:**
    a. Only admins can create categories.
    b. Admins must assign at least one category when creating a tournament.
5. **Registration Management:**
    a. Users can view the list of tournaments they are registered for.
    b. Users can withdraw their registration from a tournament with a single click.
6. **Access Control:**
    a. Admins have exclusive access to create, update, and delete tournaments and categories.
    b. Users with the "user" role can only register for or withdraw from tournaments and view their personal information.
7. **Authentication and Authorization:**
    a. The system must authenticate users before allowing access to certain features.
    b. Users can only perform actions based on their assigned roles (admin or user).
8. **Tournament Details Page:**
    a. Users can view detailed information about a tournament, including categories, dates, and location.

Non-Functional Requirements

1. **Security:**
    a. User passwords must be stored securely using hashing.
    b. The system should enforce strong password policies (e.g., minimum length, complexity).
    c. Access to admin functionalities must be restricted to authenticated and authorized admin users.
    d. Session management must be implemented to prevent unauthorized access, with sessions expiring after a period of inactivity.
2. **Performance:**
    a. The system should handle multiple concurrent users without significant performance degradation.
    b. The home page should load within 2-3 seconds, even with a large number of tournaments listed.
    c. Database queries, especially for retrieving tournaments and user data, should be optimized for quick response times.
3. **Scalability:**
    a. The system should be scalable to handle an increasing number of users and tournaments.
    b. The application should support scaling both vertically (upgrading server resources) and horizontally (adding more servers).
4. **Usability:**
    a. The user interface should be intuitive and easy to navigate for both players and admins.
    b. The design should be responsive, ensuring usability across different devices (desktop, tablet, mobile).
    c. The system should provide clear feedback messages (e.g., successful registration, error messages).
5. **Reliability:**
    a. The system should have a high uptime (e.g., 99.9% availability).
    b. The application should handle unexpected errors gracefully and provide meaningful error messages to users.
6. **Maintainability:**
    a. The codebase should be well-documented to facilitate future maintenance and updates.

   b. The application should follow a modular design, allowing for easy addition of new features or modifications.

7. **Data Integrity:**
   a. The system must ensure that data is consistently and accurately maintained, particularly for user registrations and tournament details.
   b. Transactions (e.g., registration and withdrawal) should be handled in an atomic manner to prevent data inconsistencies.