GhostCheck - An Anonymous PoW

(Name changed from Dev-Sentry ZK)

User Stories & On-Chain Requirements Document

Part A: Finalized User Personas & Function Maps

1. Core User Personas

Persona	Role	Description
The Verified Prover	Direct User	A developer or creator who wants to turn their real-world work history (e.g., GitHub, X) into a portable digital credential without revealing their personal identity.
The Gatekeeper	Beneficiary	A recruiter, community moderator, or DAO leader who needs to verify technical skills or ownership before granting access or opportunities.
The Protocol Admin	Administrator	The entity responsible for managing the "mathematical rules" (Verification Keys) and ensuring the ZK system stays secure as platforms update.

2. Core Function Map

- **The Verified Prover:**
    - Connects to a Web2 platform (like GitHub) via a secure button or QR code.
    - Generates a private "Ghost-Seal" (a mathematical proof of work).
    - Claims a permanent, non-transferable digital badge on the Solana blockchain.
- **The Gatekeeper:**
    - Checks a user's wallet for a specific "GhostCheck" badge.
    - Confirms the authenticity of the work (e.g., "This user owns a DeFi repo") without seeing the user's private profile details.

3. Core POC Requirements

Based on the "Verify and Mint" path, the following technical requirements are essential for the Proof of Concept:

- **zkTLS Witnessing:** Integration with Reclaim Protocol to capture and sign the HTTPS session data from GitHub.

- **ZK-Redaction Logic:** An SP1 Guest program that verifies the Reclaim signature and hides the username while making the repository data public.
- **On-Chain Verification:** A Solana program using the sp1-solana verifier to confirm the mathematical validity of the proof.
- **Soulbound Minting:** A process to create a **Metaplex-core ASSET** upon successful proof verification.

Part B: Potential On-Chain Requirements

User Story 1: "User creates a Proof of Work claim."

- **Math Check:** The blockchain must verify the ZK proof to ensure the work is authentic and follows protocol rules.
- **Wallet Binding:** The system must ensure the badge is locked to the specific wallet that generated the proof, preventing others from "stealing" the claim.
- **Identity Protection:** The system must ensure that no personal data (like a GitHub username) is ever written to the blockchain.

User Story 2: "User receives a permanent, private badge."

- **Anti-Fraud (Single Use):** The blockchain needs a way to "remember" that a specific repository has already been used to claim a badge so it cannot be reused.
- **The Reward:** Once the math is cleared, a digital badge must be created and sent to the user's wallet.
- **Soulbound Rule:** The badge must be permanently attached to the user's wallet—it cannot be sold, traded, or moved.

Part C: Process Appendix (Refinement Log)

1. Initial User & Function Mapping

- **Brainstorming:** I initially listed specific users like "Solana Developers" and "Web3 Recruiters." After reviewing the value proposition, I expanded these to the broader categories of "The Prover" and "The Gatekeeper" to reflect a universal reputation system.
- **AI Feedback:** The AI recommended focusing on the Developer and the Recruiter as the most critical personas for the POC to prove "Supply" and "Demand" for credentials.
- **Decision:** I agreed with the AI but added the "Protocol Admin" because the management of the mathematical "Verification Keys" is a critical technical requirement for the POC's success.

2. Adversarial Analysis & Granularity Check

- **AI Critique:** The AI noted that "verifying a repo" was too broad and could lead to "Double Minting" (one repo used for many badges).
- **Refinement:** I added an "Anti-Fraud" requirement to the on-chain logic to ensure each repository hash can only be used once.

3. Clarity & Refinement Log

Before (Technical/Jargon)	**After**	**Rationale for Change**
"Generate ZK Proof via SP1 Guest."	"Create a private mathematical 'seal' that confirms work without showing a name."	Simplified for non-technical stakeholders.
"Verify Nullifier on-chain."	"The system remembers used work to prevent cheating."	"Nullifier" is too technical; "preventing cheating" is the clear goal.
"Verify ed25519 signature."	"Confirm the digital stamp from GitHub is genuine."	Used a familiar analogy (stamp) to explain cryptographic verification.
"User verifies and receives a badge."	**Story 1:** "System checks the proof." **Story 2:** "System gives a badge."	Split into two stories for atomicity (one action per story).

Part B: Process Appendix

1) Initial Mapping Log (Brainstorming & Prioritization)

- **The Brainstormed List:** I initially considered everyone from solo developers and recruiters to job-board owners, protocol investors, and bounty-platform admins.
- **AI Prompt:** "My project's value proposition is a universal Proof of Work (PoW) claim system that hides identity. Based on this, which 2-5 user types are critical for a POC?"
- **AI Recommendation:** The AI suggested focusing on the **Developer** (the source of truth) and the **Recruiter** (the validator).
- **My Analysis:** I agreed but refined "Recruiter" into **"Gatekeeper"** because this tool is for more than just jobs—it's for DAOs, private alpha groups, and anywhere that requires reputation. I also added the **Admin** as they are the only ones who can update the system if GitHub's security changes.

2) Adversarial Analysis (The "What If?" Check)

- **The Prompt:** "Review my core requirements. If I am a malicious user, how can I break this system or look better than I actually am?"
- **The Finding:** During this analysis, we found the **"Double-Minting" Threat**. A user could take one impressive repository and use it to mint 10 different badges to 10 different wallets.
- **The Refined Requirement:** This led to the creation of the **"Nullifier" Requirement** in Part D. The blockchain now "remembers" every repo hash, ensuring it can only be used once across the entire protocol.

3) Refinement Log (The "De-Jargon" Table)

I manually reviewed my technical requirements to ensure they are understandable by anyone, regardless of their knowledge of Zero-Knowledge proofs.

Before (Technical Term)	After (Simplified Action)	Rationale for Change
"Generate ZK Proof via SP1 Guest."	"Create a private mathematical 'Ghost-Seal'."	"Seal" sounds more like a credential; "SP1" is a technical detail.

"Verify Nullifier on-chain."	"The system remembers used work to prevent cheating."	Explains the *why* instead of the *how*.
"CPI to Metaplex for minting."	"The system creates a permanent digital badge in the user's wallet."	Focuses on the user's reward, not the tool used to make it.
"Verify ed25519 signature."	"Confirm the digital stamp from GitHub is genuine."	Uses a familiar concept (stamps) for authenticity.