

User Stories

INITIAL PLAN (FINAL PLAN IS ON PAGE 3)

1. Place Bets on Matches

- As a sports enthusiast, I want to select upcoming matches and place bets on the team or player I believe will win, So that I can engage in the excitement of predicting outcomes and potentially earn rewards.

Acceptance Criteria:

- User can browse a list of upcoming matches across different sports.
- User can select a match and place a bet by choosing a team or player.
- User can specify the amount of SOL they want to wager.
- A confirmation screen is shown before the bet is finalized.

2. Receive Real-Time Updates

- As a bettor, I want to receive real-time updates on the matches I have bet on, So that I can track the progress of my bets and stay informed about the outcomes.

Acceptance Criteria:

- Live match updates are displayed to the user.
- Users can see how their bets are performing in real-time.
- Notifications are sent for significant events (e.g., goal scored, match result).

3. Manage and Withdraw Winnings

- As a winner, I want to withdraw my winnings to my connected Solana wallet, So that I can access my earnings whenever I want.

Acceptance Criteria:

- Users can view their total winnings.
- Users can initiate a withdrawal to their Solana wallet.
- The withdrawal process is secure and efficient.

4. Bet Pool Management

- As a platform operator, I want to securely store the SOL wagered by users whose bets did not succeed, So that the funds can be used to enhance the platform and fund future events.

Acceptance Criteria:

- Unsuccessful wagers are automatically added to a secure pool.
- The pool is managed transparently and used for platform improvements.

5. Community Engagement

- As a user, I want to engage with a community of fellow sports enthusiasts, So that I can share my predictions, discuss match outcomes, and celebrate victories together.

Acceptance Criteria:

- Users can post and share their predictions within the platform.
- Users can participate in discussions about ongoing or completed matches.
- Community features are integrated seamlessly with the betting experience.

6. Easy Wallet Integration

- As a user, I want to connect my Solana wallet to the platform easily, So that I can start betting without hassle and manage my funds securely.

Acceptance Criteria:

- Users can connect their preferred Solana wallet.
- The wallet connection process is simple and secure.
- Users can view their wallet balance and transaction history within the platform.

Prioritization

- **Core Features:** Place Bets, Receive Real-Time Updates, Manage and Withdraw Winnings.
- **Secondary Features:** Bet Pool Management, Easy Wallet Integration.
- **Extras:** Community Engagement.

FINAL PLAN OF USER STORIES

A) User-Centric Focus

User stories should clearly define who the user is, what functionality they need, and why it matters to them. This ensures that every feature is developed with a specific user and their needs in mind, making the product more relevant and valuable.

Example:

- **User:** Sports Enthusiast
- **Functionality:** Place bets on matches
- **Benefit:** Engage in the excitement of predicting outcomes and potentially earn rewards.

By focusing on the user's perspective, the story highlights the value the feature brings to the user, ensuring that development efforts are aligned with user expectations.

B) User Identification

It's essential to identify the different types of users involved and their specific roles. This helps in creating user stories that are targeted and relevant, ensuring that the platform meets the needs of all stakeholders.

Example:

- **Primary User:** Bettor (Sports Enthusiast)
- **Secondary User:** Platform Operator (Manages the platform and the wager pool)

For your platform:

- Bettors are the main users who place bets, track outcomes, and withdraw winnings.
- Platform Operators manage the platform, ensuring bets are processed, funds are handled securely, and the community is engaged.

Defining these roles helps to tailor features to each user type, ensuring that all user needs are addressed.

C) Single Feature Focus

Each user story should focus on a single feature or functionality to keep the story simple, manageable, and easy to implement. This prevents scope creep and makes it easier to track progress.

Example:

- **Feature:** Real-Time Updates
- **Story:** "As a bettor, I want to receive real-time updates on the matches I have bet on, so that I can track the progress of my bets and stay informed about the outcomes."

By focusing on one feature per story, you can better manage the development process, break down epics into smaller tasks, and ensure that each feature is developed thoroughly.

D) Acceptance Criteria

Acceptance criteria define the conditions that must be met for a user story to be considered complete. This ensures clarity and alignment on what "done" means for each feature, reducing misunderstandings and rework.

Example:

For the "Place Bets on Matches" story:

- **Acceptance Criteria:**
 - User can browse a list of upcoming matches across different sports.
 - User can select a match and place a bet by choosing a team or player.
 - User can specify the amount of SOL they want to wager.
 - A confirmation screen is shown before the bet is finalized.

These criteria make it clear what needs to be implemented and tested, ensuring the feature is fully functional before it's marked as complete.

E) Testability

User stories should be written in a way that allows for easy testing. If a story is too complex or broad to test effectively, it likely needs to be broken down into smaller, more manageable parts.

Example: For the "Easy Wallet Integration" story:

- **Testability:**
 - Can the user connect their Solana wallet?
 - Is the connection process secure?
 - Can the user view their wallet balance and transaction history within the platform?

If any part of this story is difficult to test, it might indicate that the story needs to be split into smaller, more specific tasks, such as "User can view wallet balance" and "User can view transaction history."

F) Prioritization

Not all features can be developed at once, so it's important to prioritize them based on user needs, business value, and technical dependencies. Prioritizing stories helps ensure that the most critical features are developed first, reducing the risk of delays and rework.

Example:

- **Core Features (High Priority):** Place Bets, Real-Time Updates, Manage and Withdraw Winnings.
- **Secondary Features (Medium Priority):** Bet Pool Management, Easy Wallet Integration.
- **Extras (Low Priority):** Community Engagement.

By prioritizing in this way, you can ensure that the essential features that form the backbone of your platform are developed first, allowing for a functional MVP before adding more advanced features.

G) Non-Functional Requirements

In addition to functional requirements (what the system should do), it's important to include non-functional requirements (how the system should perform). These include performance, security, usability, and scalability, ensuring that the platform meets operational expectations.

Example:

- **Performance:** The platform should handle real-time updates without lag, even during peak betting periods.
- **Security:** The platform must ensure secure wallet integration and transactions, protecting user funds and data.
- **Scalability:** The platform should be able to scale to accommodate a growing number of users and matches without performance degradation.
- **Usability:** The platform should offer an intuitive interface that makes placing bets and managing funds easy for all users.

Including these non-functional requirements ensures that your platform is not only feature-rich but also reliable, secure, and user-friendly, providing a high-quality experience for your users.