**Project Description:**

Paragon Devs is a mobile and web platform designed to streamline freelance collaborations by connecting skilled professionals (e.g., designers, developers) with clients through intuitive job postings, proposals, and payment tracking. Addressing pain points like fragmented communication, unclear project scopes, and payment insecurity, the app integrates HCI principles to prioritize transparency, efficiency, and trust. Target users include freelancers and clients who value seamless project management and secure transactions.

**Requirements Summary:**

|  |  |  |
| --- | --- | --- |
| **MINIMUM**  **REQUIREMENTS** | Processor | Single Core 1.0 GHz |
| OS | Android 5.0 (Lollipop) API Level 21 |
| RAM | 2 GB |
| Storage | 200 MB available space |
| Network | Not required (offline functionality) |
| **RECOMMENDED REQUIREMENTS** | Processor | Quad Core 1.4 GHz |
| OS | Android 7.0 (Nougat) API Level 24 |
| RAM | 4 |
| Storage | 1.5GB available space |
| Network | Wi-Fi or mobile data (for cloud sync) |

Table 1. System Requirements

Paragon Devs App are the technical specifications needed for the app to function properly on Android devices.

**Evaluation Technique Used:**

1. **Task-Based Testing:**

Freelancers were asked to submit proposals and update their profiles, while clients posted jobs and reviewed bids to validate key workflows.

1. **Heuristic Evaluation:**

Using Nielsen’s 10 Usability Heuristics, the design was assessed to identify interface and interaction issues before user testing.

1. **Post-Test Survey:**

A 4-point Likert scale with open-ended questions was used to measure user satisfaction and collect qualitative feedback.

**Data Analysis  
Usability Specifications:**  
Usability Specifications for the Paragon Devs freelancer app prototype, structured to ensure optimal user experience: **Efficiency**

* Users (freelancers/clients) can complete key tasks (browsing jobs, submitting proposals, managing profiles) in 3 steps.
* Search results load in 1 second with filters applied.

**Learnability**

* New users understand core functionality (Make a proposition) within 1 minute of onboarding.
* Icons/CTAs (UI tags, View Portfolio) follow platform conventions for instant recognition.

**Accessibility**

* Contrast ratios meet standards (text on buttons, review cards).
* Screen readers support all critical actions (job posts, profile edits).

**Error Prevention**

* Clear validation messages guide users when submitting proposals (character limits, file formats).
* Irreversible actions deleting projects require confirmation.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Avg. Time** | **Success Rate** | **Errors Noted** |
| Submit Proposal | 1.2 min | 90% | File upload lacked progress indicators. |
| Post Job | 45 sec | 100% | None. |
| Locate Total Gains | 2.1 min | 60% | Buried in dashboard hierarchy. |

*Table 2. Task time  
Paragon Dev testing revealed strong performance in core workflows, with a 90% success rate for proposal submissions and 100% for job postings, though users struggled to locate the "Total Gains" metric (60% success). Survey feedback highlighted high satisfaction with navigation (3.8/4) but identified trust gaps in payments (3.2/4), citing unclear escrow processes. Heuristic evaluation uncovered consistency issues (e.g., mixed button labels) and missing error safeguards (e.g., no proposal drafts). These findings prioritized improvements in dashboard clarity, payment transparency, and error handling to enhance usability.*

**Heuristic Evaluation:**

|  |  |
| --- | --- |
| **Heuristic** | **Application in Paragon Dev** |
| **Visibility of system status** | Clear deadlines and payment confirmations inform users, though loading indicators during uploads would help. |
| **Match to real world** | Natural messaging and industry terms |
| **User control and freedom** | Consistent Back buttons provide freedom, but adding proposal cancellation would enhance control. |
| **Consistency and standards** | Uniform CTAs work well, though table formats need standardization. |
| **Error prevention** | File uploads reduce errors, but character limits for proposals would prevent oversights. |
| **Recognition, not recall** | Visible skills and message history aid memory, while full project history would complete context. |
| **Flexibility and efficiency** | Quick upload file actions shine, saved templates could further speed repetitive tasks. |
| **Aesthetic, minimalist design** | Clean layouts highlight key data like budgets, though redundant "Back" buttons clutter tables. |

**Feedback**The evaluation collected valuable insights from both freelancers and clients, with participants praising the intuitive job posting process and proposal system while highlighting areas for improvement in payment transparency and navigation clarity. Some users expressed confusion about financial metrics like "Total Gains," and others requested clearer explanations of the escrow system. The feedback also noted minor UI inconsistencies in button labels and layout. These responses provided a balanced view of the prototype's strengths and opportunities for refinement.  
  
**Does the prototype need to be altered based on the feedback?**

Yes, the prototype requires targeted modifications to address user concerns, particularly in improving payment transparency and dashboard organization. Key changes include relabeling ambiguous terms like "Total Gains" to "Earnings" and adding tooltips to explain the escrow process. Additionally, UI inconsistencies, such as mixed button labels, will be standardized to enhance usability. These adjustments aim to build trust and streamline navigation based on direct user feedback.

**What improvements needed to the design?**Based on feedback, we need to enhanced the dashboard by relabeling Total Gains to Earnings for clarity and added progress indicators for file uploads. And need to standardized UI elements like button labels and improved payment transparency with escrow explanations.  
  
**What would you have done differently, knowing what you know now?**With more time and resources, we would conduct in-person testing to observe physical interactions and expand the participant pool for more statistically significant results. A higher-fidelity prototype with live payment simulations would have provided deeper insights into trust-building features. We would also iterate on the design twice testing before and after refinements to measure improvement effectiveness more accurately.  
  
**Summary of the Project**The Paragon Devs prototype successfully validated core functionalities, such as job posting and proposals, with high usability scores, while also identifying key areas for improvement in payment transparency and navigation. Remote testing provided actionable insights, though limitations in prototype interactivity and sample size underscored the need for further refinement. Planned enhancements focus on UI consistency, trust-building features, and user guidance to elevate the platform. The project highlighted the importance of iterative testing and user-centered design in creating an effective freelancer solution.