FineAnnotator

Business Requirement Specification (BRS)

Version: 1.0

Collaborators

Nazmun Nahar Tui - **2018331047**
Nishat Rahman - **2018331071**
Nafis Fuad Abir - **2018331087**
Mirza Nihal Baig - **2018331095**
Nazmus Sadat - **2018331099**

1. Introduction

FineAnnotator is a web-based platform designed to bridge the gap between researchers/companies seeking data annotation services and students looking for flexible part-time work.

2. Business Objectives and Goals

For researchers and companies:

- Simplify and streamline the process of finding and hiring qualified data annotators.
- Access a large and diverse pool of potential annotators.
- Ensure high-quality data annotation by setting specific requirements and reviewing annotator profiles.
- Streamline payment processing through the platform.

For students:

- Find flexible and rewarding part-time work opportunities.
- Utilize their skills and knowledge to contribute to real-world research and projects.
- Build their profiles and CVs with relevant experience and feedback.

For FineAnnotator:

- Become the leading platform for connecting researchers/companies and data annotators.
- Generate revenue through a fee-based system for researchers/companies.
- Foster a vibrant community of researchers, companies, and students.

3. Target Users and Needs

3.1 Researchers/Companies:

- Need large amounts of high-quality annotated data for various purposes, such as machine learning, deep learning, and natural language processing.
- Require efficient ways to find, hire, and manage annotators.
- Need to ensure data security and privacy.

3.2 Students/Annotators:

- Seek flexible part-time work opportunities to earn income.
- Want to utilize their skills and knowledge in a practical setting.
- Desire to build their resumes and gain valuable experience.

4. High-Level Functionalities and Features

4.1 For Researchers and Companies:

- Post data annotation jobs with detailed descriptions, requirements, and payment information.
- Search for and filter annotators based on skills, experience, and other criteria.
- Review annotator profiles and select the best candidates.
- Communicate directly with annotators through the platform.
- Track the progress of annotation tasks and receive updates.
- Manage payments securely through the platform.
- Rate and provide feedback to annotators.

4.2 For Students:

- Create and manage profiles with relevant skills, experience, and educational background.
- Search for available data annotation jobs based on various criteria.
- Apply for jobs and submit proposals.
- Communicate with researchers and companies through the platform.
- Access and complete assigned annotation tasks.
- Receive payments and view earnings history.
- Rate and receive feedback from researchers and companies.

4.3 For Admins:

- Monitor platform activity and user interactions.
- Approve or reject user accounts.
- Investigate and resolve disputes between researchers/companies and annotators.
- Manage platform settings and configurations.
- Generate reports and analyze platform performance.
- Address user feedback and suggestions.

5. System Model

5.1 Use-Case Diagram

There are different types of system models to present software. We have built a use-case diagram for our project. Below is a use-case diagram for our software, FineAnnotator. Here, you can see the user and their actions given altogether.

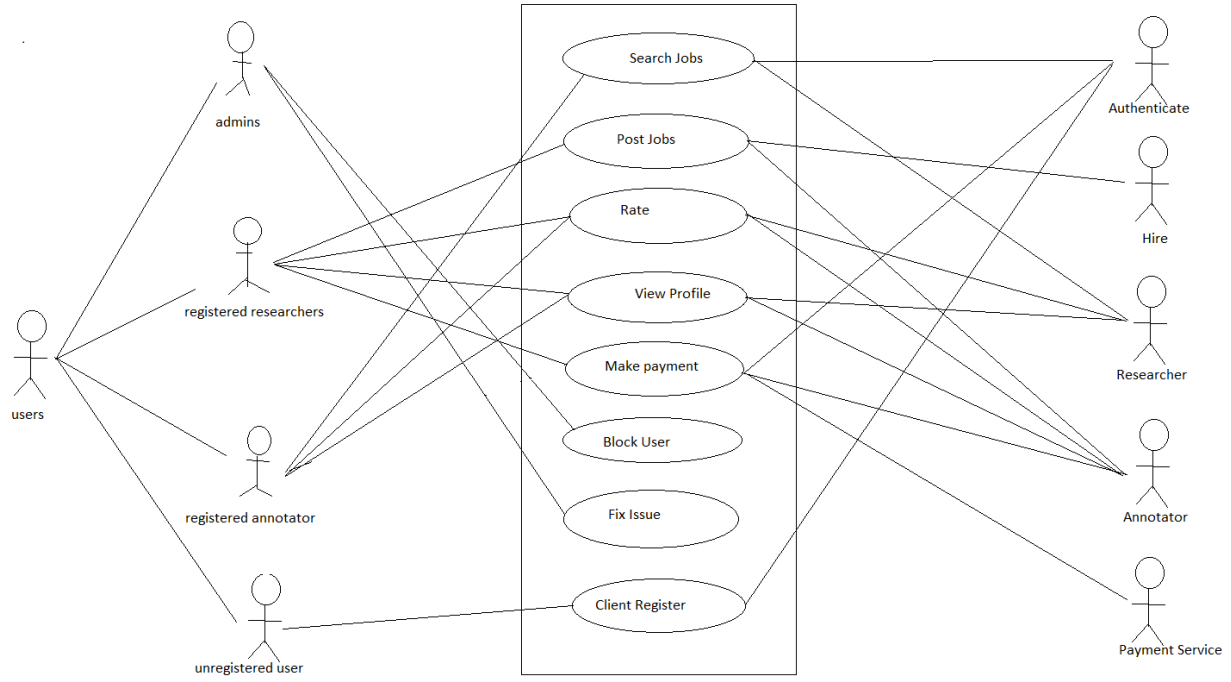

Figure: Use-Case Diagram for FineAnnotator

6. Functional and Non-Functional Requirements

FineAnnotator provides a lot of features as mentioned earlier. Below is given requirements for different type of features. Every feature or characteristic's requirements are divided into two main requirement categories.

1. Job Posting and Task Management:

Functional:

- Allow Researchers/Companies to create profiles and post annotation tasks categorized by data types.
- Enable specifying task details, requirements for annotators, payment structure, and deadlines.

Non-Functional:

- Accessibility: Ensure easy navigation and usability for posting tasks.
- Accuracy: Validate and ensure the accuracy of task details entered by Researchers/Companies.

2. Annotator Application and Profile Building:

Functional:

- Allow Annotators to create detailed profiles showcasing qualifications, skills, and experience.
- Provide a straightforward process for Annotators to apply for listed jobs and specify preferences.

Non-Functional:

- User-Friendliness: Ensure an intuitive interface for profile creation and job application.
- Performance: Ensure quick loading times for profile creation and job application sections.

3. Communication and Collaboration:

Functional:

- Implement a messaging system for interaction between Researchers/Companies and Annotators.
- Include a feedback and rating system for performance evaluation.

Non-Functional:

- Security: Ensure encryption and secure data transmission in the messaging system.
- Responsiveness: Ensure real-time or prompt notifications for messages and feedback.

4. Payment Handling and Security:

Functional:

- Develop a secure platform for financial transactions and dispute resolution.

Non-Functional:

- Security: Implement robust security measures to protect financial data and prevent fraud.
- Reliability: Ensure reliability in payment processing and transaction records.

5. Administrative Controls and Oversight:

Functional:

- Provide administrators with tools for monitoring, account management, and dispute resolution.

Non-Functional:

- Scalability: Ensure admin tools can handle increased usage and data processing as the platform grows.
- Accessibility: Ensure admin features are accessible and usable across devices and browsers.

6. User Experience and Interface:

Functional:

- Design an intuitive and easy-to-navigate platform for all user types.

Non-Functional:

- Usability: Ensure a consistent and user-friendly interface across different sections.
- Privacy: Provide robust privacy controls to protect user data and preferences.

7. Scalability and Future Enhancements:

Functional:

- Design a scalable platform to accommodate growth in user base and tasks.

Non-Functional:

- Flexibility: Ensure flexibility in the architecture for incorporating future updates and enhancements.

8. Legal and Compliance Considerations:

Functional:

- Ensure adherence to data privacy laws and regulations.

Non-Functional:

- Compliance: Maintain legal compliance by regularly updating the platform based on evolving regulations.

7. Budget and Financial Constraints

7.1. Project Budget:

Development Costs:

- Software Development: Estimate costs related to hiring developers, designers, and testers.
- Tools and Technologies: Budget for purchasing software licenses, development tools, frameworks, or third-party APIs.

Infrastructure Costs:

- Hosting and Server Expenses: Estimate expenses for cloud services, server rentals, or hosting platforms.
- Database Costs: Budget for database setup, maintenance, and any associated licensing fees.

Personnel and Operational Costs:

- Personnel Expenses: Include salaries, benefits, or contractor fees for project management and administrative tasks.
- Training and Onboarding: Budget for training sessions or workshops for the development team.

7.2. Financial Constraints:

Budget Limitations:

- Maximum Budget Allocation: Specify the maximum budget available for the project.
- Cost Control Measures: Define strategies or limitations to ensure the project remains within the allocated budget.

Resource Limitations:

- Staffing Constraints: If there's a limit on the number of developers or resources available for the project.
- Technology Budget Restrictions: Constraints on the usage of specific tools or technologies due to budget limitations.

Time Constraints:

- Time-to-Market Pressure: Describe any urgency to launch the software within a specific timeframe, potentially affecting budget allocation.
- Deadline Sensitivity: Budget constraints due to fixed deadlines requiring adherence.

Risk Management:

- Contingency Plan: Include provisions for unexpected expenses or risks that might affect the project budget.
- Risk Mitigation Strategy: Outline strategies to address potential budget risks or overruns.

Cost-Benefit Analysis:

- Expected ROI: Evaluate the potential return on investment (ROI) or benefits expected from the project against the allocated budget.
- Value Proposition: Explain the value the software will bring to justify the budget and financial resources invested.

8. System Interfaces and Integrations

There some other systems that can be needed to be integrated into our system. As our system includes payments, we have to integrate a payment system into it. Like that, there are some other integrations that we could do. Such as:

- Payment gateway integration for secure and efficient payment processing.
- Cloud storage integration for storing and managing large amounts of data.
- Data annotation tools integration for streamlined and efficient annotation tasks.
- Social media integrations for user authentication and sharing information.

9. Software Limitations

Every software comes with some limitations. Our software also has some limitations. Below are some limitations of FineAnnotator.

- Security and privacy of user data and information.
- Scalability of the platform to accommodate a large number of users and data.
- Quality assurance of data annotations.
- Compliance with relevant data privacy regulations.

10. Conclusion

FineAnnotator has the potential to revolutionize the way data annotation is done by enabling efficient collaboration between researchers/companies and students. By focusing on user needs and delivering a robust and feature-rich platform, FineAnnotator can achieve its business objectives and become a leading player in the data annotation industry.