

Software Engineering Department  
ORT Braude College

Capstone Project Phase A – 61998

**TaskTenders**

An Application that Connects Users with Freelance Services

**Authors:**

Safa Khier – Safa.Khier@e.braude.ac.il

Layan Amer – Layan.Amer@e.braude.ac.il

**Supervisor:**

Dr. Zakharia Frenkel

**24-2-D-42**

**Table of Contents**

[**Abstract** 3](#_Toc177855589)

[**1. Introduction** 4](#_Toc177855590)

[**2. Related Work** 5](#_Toc177855591)

[*2.1 Current Demand for Freelancer Services* 5](#_Toc177855592)

[*2.2 Benefits of Freelance Platforms* 6](#_Toc177855593)

[*2.3 Risks and Security Concerns in Freelancer Platforms* 7](#_Toc177855594)

[*2.4 Existing Apps that Offer Freelance Services* 8](#_Toc177855595)

[**3. Background** 9](#_Toc177855596)

[*3.1 Project Origin and Inspiration* 9](#_Toc177855597)

[*3.2 Key Features of Successful Freelancer Platforms* 10](#_Toc177855598)

[*3.3 The combination between Fiverr and Freelancer.com* 11](#_Toc177855599)

[**4. Expected Achievements** 13](#_Toc177855600)

[*4.1 Outcomes* 13](#_Toc177855601)

[*4.2 Unique Features* 14](#_Toc177855602)

[*4.3 Criteria for Success* 15](#_Toc177855603)

[**5. The Process** 16](#_Toc177855604)

[*5.1 Research – User Needs Analysis* 16](#_Toc177855605)

[*5.2 Methodology and Development Process* 16](#_Toc177855606)

[*5.3 Development Phases* 17](#_Toc177855607)

[**6. Product** 18](#_Toc177855608)

[*6.1 Our Solution* 18](#_Toc177855609)

[*6.2 Requirements* 20](#_Toc177855610)

[*6.2.1 Functional Requirements* 20](#_Toc177855611)

[*6.2.2 Non-Functional Requirements* 20](#_Toc177855612)

[*6.3 System Functionality* 21](#_Toc177855613)

[*6.3.1 Client Functionality* 21](#_Toc177855614)

[*6.3.2 Freelancer Functionality* 22](#_Toc177855615)

[*6.3.3 Administrator Functionality* 23](#_Toc177855616)

[*6.4 Technology Choices* 24](#_Toc177855617)

[*6.4.1* *Back-end: Node.js* 24](#_Toc177855618)

[*6.4.2* *Back-end: TypeScript* 26](#_Toc177855619)

[*6.4.3 Database: MongoDB* 28](#_Toc177855620)

[*6.4.4 Front-end: Flutter* 30](#_Toc177855621)

[*6.4.5 Development Environment: Visual Studio Code* 31](#_Toc177855622)

[*6.5 Project Structure* 33](#_Toc177855623)

[*6.6 Algorithms* 36](#_Toc177855624)

[*6.7 Diagrams* 38](#_Toc177855625)

[*6.7.1 use case diagram* 38](#_Toc177855626)

[*6.7.2 Activity diagram* 39](#_Toc177855627)

[*6.8 Tests* 39](#_Toc177855628)

[*6.8.1 Login Tests* 40](#_Toc177855629)

[*6.8.2 User Tests* 40](#_Toc177855630)

[*6.8.3 Freelancer Tests* 41](#_Toc177855631)

[*6.8.4 Admin Tests* 41](#_Toc177855632)

[**7. GUI** 42](#_Toc177855633)

[*7.1 Login and Sign-Up Screens* 42](#_Toc177855634)

[*7.2 User Screens* 42](#_Toc177855635)

[*7.3 Freelancer Screens* 44](#_Toc177855636)

[**8. References** 46](#_Toc177855637)

# **Abstract**

There is an increasing demand for a platform that links freelancers to individuals who need various services. The changes in the gig economy cause this situation. TaskTenders aims at bridging this gap through its user-friendly mobile application which connects service providers with users seeking to get tasks done more effectively. For instance, it allows users to post job requests such as home repairs, cleaning or painting giving details of what they need and when they need it. Consequently, freelancers can look through these job postings in order to submit their bids and offer their services at competitive rates.

This platform exploits the latest technology to enable smooth communications between users and freelancers. Furthermore, it has strong functionalities like user registration and secure authentication among others. Customers can articulate what they need, while workers can present their skills and expertise in full detail. A safe payment gateway guarantees secure financial transactions; while a within app messaging system enhances lucidity between the involved parties. This mobile application also contains a method of rating and reviewing services delivered by freelancers as well as keeping up high service standards and promoting trustworthiness within its society.

TaskTenders also has the administrators panel where you can manage all user activities, check all transactions and handle disputes. The main aim here is to create an operational, dependable and secure ecosystem that will enhance user satisfaction thereby promoting gig economy. By providing this platform, TaskTenders enables users to find competent freelancers who suit their requirements while at the same time granting freelancers access to numerous job opportunities. This project is developed using a modern tech stack, including Flutter for the frontend, Node.js and Express.js for the backend, and MongoDB for data management.

# **1. Introduction**

In the dynamic landscape of the gig economy, the demand for reliable and skilled freelance workers is burgeoning. Individuals and businesses alike seek efficient and cost-effective solutions to fulfill their varied service needs, ranging from home repairs and cleaning to specialized tasks like wall painting. However, connecting these users with competent freelancers remains a significant challenge. Traditional methods of finding and hiring freelancers can be time-consuming, unreliable, and often lack transparency.

Exploring the intricacies of this problem, users often face difficulties in verifying the credibility and quality of freelancers. Miscommunications, inconsistent service quality, and concerns about secure transactions further complicate the process. On the other hand, freelancers struggle to find a steady stream of job opportunities that match their skills and availability. The absence of a streamlined platform that bridges this gap leaves both parties underserved and their needs unmet.

Given these challenges, we propose an innovative solution in the form of TaskTenders, a mobile application designed to connect users with freelance workers efficiently and securely. TaskTenders allows users to post detailed job requests, specifying their requirements, deadlines, and budget. Freelancers can then browse these postings and submit bids, offering their services at competitive rates. The application's advanced matching algorithm ensures that users are paired with freelancers who meet their specific needs and preferences.

TaskTenders incorporates robust features to enhance user experience and build trust. A secure authentication system safeguards user data, while a comprehensive review and rating system ensures accountability and maintains high service standards. The in-app messaging feature facilitates clear communication between users and freelancers, reducing the potential for misunderstandings. Furthermore, an integrated payment gateway ensures secure and seamless financial transactions.

By leveraging modern technology, TaskTenders aims to streamline the process of finding and hiring freelancers, providing a reliable platform where users can easily access skilled professionals, and freelancers can find job opportunities that suit their skills and schedules. This application not only simplifies the connection between service seekers and providers but also contributes to the growth and efficiency of the gig economy. Through TaskTenders, we aspire to foster a more interconnected and productive community, addressing the evolving needs of both users and freelancers in today's fast-paced world.

# **2. Related Work**

## *2.1 Current Demand for Freelancer Services*

The current demand for freelancer services among individual clients is growing as more people recognize the benefits of hiring independent professionals for personal projects and needs. This trend is largely driven by the diversity of skills that freelancers bring to the table, enabling individuals to access expert services that might otherwise be unavailable or unaffordable through traditional providers.

For example, individual clients often seek freelancers for home improvement projects, personal finance advice, fitness coaching, or even learning new skills such as a musical instrument or a foreign language. Freelancers offering personalized tutoring or bespoke art and craft services are especially popular, as clients look for customized solutions that cater specifically to their personal interests or needs. This one-on-one service model offered by freelancers is highly attractive because it allows for more tailored, flexible, and responsive interactions than those typically provided by larger organizations.

Moreover, the demand is bolstered by the personal touch that freelancers can offer. Unlike large companies, an individual freelancer is likely to deliver a more personalized service, focusing on building a relationship with the client to better understand and meet their unique requirements. This aspect is particularly appealing in fields like event planning or personal styling, where understanding the client's vision is essential.

Additionally, the growth in demand can also be attributed to the increasing awareness and acceptance of freelancing as a legitimate and professional option for various services. As societal attitudes shift, individuals feel more comfortable and secure engaging freelancers for both occasional needs and ongoing projects. This trust is crucial in fields such as personal healthcare or elderly care, where reliability and professionalism are paramount.

In conclusion, the current demand for freelancer services among individual clients is characterized by a desire for specialized, personalized, and flexible service delivery. This trend not only underscores the shifting dynamics in how services are procured and delivered but also highlights a broader move towards more individualized consumption patterns, where the personalization of services is becoming a key factor in client satisfaction.

## *2.2 Benefits of Freelance Platforms*

Freelance platforms offer a range of benefits that streamline the interaction between freelancers and clients, making these platforms essential to the modern freelance economy. One of the most significant advantages is accessibility. These platforms eliminate geographical barriers, allowing clients to tap into a global talent pool and freelancers to access opportunities worldwide. They also offer a diverse range of jobs across various industries, giving freelancers numerous opportunities to choose from, which enhances their career growth and job satisfaction.

Flexibility is another key benefit. Freelancers enjoy the autonomy to set their own work hours and choose their working location, offering a work-life balance often unattainable in traditional employment settings. Additionally, freelancers can select projects that align with their skills and interests, leading to greater job satisfaction and continuous professional development.

Reduced overheads are a critical advantage for both freelancers and clients. The absence of a need for physical office spaces cuts down on significant expenses, benefiting everyone involved. Digital tools provided by these platforms streamline project management, billing, and communication, enhancing operational efficiency and reducing costs further.

Innovation and speed are also facilitated by freelance platforms. The competitive nature of these platforms often results in a quicker turnaround of projects. Moreover, the diverse pool of talent available introduces innovative solutions and fresh perspectives to projects, driving creativity and efficiency. Overall, freelance platforms not only make hiring and managing freelance talent more convenient but also enhance the quality and speed of work, benefiting the broader economy.

## *2.3 Risks and Security Concerns in Freelancer Platforms*

Freelancer platforms, while offering numerous benefits, also come with inherent risks and security concerns that can impact both clients and freelancers. These platforms face significant challenges related to data security, payment security, and the integrity of interactions between the two parties.

Data Privacy and Security: One of the primary concerns is the risk of personal and financial data exposure. The platforms store sensitive information that, if compromised, could lead to significant privacy violations and financial fraud. This risk is exacerbated by the increasingly sophisticated techniques employed by cybercriminals targeting such platforms.

Payment Security: Another significant concern is related to the financial transactions that occur on these platforms. Both freelancers and clients are vulnerable to risks of non-payment, delayed payments, and fraudulent transactions. This can lead to financial instability for freelancers who rely on timely payments for their livelihood, and frustration for clients who expect reliable service delivery.

Account Takeover: The risk of unauthorized access to user accounts is a constant threat. Through tactics like phishing or other hacking methods, malicious actors can gain control of freelancer or client accounts, leading to data theft, financial loss, and damage to professional reputations.

Fraudulent Listings and Scams: These platforms can also be used as conduits for scams through fake job postings or profiles. These scams are designed to either siphon money directly or to gather sensitive information such as banking details or personal identification numbers.

Quality Assurance and Disputes: Disputes over the quality of work delivered or the terms of service can also pose risks, affecting the credibility and operational integrity of the platform. Such disputes may arise from miscommunication, dissatisfaction with services rendered, or disagreements over project specifications.

## *2.4 Existing Apps that Offer Freelance Services*

The market for freelance services features a variety of platforms that cater to a wide range of industries and specializations. Understanding the nuances of these platforms is essential for grasping market trends, user expectations, and identifying areas that are ripe for innovation.

Overview of Major Platforms:

1. Upwork: This platform offers a broad range of categories, from web development to customer service, and includes features such as job posting, freelancer bidding, direct messaging, and a payment protection plan. Its strengths lie in robust filtering options for finding freelancers and detailed freelancer profiles, along with a built-in work diary to ensure hours worked are hours paid. However, its high fees for both freelancers and clients can deter smaller projects or lower-budget clients.
2. Fiverr: Known for focusing on gigs, Fiverr allows freelancers to offer specific tasks at set prices, with categories ranging from graphic design to digital marketing. It boasts a user-friendly interface where there is no bidding—clients choose from pre-set packages, and freelancers set their rates and packages. The downside is its limited scope for custom projects and a platform structure that can favor more established freelancers, making it hard for newcomers to get noticed.
3. Freelancer.com: This platform includes job posting and contest options, allowing freelancers to compete by submitting work based on the client’s brief. Its contest feature can yield creative results and allows clients to choose from multiple submissions. However, it can be overwhelming due to the number of bids per project, and the quality of work can vary significantly.
4. Toptal: Specializing in connecting clients with the top 3% of freelancers in fields such as software development, finance, and product management, Toptal's rigorous screening process ensures that only highly skilled professionals are listed, offering high-quality results for clients. The main drawback is the higher costs associated with the high caliber of talent, which may not be accessible for all businesses.

Analysis of Common Features and Gaps:

Most platforms facilitate direct communication between clients and freelancers, provide some form of payment security, and offer varied levels of service categorization. However, there are gaps in personalized service matching, support for complex project management, lower fee structures, and enhanced security measures that are often cited as needing improvement.

Conclusion:

By analyzing these platforms, one can see where there is room for improvement and innovation. Understanding the landscape of existing freelance service apps is crucial for new entrants to carve out their niche. By leveraging the insights gained from current market leaders and improving upon their shortcomings, new platforms can position themselves to offer unique and valuable services that better meet the needs of freelancers and clients alike.

# **3. Background**

## *3.1 Project Origin and Inspiration*

The idea for TaskTenders arose from observing the growing demand for efficient and reliable freelance services in the gig economy. With the increasing number of individuals and small businesses seeking flexible, skilled workers for various tasks such as home repairs, cleaning, and other professional services, it became evident that the traditional methods of finding and hiring freelancers are often inefficient. Users are often left with unreliable service providers, while freelancers struggle to consistently find job opportunities that match their skills and schedules.

Inspiration for the Project came from analyzing existing freelance platforms and identifying key pain points faced by both service seekers and freelancers. Many platforms are plagued with issues such as poor communication, lack of transparency in the bidding process, and unreliable payment systems. Additionally, users often have to navigate multiple platforms or rely on informal networks to find freelancers for their specific needs, which further complicates the process.

TaskTenders was conceptualized to provide a solution to these issues by creating a platform where users can easily post jobs, freelancers can bid on tasks, and the selection process is transparent and secure. The inspiration also stems from the desire to empower both users and freelancers by giving them a reliable, streamlined platform to connect, negotiate deals, and complete tasks efficiently.

By addressing these issues, TaskTenders seeks to bridge the gap between users who need quality services and freelancers looking for consistent work opportunities. The project's vision is to make freelance service engagement easier, more transparent, and accessible to a broader audience, contributing to the evolving gig economy in a meaningful way.

## *3.2 Key Features of Successful Freelancer Platforms*

Successful freelancer platforms have revolutionized the gig economy by offering reliable, user-friendly solutions that efficiently connect service providers with seekers. These platforms have become essential for freelancers and clients alike, thanks to several key features that ensure usability, security, and trust.

A user-friendly interface is crucial for the success of any freelancer platform. It should allow both users and freelancers to post jobs, bid on tasks, and communicate effortlessly. An intuitive and easy-to-navigate interface boosts user engagement and creates a positive experience, encouraging more frequent use.

An efficient matching system is another fundamental component. Such a system uses advanced algorithms to connect users with freelancers based on criteria like location, skills, and availability. This automation speeds up the process of finding the right freelancer for the job, ensuring that freelancers are matched with tasks that suit their expertise.

Transparency in the bidding and payment processes is also vital. Users should be able to easily compare freelancer offers, considering factors such as pricing, ratings, and work history. Secure payment gateways are equally important, as they ensure seamless financial transactions. Many platforms incorporate escrow systems to safeguard funds until the job is satisfactorily completed, protecting both parties involved.

Comprehensive profiles and rating systems are essential for building trust. These platforms allow freelancers to showcase their skills, experience, and portfolios, while also enabling both parties to rate and review each other post-job completion. This transparency helps future users make informed decisions based on reliable feedback.

In-app messaging and communication facilitate direct and secure discussions about project details, updates, and clarifications. This feature helps prevent misunderstandings and ensures that both parties are fully aligned on the project’s scope and requirements.

Secure authentication and user verification play a critical role in preventing fraud and ensuring the legitimacy of users and freelancers. Implementing strong authentication measures, such as two-factor authentication and verified profiles, enhances overall platform security and trust.

The ability to handle escalations and disputes effectively is another crucial feature. Successful platforms provide mechanisms for resolving conflicts, offering a way to fairly and efficiently handle any issues that arise during the course of a project.

Mobile accessibility is increasingly important in today’s digital age. A well-developed mobile application allows users and freelancers to manage jobs and bids on the go, thereby increasing the platform’s accessibility and user engagement.

Task categorization and robust filtering options make it easier for users to find specific services and for freelancers to locate jobs that match their skills. This functionality enhances the platform’s efficiency and user satisfaction.

Lastly, supporting multiple payment methods offers flexibility and convenience, making it easier for users to make payments and for freelancers to receive their earnings.

By integrating these key features, successful freelancer platforms can cultivate a strong user base, deliver high-quality services, and build a robust sense of trust and reliability between users and freelancers, ensuring their ongoing relevance and competitiveness in the freelance marketplace.

## *3.3 The combination between Fiverr and Freelancer.com*

The upcoming project merges the strengths of two distinct platforms, Fiverr and Freelancer.com, to create a hybrid solution that caters to a broad range of freelancing needs. This combination aims to harness the unique capabilities of each platform and provide a comprehensive service offering for both freelancers and clients.

Drawing Inspiration from Fiverr: Like Fiverr, our new platform will focus on the ease of buying and selling freelance services. It will adopt Fiverr’s innovative approach to structured gig offerings, allowing freelancers to create predefined service packages. This feature simplifies the process for clients as they can easily browse, compare, and purchase services that best fit their needs without the need for extensive negotiations. It particularly benefits those looking for quick turnaround times and standardized service scopes.

Incorporating Elements from Freelancer.com: From Freelancer.com, we will integrate the robust bidding system that allows freelancers to pitch directly to potential clients who post specific project descriptions. This feature enables a dynamic marketplace where clients can receive competitive bids from multiple freelancers and make informed decisions based on freelancer proposals, ratings, and portfolios. It emphasizes a tailored approach to project fulfillment and leverages the competitive aspect of bidding to ensure clients get the best value for their projects.

Combination Benefits: By merging these two models, the new platform will offer comprehensive flexibility in how services are procured and delivered. Clients will have the option to choose between quick, predefined gigs for faster execution and a bidding process for projects that require more customization or involve complex requirements. For freelancers, this hybrid model opens up more opportunities to showcase their skills in a format that best suits their strengths—whether that’s through showcasing standard gigs or responding to custom project requests.

Enhanced User Experience: The platform will prioritize a user-friendly interface, drawing on the best design elements from both Fiverr and Freelancer.com to create an intuitive, accessible, and efficient online environment. This approach not only enhances the user experience but also caters to a broader user base, ranging from tech-savvy clients to those who may be new to the freelancing digital marketplace.

In conclusion, this innovative combination aims to create a versatile and robust platform that meets diverse client needs while providing freelancers with multiple avenues to engage and succeed. It’s designed to be a one-stop-shop for freelancing services, enhancing the way freelance work is done by providing a seamless, integrated experience that leverages the best of both worlds from Fiverr and Freelancer.com.

# **4. Expected Achievements**

## *4.1 Outcomes*

The expected achievements of TaskTenders reflect its mission to revolutionize the freelance marketplace by offering an innovative, secure, and efficient platform for both freelancers and clients. Here are the key outcomes the project aims to achieve:

Enhanced Market Efficiency: TaskTenders aims to improve the speed and accuracy of match-making between freelancers and clients. This will be measured by a reduction in the average time from job posting to job acceptance, with a target improvement of 30% within the first year. This outcome is crucial for demonstrating the platform's efficiency in connecting parties quickly and effectively.

Increased Freelancer Engagement: Another goal is to foster a more active and engaged community of freelancers on the platform. Success in this area will be measured by growth in active freelancer registrations and an increase in the average number of bids per project by 20% within six months. Engaging more freelancers will create a more vibrant and competitive marketplace, benefiting all users.

Higher Client Satisfaction: TaskTenders is committed to providing clients with high-quality services that meet or exceed their expectations. This will be measured by achieving an average client satisfaction rate of over 85%, as indicated by post-project surveys, and a net promoter score (NPS) of 40 or higher. High client satisfaction is vital for building trust and retaining users.

Robust Security Implementation: Ensuring the security and integrity of all transactions and interactions on the platform is a primary concern. This will be measured by maintaining zero breaches or significant security incidents reported, alongside regular successful audits of the platform’s security measures. Strong security practices are essential to protect user data and build confidence in the platform.

Innovation in Freelance Project Management: TaskTenders plans to introduce new tools and features that enhance project management capabilities for freelancers and clients. The success of this objective will be gauged by the adoption rate of new features by at least 50% of the user base and positive feedback on their effectiveness in streamlining project management processes.

Scalability and Growth: The platform aims to scale to handle a larger user base and more simultaneous transactions without performance hiccups. This will be measured by the successful handling of a 100% increase in user activity compared to launch, with no significant decrease in system performance.

Conclusion: The expected outcomes for TaskTenders are ambitious yet achievable and are designed to position the platform as a leader in the freelancing market. Achieving these outcomes will not only validate the effectiveness of the platform’s features and design but also significantly enhance the overall user experience. Continuous monitoring and adaptation of strategies will be essential to meet these benchmarks and to accommodate the evolving needs of the freelance community.

## *4.2 Unique Features*

TaskTenders is poised to make a significant impact in the competitive freelancing market by introducing a suite of unique features that enhance user experience, improve efficiency, and ensure security, it will be a combination of "freelancer.com" and "Fiverr" application and will adopt the simpleness of those. Here are the key unique features that distinguish TaskTenders from other platforms:

Smart Matching Engine: The platform's smart matching engine uses artificial intelligence to find the perfect match between clients and freelancers, taking into account skills, budget, past work relationships, and client feedback. This approach significantly reduces the time needed to screen unsuitable candidates and increases client satisfaction by ensuring freelancers are ideally suited for the tasks they perform. Freelancers receive recommendations for jobs that match their skills and are also surprised by notifications of matching job postings when clients receive offers for freelancers that match the job they are posting. This combination of advanced technologies shortens the process and improves the experience of using the platform for all users.

Real-Time Communication Tools: Integrated messaging tools within the platform facilitate seamless communication between clients and freelancers. This feature supports immediate clarification of job details, negotiation, and ongoing project collaboration, enhancing collaboration and speeding up project completion times while reducing misunderstandings and delays.

Dynamic Pricing Model: TaskTenders incorporates a dynamic pricing model that adjusts in real-time based on market demand, the availability of freelancers, and the urgency of the job. This approach ensures fair pricing for both freelancers and clients, making the platform economically attractive and competitive.

Enhanced Security Protocols: Beyond standard security measures, TaskTenders implements blockchain technology for transaction logging and smart contracts for automating payments upon job completion. This increases trust and transparency in the platform’s transactions, providing an additional layer of fraud protection and payment security.

Project Management Dashboard: The platform offers a comprehensive dashboard that allows users to track project progress, set milestones, and manage payments all in one interface. This streamlines project management, making it easier for both freelancers and clients to keep projects on track and within budget.

Community-Driven Development: TaskTenders includes a feature that allows users to submit and vote on ideas for new features or improvements, directly involving the user community in the platform's development. This ensures the platform evolves in alignment with user needs and preferences, fostering strong community engagement.

## *4.3 Criteria for Success*

The success of TaskTenders, designed to revolutionize the freelance marketplace, hinges on several key criteria:

Efficient Matching Engine: The effectiveness of TaskTenders' matching engine in pairing clients with freelancers based on skills, budget, past working relationships, and client feedback is crucial. Success is measured by the engine's ability to reduce time spent sorting through candidates and increasing satisfaction by ensuring freelancers are well-suited to the tasks they take on.

User-Friendly Interface: TaskTenders must provide a seamless and intuitive user experience for both freelancers and clients. Success is gauged by positive feedback on the interface's accessibility, ease of navigation, and the ability for users to input and update their profiles effortlessly.

Real-Time Communication Tools: The success of the platform also relies on providing secure and efficient communication channels. This includes integrated messaging and video conferencing tools that support immediate clarification of job details, negotiation, and ongoing project collaboration, ensuring transparency and coordination.

Dynamic Pricing Model: TaskTenders incorporates a dynamic pricing model that adjusts in real-time based on market demand and the urgency of the job. Success in this area ensures fair pricing for all parties, making the platform economically attractive and competitive.

Robust Security Measures: With advanced security protocols including blockchain technology for transaction logging and smart contracts, success is measured by the platform's ability to avoid security breaches and maintain trust among its users.

Community Engagement and Feedback: The platform's success extends to how well it engages its community of users. Features that allow users to vote on new features or improvements and provide regular feedback are key metrics. Success is seen in user participation rates and positive changes implemented based on feedback.

By consistently meeting and exceeding these criteria, TaskTenders can be deemed successful in its mission to provide a secure, efficient, and user-friendly freelancing platform that adapts to the needs of its users and sets new standards in the freelance marketplace.

# **5. The Process**

## *5.1 Research – User Needs Analysis*

Understanding user needs is crucial to developing a platform that is deeply aligned with its target audience. To this end, we conducted a comprehensive analysis that aimed to uncover the specific needs and expectations of freelancers and clients engaging with online freelancing platforms.

The research was organized around several key objectives: identifying the core needs and challenges faced by both freelancers and clients, exploring their expectations regarding functionality, ease of use, and platform security, and identifying factors that impact user satisfaction and retention within digital marketplaces.

The methodology used included a detailed competitor analysis to benchmark the landscape of existing freelancing platforms. This included reviewing the top platforms to understand service offerings, user interface design, pricing models, and customer feedback. The goal was to identify market gaps and current opportunities where TaskTenders could effectively differentiate itself and excel.

The key findings from this research highlighted the distinct needs of both freelancers and clients. Freelancers value reliable payment systems to ensure timely compensation, effective communication tools to clearly and quickly interact with clients, and enhanced project management features to better track progress and deadlines. On the other hand, clients are interested in robust screening processes to ensure the quality and reliability of freelancers, flexible and transparent pricing structures, and comprehensive tools for tracking and managing projects.

In conclusion, the research provided a clear direction for development, with a focus on improving the user experience to enhance satisfaction and retention rates among its users. By addressing the identified needs and challenges, superior services can be developed and delivered that stand out in the competitive freelancing market. This user-centric approach is essential for the platform’s long-term success and sustainability.

## *5.2 Methodology and Development Process*

This section outlines the overall approach to managing and executing the development of TaskTenders, focusing on the development methodology and tools used to ensure efficient and responsive platform creation.

Development Methodology: TaskTenders has adopted an Agile development methodology, which is ideally suited to projects requiring flexibility and iterative evaluation to adapt quickly to changes. This approach allows for continuous feedback and modifications throughout the development process, ensuring that the final product closely aligns with user needs and market demands. Within the Agile framework, TaskTenders employs the Scrum methodology, which supports frequent sprints and enables the team to concentrate on delivering specific features or functionalities in short, productive cycles. This method promotes a dynamic work environment where project goals are regularly reassessed, and team roles are clear, fostering both accountability and progress.

Tools and Technologies: To effectively manage this Agile development process, TaskTenders utilizes several key tools and technologies:

* Project Management Tools: Jira is employed to track progress, manage sprints, and log issues. This tool is crucial for maintaining an organized approach to project management, ensuring that all team members are aware of deadlines, dependencies, and priorities.
* Version Control System: Git is chosen for version control, hosted on platforms like GitHub. This setup facilitates collaborative coding, allows for effective code review processes, and helps manage changes in a centralized repository. Using GitHub also enhances transparency among team members and contributes to a more cohesive development effort.

The combination of Agile methodologies, specifically Scrum, with robust project management and version control tools, forms the backbone of TaskTenders' development strategy. This structured yet flexible approach ensures that the platform can evolve in response to feedback and changing requirements, ultimately leading to a more reliable and user-friendly product.

## *5.3 Development Phases*

The development of TaskTenders is methodically divided into distinct phases, each tailored to address specific facets of the platform's creation and refinement:

**Phase 1:** Planning and Design In this initial stage, the focus is on laying the foundational work for the platform. The activities involved include requirement gathering to understand what features and functionalities are essential, system architecture design to establish the technical framework, and wireframing along with prototyping to visualize the user interface and experience. The deliverables from this phase are critical and include a comprehensive project plan, detailed design documents, and initial prototypes that provide a first look at the intended end product. These deliverables ensure that there is a clear roadmap and visual guide before development begins.

**Phase 2:** Development This phase is where the theoretical design transforms into tangible software. The activities predominantly involve coding the core functionalities of both the frontend and backend. These functionalities are developed based on the approved designs from Phase 1 and include essential features such as user registration, job posting, bidding systems, and messaging capabilities. The deliverables from this phase are the working software components that enable the basic operations of TaskTenders, setting the stage for more detailed functionality and optimization in subsequent phases.

**Phase 3:** Testing and Iteration Following development, this phase ensures the platform operates smoothly and meets the original specifications. It includes comprehensive testing such as unit tests to check individual components, integration tests to ensure that the components work together correctly, and user acceptance testing (UAT) to validate the overall system with end-users. Feedback from this initial user testing is crucial and is integrated back into development to refine and optimize the platform. The deliverables include detailed test reports, the final adjustments made based on user feedback, and the preparation for the platform's deployment.

**Phase 4:** Deployment and Maintenance The final phase involves moving the fully tested and approved platform from a development setting into a production environment where real users can begin utilizing it. The activities include the actual deployment of the software, followed by continuous monitoring to ensure operational stability and implementing regular updates based on user feedback and system performance analysis. The deliverables from this phase include the deployed application, a set schedule for ongoing maintenance, and plans for ongoing support and updates to continually enhance platform functionality and user satisfaction.

By structuring the development into these phases, TaskTenders ensures a thorough, systematic approach that enhances project management efficiency and product quality, setting the stage for a successful platform launch and reliable operational performance.

# **6. Product**

## *6.1 Our Solution*

TaskTenders revolutionizes the freelancing market by providing a robust, intuitive, and secure platform that connects clients and freelancers across various service categories. Our solution is crafted to tackle common challenges in the freelancing ecosystem such as job matching precision, transparent bidding processes, and secure payment transactions. Here’s how TaskTenders stands out:

**Comprehensive Job Matching**

TaskTenders uses advanced AI-powered algorithms to match freelancers with advertised jobs, ensuring that clients receive offers from the most qualified freelancers. These matches are based on several criteria including skill set, past job performance, and client reviews. This targeted matching system not only enhances the relevance of job matches, but also increases the chances of job satisfaction for both parties.

**Transparent Bidding Mechanism**

Our platform introduces a transparent and fair bidding system where freelancers can submit bids for job postings. Clients can view detailed profiles of bidding freelancers, including their ratings, previous work, and reviews. This open system helps clients make informed decisions while ensuring freelancers are fairly compensated.

**Secure Payment Gateway**

**TaskTenders** integrates a secure payment system that holds funds in escrow until job completion, safeguarding both clients and freelancers. Payments are only released when both parties confirm that the job has been completed to their satisfaction. This mitigates the risk of non-payment and disputes, creating trust and reliability within the platform.

**Real-Time Communication Tools**

The platform features built-in messaging and notification systems that allow seamless communication between clients and freelancers. These tools facilitate clear and continuous dialogue from the initial job posting to final delivery, enhancing collaboration and efficiency.

**Mobile and Cross-Platform Accessibility**

Designed using Flutter, TaskTenders offers a responsive and user-friendly interface accessible on both iOS and Android devices. This cross-platform accessibility ensures that users can manage their projects and respond to job updates from anywhere, at any time.

**Robust Admin Dashboard**

For platform administrators, TaskTenders provides a powerful dashboard that offers insights into user activity, financial transactions, and system health. Admins can manage user accounts, resolve disputes, and generate detailed reports, all from a single interface.

**Benefits of Our Solution**

* **Increased Efficiency**: Automated job matching and real-time updates streamline the process from job posting to completion.
* **Enhanced Security**: Advanced security protocols and escrow payment systems protect user information and funds.
* **Greater Market Reach**: Easy access through mobile apps expands the user base, providing more opportunities for freelancers and a broader selection of talent for clients.
* **Improved User Experience**: Intuitive design and comprehensive support services ensure user satisfaction and platform loyalty.

TaskTenders is not just a freelancing platform; it is a complete ecosystem designed to optimize the interaction between those who need services and those who provide them.

## *6.2 Requirements*

### *6.2.1 Functional Requirements*

1. The application shall allow users to create, edit, and delete their accounts.
2. The application shall offer a secure process for users to recover forgotten passwords.
3. The application shall enable users to update their profile information including skills, contact details, and preferences.
4. The application shall allow clients to post new jobs with specific details such as job description, required skills, budget, and deadlines.
5. The application shall allow clients to edit, close, or delete their job postings.
6. The system shall provide a search and filter mechanism for finding specific jobs.
7. The application shall enable freelancers to browse job listings and submit bids.
8. The application shall allow freelancers to withdraw or modify their bids before acceptance.
9. The application shall enable clients to view all bids, compare them, and accept the most suitable one.
10. The application shall integrate with secure payment gateways to handle transactions.
11. The application shall implement an escrow system to hold payments until job completion.
12. The application shall provide detailed transaction histories for users.
13. The application shall support real-time messaging between clients and freelancers.
14. The application shall send automated notifications for job updates, bid statuses, and payment alerts.
15. The application shall include an admin dashboard for managing user accounts, job posts, and dispute resolution.
16. The application shall enable admins to generate reports on user activity, financial data, and platform usage.
17. The application shall provide tools for monitoring and managing platform security.
18. The application shall allow users to browse freelancers job offers

### *6.2.2 Non-Functional Requirements*

1. The system shall be designed to handle a growing number of users and data as the platform expands.
2. The application shall implement robust security measures to protect user data and privacy.
3. The system shall be reliable, minimizing downtime and ensuring consistent availability.
4. The application shall be compatible with various devices (e.g., smartphones, tablets) and operating systems to maximize accessibility.
5. The application shall include accessibility features, such as adjustable font sizes and compatibility with screen readers, to accommodate users with different needs.
6. The system shall have regular data backup procedures and a reliable recovery mechanism to prevent data loss.
7. The application shall ensure fast and responsive user interactions and data processing to enhance user experience.
8. The system's design shall support easy updates and scalability without significant downtime or disruptions.

## *6.3 System Functionality*

### *6.3.1 Client Functionality*

Clients on TaskTenders, who primarily use the platform to post jobs and seek freelancing services, have access to a range of functionalities designed to streamline their experience. These include:

Account Management: Clients can register, log in, and manage their account settings with ease. They have the option to customize their profile information, which encompasses contact details, payment methods, and personal preferences, ensuring a tailored platform experience.

Job Management: Clients can post new jobs, providing detailed descriptions, specifying required skills, setting budgets, and defining deadlines. They also have the ability to edit, close, or delete their job postings as necessary. Moreover, the platform allows clients to view and manage applications and bids from freelancers, giving them control over whom they choose to hire based on the proposals received.

Package Jobs Browsing: A feature allows clients to browse predefined service packages posted by freelancers. This feature is designed for clients looking for standardized service offerings, which simplifies the hiring process by providing clear expectations and fixed prices, reducing the need for negotiation and expediting the engagement process.

Communication: Effective communication is facilitated through in-app messaging, allowing clients to discuss job specifics directly with freelancers. Additionally, clients receive notifications about job status updates, bids, and messages, which helps keep them informed and responsive throughout the freelance engagement process.

Payment and Invoicing: Secure payments can be made through an integrated payment system, which not only simplifies transactions but also ensures the security of financial data. Clients can view their transaction histories and receive invoices for completed jobs, making financial management straightforward and transparent.

Feedback and Ratings: After the completion of a job, clients are encouraged to rate and review freelancers. This feedback is crucial as it helps other clients make informed decisions and also influences the reputation of freelancers on the platform. Clients also receive recommendations based on their past hiring and feedback, which can aid in making future hiring decisions more aligned with their satisfaction.

These functionalities collectively enhance the user experience on TaskTenders, making it a comprehensive platform for clients to manage their freelancing needs efficiently and securely.

### *6.3.2 Freelancer Functionality*

Freelancers on TaskTenders, as the primary service providers, have access to a comprehensive suite of functionalities that are specifically designed to assist them in efficiently finding and completing jobs:

Account Management: Freelancers start their journey on TaskTenders by registering and logging into their accounts, where they can tailor their profiles. This customization includes adding their skills, qualifications, and work portfolios, which aids in showcasing their capabilities and attracting potential clients.

Service Posting with Packages: In addition to responding to job postings, freelancers can post their services directly, offering several predefined packages. This feature allows freelancers to specify different service levels and price points, which clients can browse and purchase directly. This capability not only diversifies the freelancers’ opportunities to generate income but also simplifies the process for clients to engage services that meet their immediate needs.

Job Search and Bidding: The platform enables freelancers to browse and search through job listings to find opportunities that match their skills. They can submit bids for these postings, detailing their proposed terms and rates. Freelancers also have the flexibility to withdraw or modify their bids before they are accepted, which allows them to adapt to changing circumstances or information.

Project Management: Once engaged in a project, freelancers can track ongoing projects and deadlines to ensure timely delivery of work. The platform facilitates the submission of work directly through it for client approval, streamlining the process of review and acceptance.

Communication: Effective communication is critical, and the platform supports this through in-app messaging, enabling freelancers to clarify details or negotiate terms directly with clients. Real-time notifications keep freelancers updated on job statuses and any communications from clients, ensuring they remain informed and responsive.

Financial Management: Freelancers manage their payments through the platform's secure escrow system, which safeguards their earnings until project milestones are met. They also have access to their financial records and transaction histories, which assists in keeping track of their earnings and financial health.

Ratings and Feedback: After the completion of a job, freelancers can provide feedback on their experience with the client, contributing to a cooperative working environment. They also build their reputation on the platform through client ratings and reviews, which can influence their success in securing future projects.

These functionalities collectively empower freelancers on TaskTenders to effectively manage their freelance careers, from securing projects to managing their financials, all within a supportive and structured environment. This setup not only enhances the freelancers' ability to market their services but also increases the platform's utility for clients looking for a diverse range of freelance solutions.

### *6.3.3 Administrator Functionality*

System administrators play a crucial role in managing and maintaining TaskTenders, ensuring that the platform operates smoothly and adheres to various regulations:

User Management: System administrators are responsible for overseeing user accounts. This includes monitoring account activity, handling user verifications, and making decisions on approving or denying registrations. Their vigilance helps maintain the integrity and security of the platform, ensuring that all users meet the necessary criteria to participate.

Content Moderation: Another vital responsibility is content moderation. Administrators review and manage job postings and bids to ensure they adhere to platform policies. They also moderate user interactions to maintain compliance with community guidelines, creating a safe and professional environment for all users.

System Monitoring and Reports: System administrators access detailed analytics and reports on user activity, financial transactions, and overall system performance. They monitor the health of the system and are in charge of managing backups, updates, and security protocols. This comprehensive monitoring helps prevent system failures and ensures that the platform remains robust and secure.

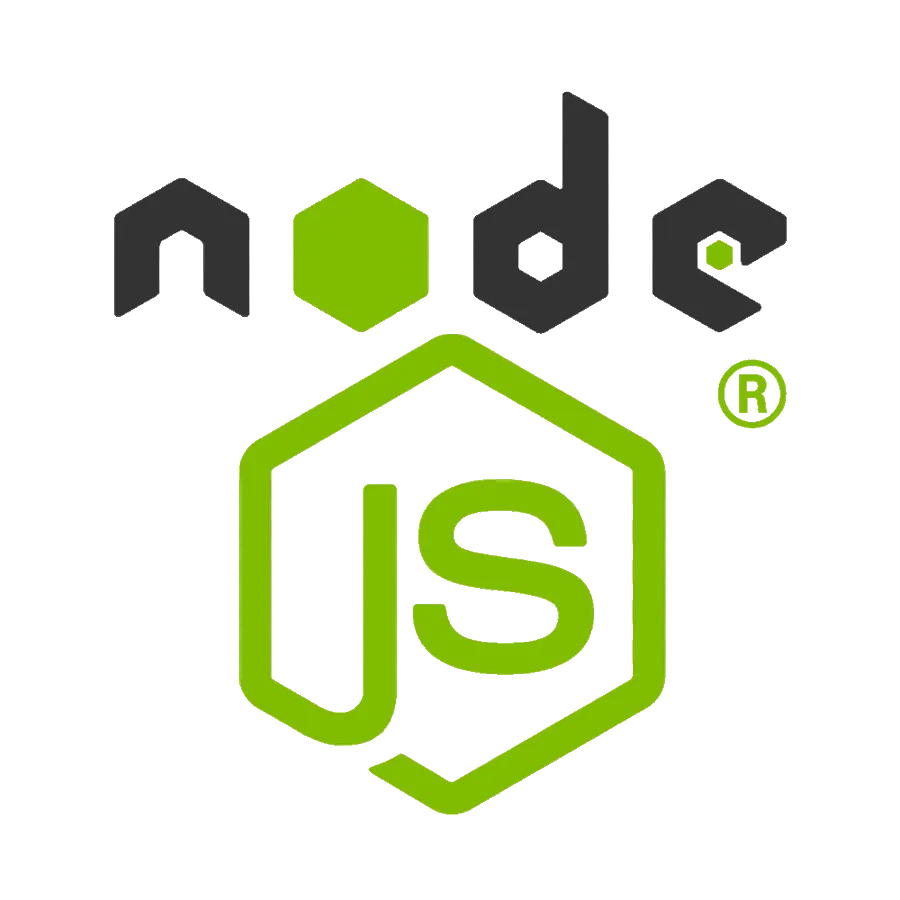
Dispute Resolution: In the event of disputes between clients and freelancers, system administrators act as mediators. They review evidence from communication logs and job documentation to make informed decisions that aim to resolve conflicts fairly and efficiently. This function is critical in maintaining trust and fairness within the platform.

Policy and Compliance Management: System administrators also update and enforce platform policies, ensuring that all operations comply with legal standards and data protection regulations. This responsibility includes adapting to new laws and regulations that affect the platform, thereby safeguarding the platform’s compliance and operational legality.

Overall, the role of system administrators is integral to the functionality and reliability of TaskTenders, as they ensure that the platform remains a secure, efficient, and compliant place for both freelancers and clients to conduct their business.

## *6.4 Technology Choices*

### *6.4.1* *Back-end: Node.js*

Node.js is an open-source, cross-platform JavaScript runtime environment that enables developers to execute JavaScript on the server side. Built on Chrome's V8 JavaScript engine, Node.js is designed for efficiency and scalability, making it an ideal choice for building network applications. Its event-driven, non-blocking I/O model allows Node.js to handle a large number of concurrent connections with minimal overhead, making it particularly well-suited for real-time, data-intensive applications like TaskTenders.

In TaskTenders, Node.js powers the server-side logic, handling tasks such as job postings, bids, and real-time communication. Additionally, TypeScript is used in conjunction with Node.js to add static typing, improving code reliability, scalability, and maintainability.

**Node.js advantages:**

1. High Scalability: Node.js excels at managing numerous concurrent connections, making it ideal for real-time applications like messaging, live updates, and bidding in TaskTenders.
2. Fast Performance: Powered by the V8 engine, Node.js executes JavaScript code at high speeds, resulting in better performance for web applications.
3. Non-blocking I/O: The event-driven, non-blocking architecture enables Node.js to handle multiple requests without waiting for operations to complete, increasing efficiency.
4. Single Language (JavaScript/TypeScript): Node.js allows developers to use JavaScript (or TypeScript) for both frontend and backend development, promoting code consistency and easier collaboration between different layers of the stack.
5. Large Ecosystem: The Node.js ecosystem, supported by npm (Node Package Manager), provides access to thousands of open-source libraries and modules, speeding up development and adding flexibility.
6. Real-time Capabilities: Node.js is highly efficient for real-time applications like chat systems, push notifications, and live bidding, all essential features of TaskTenders.
7. RESTful APIs: Node.js is a great choice for building lightweight, fast RESTful APIs that enable smooth communication between the frontend (mobile app) and backend.

**Node.js disadvantages:**

1. CPU-Intensive Tasks: Node.js’s single-threaded nature makes it less efficient for handling heavy computational tasks, which can slow down the event loop.
2. Callback Hell: The asynchronous nature of Node.js can lead to complicated nested callbacks, though modern JavaScript features like async/await mitigate this problem.
3. Maturity of Modules: Although Node.js has a vast ecosystem, some third-party libraries and modules may be less stable or mature than those in more established languages.
4. Single Thread Limitations: While Node.js is excellent for I/O operations, it may face performance issues for CPU-heavy tasks, as it operates on a single thread.

**Express.js Framework**

Node.js is paired with Express.js, a minimal and flexible web application framework that simplifies the development of APIs and web applications. Express.js provides robust routing, middleware, and request-handling capabilities, making it easy to build a scalable and maintainable backend for TaskTenders.

**Database Integration**

Node.js works seamlessly with various databases, both relational and NoSQL. For TaskTenders, we use MongoDB, a NoSQL database, which offers the flexibility required to handle dynamic, unstructured data such as job postings, user profiles, and bids. MongoDB’s document-based structure makes it ideal for handling large volumes of complex data in a scalable manner.

**Real-time Communication with Socket.io**

Node.js integrates with Socket.io to provide real-time, bi-directional communication between the client and the server. This is crucial for features like instant messaging, notifications, and live bidding within TaskTenders, allowing users and freelancers to interact in real-time without delays.

**Node.js for RESTful APIs**

Node.js is well-suited for building lightweight, fast, and scalable RESTful APIs. In TaskTenders, these APIs enable efficient communication between the mobile app and the server, handling tasks like user authentication, job postings, bidding, and payment processing. The non-blocking architecture ensures that requests and responses are handled quickly and efficiently, improving the overall user experience.

### *6.4.2* *Back-end: TypeScript*

TypeScript is a statically-typed superset of JavaScript that enhances the development process by adding static typing, interfaces, and advanced programming features to JavaScript. In the TaskTenders project, TypeScript is used alongside Node.js to ensure more reliable and maintainable server-side development. TypeScript’s type safety helps developers catch errors early in the development phase, leading to fewer bugs in production and making the codebase easier to scale as the project grows.A blue square with white letters

Description automatically generated

TypeScript integrates seamlessly with JavaScript and the Node.js runtime, allowing developers to build and maintain a robust backend while leveraging JavaScript’s flexibility. The TypeScript compiler converts TypeScript code into JavaScript, ensuring compatibility with the Node.js ecosystem while adding the benefits of strong typing and enhanced tooling.

**TypeScript advantages:**

1. **Static Typing:** TypeScript introduces static types to JavaScript, allowing developers to define the types of variables, function parameters, and return values. This helps catch potential bugs during development rather than at runtime, improving the overall quality of the code.
2. **Early Error Detection:** With TypeScript, errors related to type mismatches or undefined behaviors are caught at compile time. This reduces the likelihood of runtime errors, resulting in more stable applications.
3. **Improved Code Readability and Maintainability:** By explicitly stating types, TypeScript makes the code more readable and easier to understand. This is particularly beneficial when working in teams or on large codebases, as it provides clear expectations about the data and behavior of functions.
4. **Enhanced Tooling:** TypeScript provides better support for IDE features like autocompletion, navigation, and refactoring. These features, especially in Visual Studio Code, improve developer productivity by providing real-time feedback and intelligent suggestions.
5. **Seamless Integration with JavaScript Libraries:** TypeScript is fully compatible with existing JavaScript libraries and frameworks. In the TaskTenders project, TypeScript integrates easily with Node.js and Express, ensuring that the development workflow remains smooth while adding type safety and error prevention.
6. **Scalability:** As TaskTenders grows, TypeScript’s type system ensures that new features can be added without introducing unforeseen bugs. The strong typing and modular nature of TypeScript make it easier to refactor and scale the application as its functionality expands.
7. **Better Refactoring:** TypeScript’s static typing allows for safer and more efficient refactoring of code. Since types are explicitly defined, developers can modify code with confidence, knowing that type-checking will catch any potential issues introduced during refactoring.

**TypeScript disadvantages:**

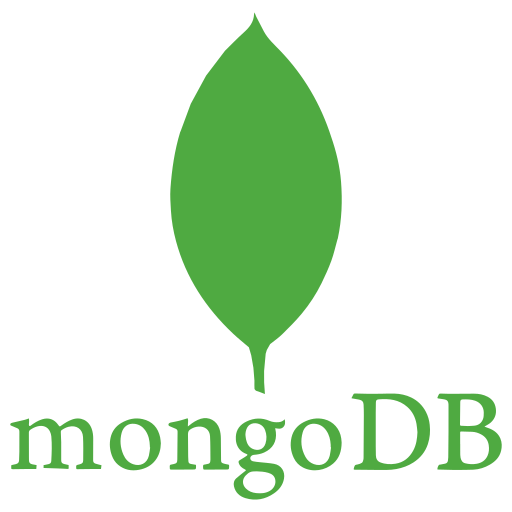
1. **Learning Curve:** Developers familiar with JavaScript may need time to adjust to TypeScript’s strict type system and new features. However, the long-term benefits of using TypeScript often outweigh the initial learning curve.
2. **Longer Development Time:** Writing type definitions and handling TypeScript’s strict rules may slow down development in the short term. However, this can result in fewer bugs and less debugging time later in the process.
3. **Additional Compilation Step:** Unlike JavaScript, TypeScript needs to be compiled into JavaScript before it can be executed by Node.js. This introduces an additional step in the development workflow, though modern tools and build processes can automate this.

**Why TypeScript for TaskTenders?**

TypeScript was chosen for TaskTenders to ensure that the backend code is more maintainable and scalable. By catching errors during the development phase and enforcing a consistent structure across the codebase, TypeScript helps improve code quality and developer productivity. The integration with Node.js and Express ensures that the application benefits from modern JavaScript features, with the added security and reliability of a statically-typed language.

In the TaskTenders backend, TypeScript enhances the development process by providing better code structure, clearer data flow, and a more reliable foundation for building RESTful APIs and handling real-time communication.

### *6.4.3 Database: MongoDB*

MongoDB is a popular, open-source NoSQL database known for its scalability, flexibility, and ability to handle large volumes of data in a non-relational format. Unlike traditional relational databases that use tables and rows, MongoDB stores data in flexible, JSON-like documents, making it ideal for applications that need to handle dynamic and unstructured data.

For **TaskTenders**, MongoDB provides the flexibility needed to manage various types of user-generated data, including job postings, bids, user profiles, and messages between users and freelancers. Its document-based structure allows us to store this information in a way that adapts easily to changes in the app’s data requirements.

**MongoDB advantages:**

1. **Schema Flexibility:** MongoDB’s flexible schema allows for easy updates and changes to data structures without the need for complex migrations, making it ideal for rapidly evolving applications like TaskTenders.
2. **Scalability:** MongoDB is designed for horizontal scalability, allowing it to handle large amounts of data across multiple servers. This ensures the app can grow as the user base expands.
3. **High Performance:** MongoDB’s efficient querying and indexing system enables fast read and write operations, ensuring that TaskTenders delivers quick responses to user actions such as posting jobs, bidding, and messaging.
4. **Document-Based Storage:** MongoDB stores data in JSON-like documents (BSON), which are more intuitive and flexible compared to rigid relational tables. This allows TaskTenders to manage complex data relationships efficiently.
5. **NoSQL Model:** Being a NoSQL database, MongoDB doesn’t require fixed schemas or complex joins, simplifying the management of large datasets and making it easy to store diverse types of information.
6. **Cloud Integration:** MongoDB integrates seamlessly with cloud platforms like AWS, GCP, and MongoDB Atlas, ensuring easy scaling and management in the cloud.
7. **Sharding for Scalability:** MongoDB’s sharding feature helps distribute data across multiple machines, enhancing both storage capacity and performance for large-scale applications.

**MongoDB disadvantages:**

1. **Memory Usage:** MongoDB can require significant memory, as it keeps frequently accessed data in RAM for fast performance.
2. **Limited Support for Complex Transactions:** Although MongoDB introduced multi-document ACID transactions, it still may not be as robust as relational databases when it comes to handling complex transactions.
3. **Less Mature Tooling:** Compared to mature relational databases like MySQL or PostgreSQL, some MongoDB tools, especially for analytics and reporting, are less developed.
4. **Consistency:** MongoDB offers eventual consistency in distributed systems, meaning data may not be immediately updated across all nodes, which could be a challenge in specific use cases.

**Why MongoDB for TaskTenders:**

* **Flexible Data Storage:** The dynamic nature of the data in TaskTenders, such as job postings, user profiles, freelancer bids, and messaging, fits well with MongoDB’s document-oriented approach. This allows the application to evolve without requiring major changes in the database structure.
* **Scalability:** As TaskTenders grows, MongoDB can scale horizontally to handle increased data loads and user traffic, ensuring smooth performance even with a large user base.
* **Fast Prototyping:** MongoDB’s flexible schema design allows rapid prototyping and deployment, making it perfect for the iterative development process required for TaskTenders.

**Integration with Node.js:** MongoDB integrates seamlessly with Node.js through the **Mongoose** library, which provides an elegant, object-oriented interface for interacting with the database. Mongoose simplifies database operations like querying, validation, and data modeling, making it easier to work with MongoDB’s flexible documents in the context of Node.js.

**Real-time Updates:** Using MongoDB’s ability to efficiently handle frequent read and write operations, **TaskTenders** can provide real-time updates for job postings, bid submissions, and user interactions, ensuring a responsive user experience.

### *6.4.4 Front-end: Flutter*

Flutter is an open-source UI software development kit created by Google, enabling developers to build natively compiled applications for mobile (iOS and Android), web, and desktop from a single codebase. Flutter uses the **Dart programming language**, which is designed for fast development and high-performance applications. Dart is optimized for building user interfaces and compiles to native machine code, ensuring efficient performance across platforms.

Flutter provides a rich set of pre-designed widgets that allow for fast and flexible UI development. It uses a reactive framework, like React, but with its own customizable widgets to render the UI, ensuring consistency and fluidity on any platform.

**Flutter advantages:**

1. **Cross-Platform Development:** Write once, deploy on iOS, Android, web, and desktop.
2. **High Performance:** Flutter compiles to native machine code, providing near-native performance.
3. **Rich Widgets Library:** A vast library of pre-built widgets ensures a consistent look across platforms.
4. **Fast Development:** Hot reload allows real-time updates without restarting the app, enhancing productivity.
5. **Customizable UI:** Flutter’s flexibility allows for complex UI designs with smooth transitions and animations.
6. **Single Codebase:** A single codebase for all platforms reduces development time and effort.
7. **Growing Community and Ecosystem:** Flutter’s popularity has fostered a large community with extensive resources and support.

**Flutter disadvantages:**

1. **Large App Size:** Flutter apps tend to have larger file sizes due to the Flutter engine being bundled with the app.
2. **Limited Third-Party Libraries:** While Flutter has a growing ecosystem, it may lack some niche libraries compared to older frameworks.
3. **Limited Native Functionality:** Some advanced native features may require custom platform-specific code.
4. **Learning Curve:** Developers may need time to get familiar with Dart, Flutter's primary programming language.
5. **Platform-Specific UI Elements:** Additional customization may be required to achieve a native look and feel for platform-specific UI elements.

**Flutter's DevTools** offer a suite of performance and debugging tools that enable developers to inspect the widget tree, analyze layout and rendering, debug code, and track performance bottlenecks, making development faster and more efficient.

### *6.4.5 Development Environment: Visual Studio Code*

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft, and it is one of the most widely used development environments for modern web and mobile application development. It provides a lightweight yet powerful environment with support for numerous programming languages, including JavaScript, Dart (for Flutter), and Node.js, making it an ideal choice for the development of **TaskTenders**.A blue logo with a cross

Description automatically generated

VS Code’s flexibility and extensive ecosystem of extensions make it well-suited for full-stack development, ensuring seamless integration between frontend and backend development tasks.

**Visual Studio Code advantages:**

1. **Lightweight and Fast:** VS Code is lightweight, meaning it can run on almost any system without consuming excessive resources, while still providing robust functionality for large projects like TaskTenders.
2. **Rich Extension Ecosystem:** With a vast marketplace of extensions, developers can enhance VS Code with tools for linting, debugging, Git integration, and language support. Popular extensions like Prettier, ESLint, and Flutter/Dart extensions improve code quality and efficiency.
3. **Built-in Git Integration:** VS Code has built-in support for Git version control, allowing developers to easily manage source code repositories, commit changes, and track project history without leaving the editor.
4. **Debugging Capabilities:** VS Code provides a powerful, built-in debugger for JavaScript and Node.js, and with extensions, it supports Dart for Flutter. This enables developers to debug frontend and backend code within the same environment.
5. **Cross-Platform:** VS Code is available on Windows, macOS, and Linux, making it a flexible choice for developers working in different environments.
6. **Terminal Integration:** The integrated terminal allows developers to run shell commands, npm scripts, and interact with Git directly within VS Code, streamlining the development workflow.
7. **Code Autocompletion:** The IntelliSense feature provides intelligent code suggestions and autocompletion for JavaScript, Dart, and other languages, making coding faster and reducing errors.
8. **Customizable:** VS Code is highly customizable, with themes, keybindings, and configuration options that allow developers to tailor the editor to their workflow.

**Visual Studio Code disadvantages:**

* **Resource Usage:** Although lightweight compared to traditional IDEs, VS Code can become resource-intensive when running multiple extensions and large projects.
* **Learning Curve for Extensions:** While extensions greatly enhance functionality, configuring and managing them can add complexity, especially for new users.
* **Limited Native Features:** Some features available in full Integrated Development Environments (IDEs) might require extensions to be enabled in VS Code, potentially making it less straightforward compared to IDEs designed specifically for a particular language.

**Why Visual Studio Code for TaskTenders:**

* **Frontend and Backend Development:** VS Code’s excellent support for both **Node.js** (backend) and **Flutter/Dart** (frontend) makes it an ideal choice for developing TaskTenders. Developers can switch seamlessly between backend services in Node.js and the frontend interface in Flutter using a single editor.
* **Git Integration:** With the built-in Git integration, the development team can easily collaborate on the project, manage source code changes, and keep track of version history for TaskTenders’ codebase.
* **Real-time Collaboration:** Using extensions like **Live Share**, VS Code allows developers to collaborate in real-time, making pair programming and team collaboration easier during the development process.
* **Customization for TaskTenders:** By configuring custom tasks, snippets, and key bindings, the development team can streamline repetitive workflows and automate common tasks related to building, testing, and deploying the app.

**Extensions for TaskTenders Development:**

* **Flutter & Dart:** For front-end development, the **Flutter** and **Dart** extensions provide support for building and debugging the mobile app.
* **Node.js Extension Pack:** Includes tools like **npm**, **Express Snippets**, and **REST Client** for managing the backend API and services.
* **Prettier and ESLint:** Code formatting and linting tools to ensure code quality and consistency across the team.
* **MongoDB Extension:** Enables developers to interact with the MongoDB database directly from VS Code, making it easier to test queries and visualize data.

## *6.5 Project Structure*

The TaskTenders project is organized to facilitate efficient development, scalability, and maintainability. It follows a modular structure to separate concerns between the frontend, backend, and shared resources, ensuring that each part of the system is independently manageable and easy to collaborate on. This organization also allows for better testing, debugging, and future expansions.

**1. Frontend (Flutter)**

The frontend of TaskTenders is built using Flutter, a framework for building cross-platform mobile applications. The Flutter project structure is organized to separate UI components, business logic, and services, making it easier to maintain and scale.

* **lib/**: Main directory for all Flutter app code.
  + **screens/**: Contains all the screen widgets (e.g., login screen, job posting screen, freelancer bidding screen).
  + **widgets/**: Reusable UI components, such as buttons, cards, or form fields.
  + **models/**: Data models that define the structure of objects such as user profiles, job postings, and bids.
  + **providers/**: Handles the state management and business logic using Provider or Riverpod for managing state across different screens.
  + **services/**: Includes code for interacting with backend APIs, authentication, and other external services.
  + **utils/**: Utility functions such as formatters, validators, or constants used throughout the app.
  + **assets/**: Contains images, fonts, and other static resources used in the app.

**2. Backend (Node.js/Express.js)**

The backend is responsible for handling server-side logic, managing data storage, processing requests, and providing APIs for the frontend. It is built using Node.js with the Express.js framework.

* **src/**: Main directory for backend code.
  + **controllers/**: Contains logic for handling incoming API requests (e.g., user registration, job posting, bidding process).
  + **routes/**: Defines the routes and endpoints for the APIs (e.g., /api/users, /api/jobs).
  + **models/**: Defines data schemas and models using Mongoose to interact with the MongoDB database.
  + **middleware/**: Contains middleware functions for authentication, authorization, and validation.
  + **services/**: Handles services like sending emails, processing payments, and interacting with third-party APIs.
  + **config/**: Contains configuration files (e.g., environment variables, database connection strings).
  + **utils/**: Utility functions, such as for data validation, formatting responses, or logging.
  + **tests/**: Unit and integration tests for the backend logic to ensure code reliability.

**3. Database (MongoDB)**

MongoDB is the NoSQL database used to store and manage dynamic data for TaskTenders. The database is organized with a focus on scalability and flexibility, and is accessed through the backend using Mongoose.

* **Collections**:
  + **Users**: Stores user profiles for freelancers and clients, including credentials, preferences, and roles.
  + **Jobs**: Contains job postings, including descriptions, budget, deadlines, and job status.
  + **Bids**: Manages freelancer bids on job postings, including bid amounts, timelines, and freelancer profiles.
  + **Transactions**: Tracks payments and transaction histories between users and freelancers.
  + **Messages**: Stores conversation data between users and freelancers for in-app communication.

**4. Shared Resources**

This section covers assets and shared utilities that are used across the entire project.

* **docs/**: Project documentation, including requirements, API specs, and architectural diagrams.
* **logs/**: Log files generated by the backend for monitoring application performance and debugging.
* **tests/**: Contains test cases for both frontend and backend code to ensure the reliability of features.

**5. Version Control**

TaskTenders uses Git for version control, with the source code hosted on GitHub. The project structure supports branching strategies such as feature branches, bugfixes, and hotfixes to streamline team collaboration and ensure code stability.

* **master/main branch**: Contains the stable code that is ready for production.
* **development branch**: Used for integrating new features and performing testing before merging into the main branch.
* **feature branches**: Each new feature or functionality is developed in its own branch, allowing developers to work in isolation without affecting the main project.
* **bugfix branches**: For resolving issues and bugs reported in the system.

**6. Testing**

Both the frontend and backend include tests to ensure the reliability and stability of the application.

* **Frontend Tests**: Uses Flutter’s testing framework for unit tests and widget tests, ensuring that UI components and business logic function correctly.
* **Backend Tests**: Uses Mocha/Chai for unit and integration tests on the Node.js backend, testing API endpoints, data processing, and user authentication.

## *6.6 Algorithms*

**1. Matching Algorithm**

The matching algorithm is fundamental to the TaskTenders platform, ensuring that user's job postings are accurately matched with freelancers who have the requisite skills and experience. This algorithm analyzes various factors such as:

* **Skill Relevance**: Matching based on the skills and qualifications listed by freelancers against the requirements of the job posting.
* **Geographical Location**: Preferentially matching freelancers who are in close proximity to the job location, if relevant.
* **Availability**: Aligning freelancer availability with the job timeline.
* **Rating and Performance History**: Prioritizing freelancers with higher ratings and a history of successful job completions.

This algorithm uses a weighted scoring system where each factor is assigned a score based on its importance. The scores are then aggregated to rank freelancers for each job posting.

**2. Bidding Algorithm**

The bidding algorithm facilitates fair and competitive bidding by allowing freelancers to place bids on job postings. It includes features such as:

* **Dynamic Pricing Suggestions**: Based on historical data, the algorithm suggests a competitive bid amount that freelancers might consider.
* **Time Decay Factor**: Bids might be adjusted based on the urgency of the job posting, with more immediate needs potentially driving higher bid values.
* **Bid Ranking**: Sorting bids not only by price but also considering factors like freelancer rating and the number of previously completed jobs.

**3. Rating System**

The rating system algorithm is designed to ensure fairness and accuracy in how freelancers' performance is assessed. It considers:

* **Client Feedback**: Ratings provided by clients after a job is completed.
* **Complexity of Jobs**: Weighing ratings according to the complexity of the tasks completed.
* **Timeliness and Communication**: Factors such as adherence to deadlines and communication quality affect the final rating.

**4. Dynamic Job Recommendation Engine**

For users, a recommendation engine suggests potential jobs based on their previous search behaviors, skills, and ratings. This uses machine learning techniques to learn user preferences over time and improve the relevance of job recommendations.

**5. Fraud Detection Algorithm**

To maintain the integrity of the platform, a fraud detection algorithm continuously analyzes user activities to flag potentially fraudulent or abnormal behaviors such as:

* **Multiple Accounts**: Identifying users who operate multiple freelancer or client accounts to manipulate bidding or ratings.
* **Payment Anomalies**: Detecting unusual patterns in payment transactions that could indicate fraudulent activity.

**Implementation Considerations**

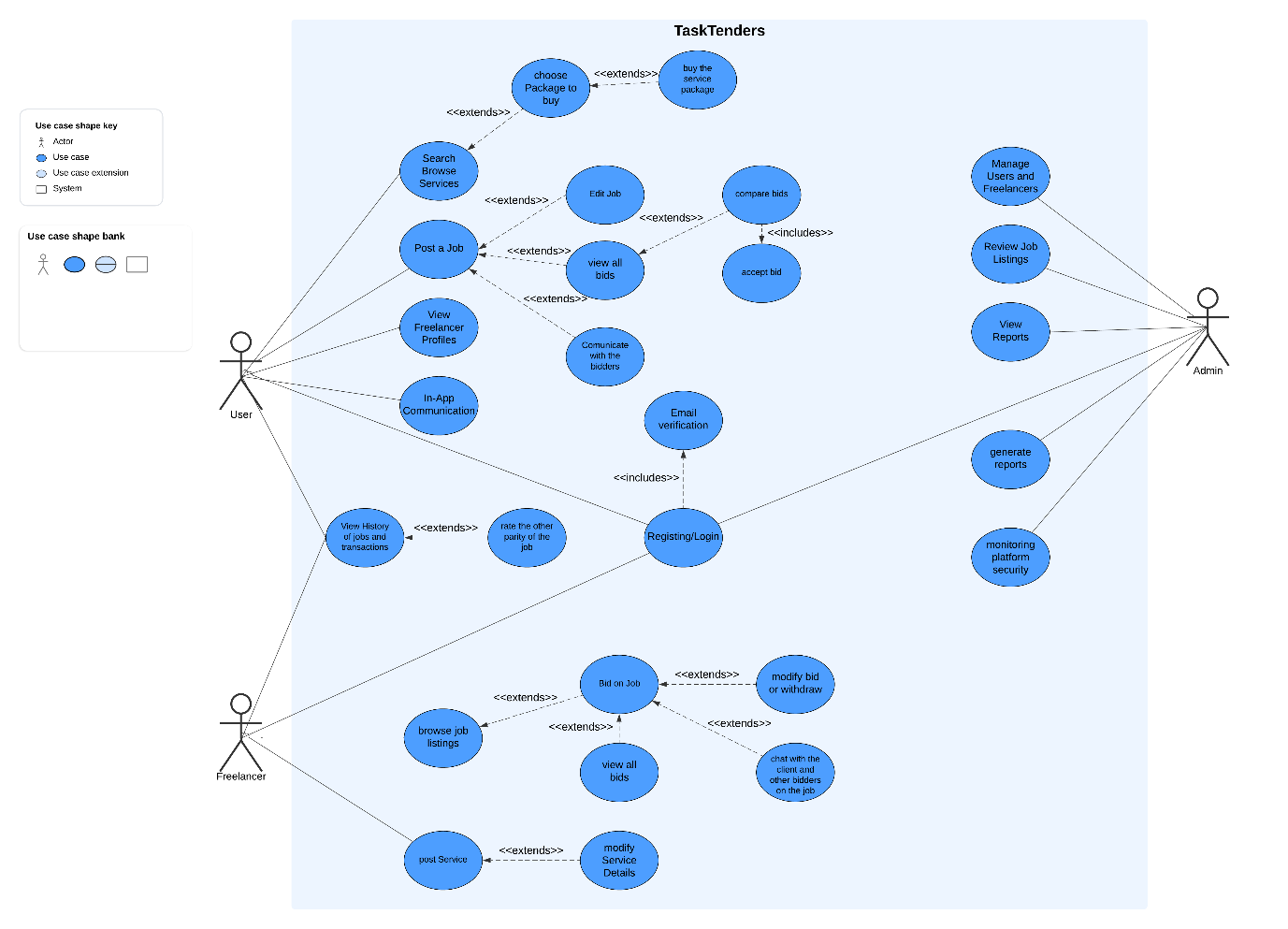
These algorithms are implemented using a combination of traditional programming techniques and modern machine learning methods, depending on the complexity and requirements of each task. The backend, developed with Node.js and TypeScript, integrates these algorithms to ensure that data handling is efficient and secure. The algorithms are regularly updated based on feedback and new data to improve accuracy and performance.

**Conclusion**

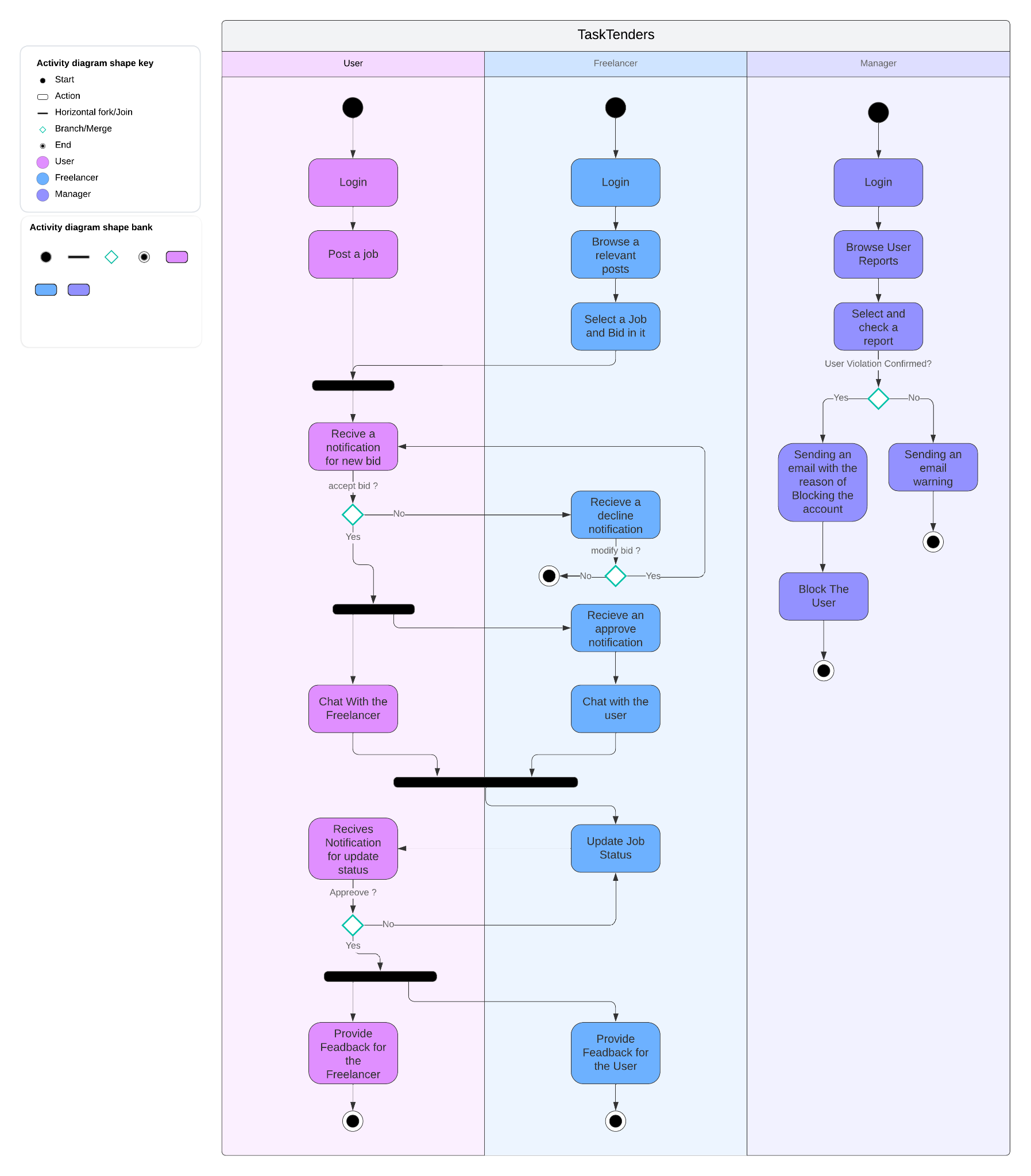
The algorithms section outlines the critical computational processes that support the TaskTenders platform. By elaborating on these key algorithms, stakeholders can understand how the platform operates dynamically to match freelancers with jobs, facilitate competitive bidding, ensure fair ratings, recommend relevant jobs, and prevent fraud.

## *6.7 Diagrams*

### *6.7.1 use case diagram*



### *6.7.2 Activity diagram*



## *6.8 Tests*

This section outlines the test cases designed to validate the functionality, security, and performance of the TaskTenders platform, segmented by different user roles and system administration.

### *6.8.1 Login Tests*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Method | Expected Outcome | Status |
| L1 | Correct credentials login | Automated Test | User is logged in successfully | Pending |
| L2 | Incorrect password | Automated Test | Login fails with an error message | Completed |
| L3 | Password reset process | Manual Testing | User can reset password and log in | Pending |
| L4 | Login session expiration | Automated Test | User session expires as expected | Completed |
| L5 | Login with two-factor authentication | Automated Test | User is prompted for secondary authentication method | Pending |
| L6 | Login via social media accounts | Integration Test | User can log in using external social media accounts | Pending |
| L7 | Account lockout after multiple failed attempts | System Test | Account is temporarily locked after consecutive failed login attempts | Pending |
| L8 | Captcha verification on login | Manual Testing | Captcha is required and works correctly after failed login attempts | Completed |
| L9 | Handling of session tokens during login | Security Test | Session tokens are handled securely during and after login | Pending |
| L10 | Response to SQL injection attempts during login | Security Test | System securely rejects SQL injection in login fields | Completed |

**Purpose**: To ensure that the login mechanism is secure, user-friendly, and robust against unauthorized access attempts.

### *6.8.2 User Tests*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Method | Expected Outcome | Status |
| U1 | Update profile information | Integration Test | Profile is updated with new information | Pending |
| U2 | Delete user account | Manual Testing | User account is deleted correctly | Completed |
| U3 | View job history | Automated Test | User can view past jobs accurately | Pending |
| U4 | Account registration verification | Automated Test | User receives verification email | Pending |
| U5 | Change password functionality | Manual Testing | Password change is processed correctly | Completed |
| U6 | Mobile app login with biometrics | System Test | User can log in using biometric data | Pending |
| U7 | Handling of user session timeouts | Automated Test | User is logged out after inactivity | Completed |
| U8 | Search and filter jobs | Integration Test | Search returns accurate results | Pending |

**Purpose**: These tests ensure that all user-related functionalities are robust, secure, and user-friendly, enhancing the overall experience of the clients and freelancers on the platform.

### *6.8.3 Freelancer Tests*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Method | Expected Outcome | Status |
| F1 | Bid on job posting | Automated Test | Freelancer can submit bids successfully | Pending |
| F2 | Withdraw from bid | Manual Testing | Freelancer can withdraw bid correctly | Completed |
| F3 | Complete job and receive payment | Integration Test | Job completion triggers payment correctly | Pending |
| F4 | Manage portfolio | Manual Testing | Freelancer can update their portfolio | Completed |
| F5 | Receive and respond to client queries | Integration Test | Communication with clients is seamless | Pending |
| F6 | Notification settings adjustment | System Test | Freelancer can customize notifications | Completed |
| F7 | Review and rate clients | Automated Test | Freelancer can submit reviews and ratings | Pending |

**Purpose**: To test and verify that all freelancer-specific functionalities are efficient, reliable, and facilitate their engagement with job postings and clients seamlessly.

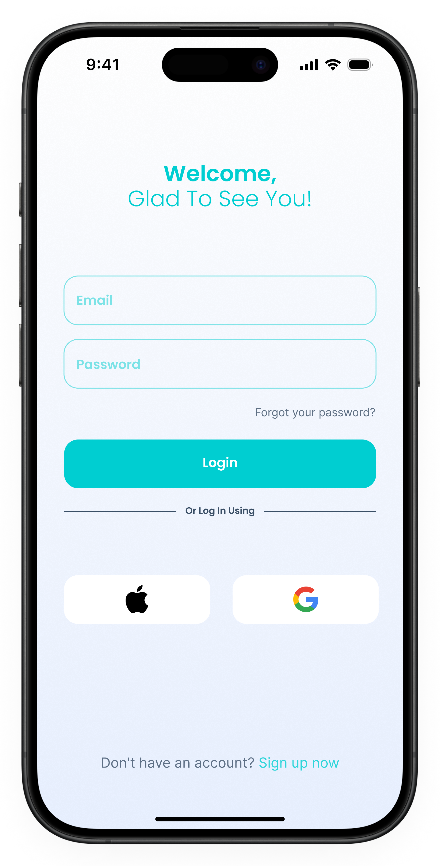
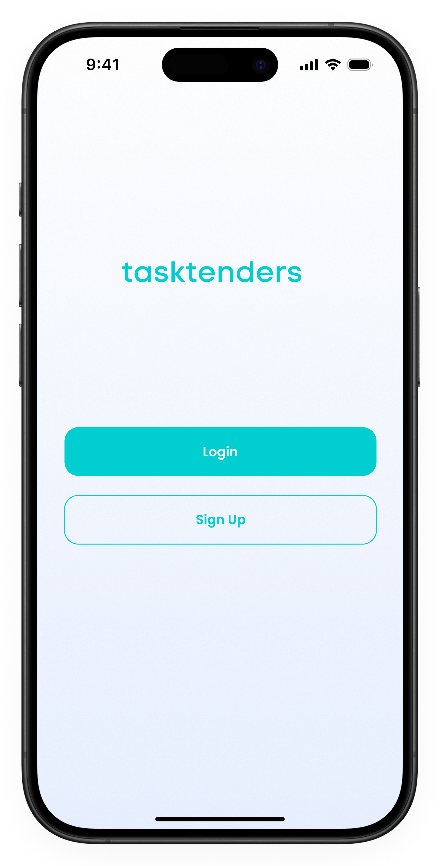
### *6.8.4 Admin Tests*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Method | Expected Outcome | Status |
| A1 | Add new job categories | Manual Testing | Admin can add new categories successfully | Pending |
| A2 | Ban a user | Automated Test | User is banned and cannot access system | Completed |
| A3 | Generate reports | Integration Test | Admin can generate accurate reports | Pending |
| A4 | Manage user complaints | Manual Testing | Complaints are resolved efficiently | Completed |
| A5 | Update platform policies | System Test | Policy updates are deployed correctly | Pending |
| A6 | Monitor user activity | Automated Test | Admin can monitor and review user activity | Completed |
| A7 | Backup and data recovery procedures | Integration Test | Data recovery processes function properly | Pending |

**Purpose**: To ensure that administrative functionalities are robust, secure, and allow for efficient management and oversight of the TaskTenders platform.

# **7. GUI**

## *7.1 Login and Sign-Up Screens*

A screen shot of a phone

Description automatically generated

## *7.2 User Screens*

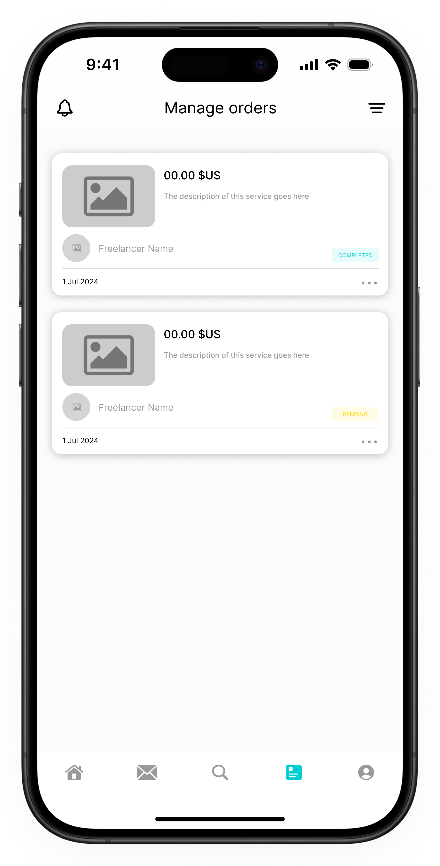
Home Screen – Tabs:

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

 A screenshot of a phone

Description automatically generated

Browse Job, browse Job Bidding and job chat (user can preview freelancers name and profile):

A screen shot of a phone

Description automatically generated A screenshot of a phone

Description automatically generatedA cell phone with a chat screen

Description automatically generated

## *7.3 Freelancer Screens*

Home Screen – Tabs:

**A screenshot of a phone

Description automatically generated**A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

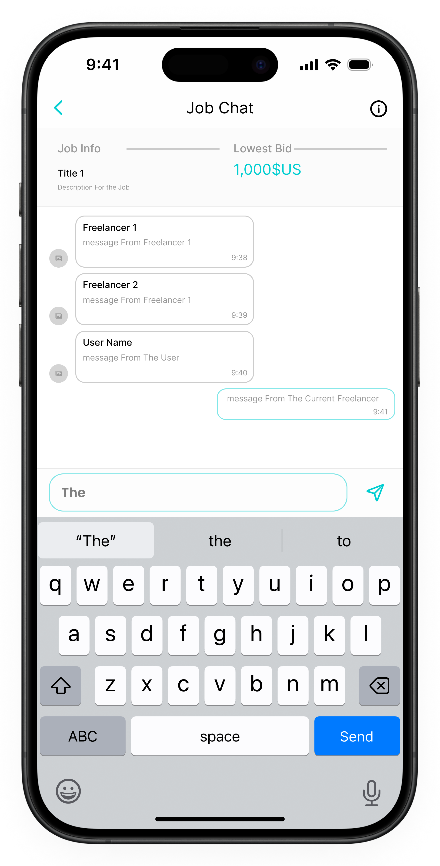
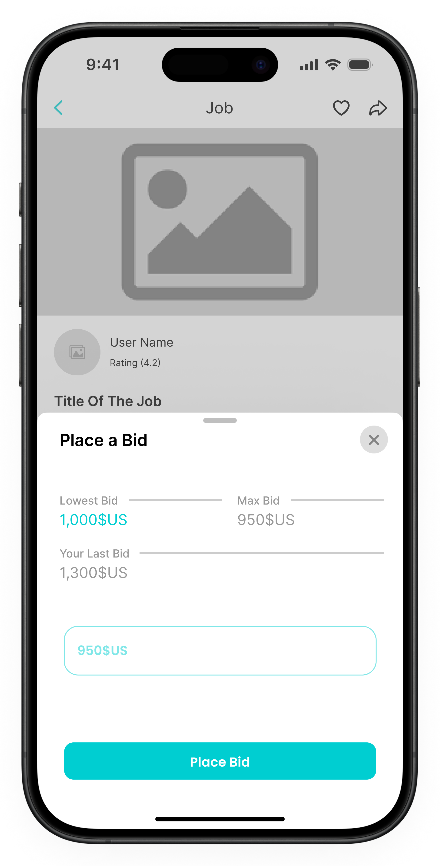
A screenshot of a phone

Description automatically generated A screenshot of a phone

Description automatically generated

Browse Jobs, bid and chat with other freelancers (freelancer can’t see another freelancer’s name):

**A screenshot of a phone

Description automatically generated**

# **8. References**

1. Microsoft. (n.d.). *Visual Studio Code - Code Editing. Redefined.* Retrieved from <https://code.visualstudio.com/>
2. Microsoft. (2021). *Using Git in Visual Studio Code.* Retrieved from <https://code.visualstudio.com/docs/editor/versioncontrol>
3. Node.js — Introduction to Node.js. (n.d.). <https://nodejs.dev/en/learn/>
4. GeeksforGeeks. (2023, December 1). MongoDB Advantages Disadvantages. GeeksforGeeks. <https://www.geeksforgeeks.org/mongodb-advantages-disadvantages/>
5. Google. (n.d.). *Flutter - Beautiful native apps in record time.* Retrieved from <https://flutter.dev/>
6. Google. (n.d.). *Dart Language Overview.* Retrieved from <https://dart.dev/guides>
7. Marco, E. (2019). *Beginning Flutter: A Hands-On Guide to App Development.* Apress.
8. Fiverr. (n.d.). Retrieved from <https://www.fiverr.com/>
9. Freelancer.com. (n.d.). Retrieved from <https://www.freelancer.com/>
10. Upwork. (n.d.). Upwork: The World's Work Marketplace. Retrieved from <https://www.upwork.com/>
11. Toptal. (n.d.). Toptal - Hire Freelance Talent from the Top 3%. Retrieved from <https://www.toptal.com/>
12. Finances Online. (n.d.). Freelance Platforms Analysis: Features, Benefits, Pricing. Retrieved from <https://financesonline.com/freelance-platforms-analysis-features-benefits-pricing/>