**Face Recognition Student Attendance System**

A research proposal submitted in partial fulfillment of the requirements for the course, Research Methodology as partial fulfillment of the degree of Bachelor of Science in Software Engineering.

of

The Ghana Communication Technology University

By

**Kelvin Amankwah Nketia**

(**4231220371**)

7th July 2024

Supervisor: **XXXXXX**

**Table of Contents**

[List of Figures 2](#_Toc1013368576)

[List of Tables 3](#_Toc1176006442)

[RESEARCH PROPOSAL 4](#_Toc2086456364)

[Abstract 4](#_Toc1424336936)

[1.1. Introduction 4](#_Toc1419171991)

[1.2. Background 5](#_Toc1931673839)

[1.3. Problem Statement 5](#_Toc1515506712)

[1.4. Aims and Objectives 6](#_Toc54744936)

[1.4.1. Aim 6](#_Toc1932603111)

[1.4.2. Objectives 6](#_Toc1042012569)

[1.5. Research Questions 7](#_Toc1864300541)

[1.6. Significance 8](#_Toc1192006454)

[1.7. Justification 10](#_Toc1661236665)

[1.8. Limitation and Delimitation 12](#_Toc2106208388)

[1.8.1. Limitations 13](#_Toc866656095)

[1.8.2. Delimitations 13](#_Toc1604773746)

[1.9. Preliminary Literature Review 15](#_Toc1641246048)

[1.10. Research Methodology 16](#_Toc948377903)

[1.10.1. Research Design 17](#_Toc1113813278)

[1.10.2. Research Methods 17](#_Toc479916740)

[1.10.3. Research Population 19](#_Toc1919374765)

[1.10.4. Research Sample 20](#_Toc888426026)

[1.10.5. Research Instrument 21](#_Toc1264791967)

[1.10.6. Functional and Non-Functional Requirements 22](#_Toc1079519772)

[1.10.7. System Design 25](#_Toc1918196345)

[1.10.8. Minimum System Requirements 32](#_Toc279637648)

[1.11. Research Ethics 34](#_Toc835756815)

[References 35](#_Toc610285661)

# **List of Figures**

# **List of Tables**

# **RESEARCH PROPOSAL**

## **Abstract**

The Face Recognition Student Attendance System aims to revolutionize the process of recording student attendance in educational institutions by leveraging facial recognition technology. This system is designed to register student presence with high accuracy and efficiency, eliminating the need for traditional manual methods such as roll calls and paper sign-in sheets. By automating attendance tracking, this innovative solution not only saves valuable time for both students and teachers but also enhances the security and reliability of attendance data. The system offers real-time reporting and alerts, enabling administrators to monitor attendance patterns and promptly address any issues. This research focuses on developing a user-friendly, secure, and dependable system that tackles common challenges in attendance tracking, including time consumption, human error, and fraudulent sign-ins.

## **1.1 Introduction**

Attendance tracking is a critical aspect of educational administration, playing a significant role in monitoring student participation, ensuring academic compliance, and maintaining institutional records (Ribeiro-Navarrete et al., 2021). Traditional methods of recording attendance, such as manual roll calls or paper sign-in sheets, are often time-consuming, prone to human error, and susceptible to fraudulent practices (Wenjian et al., 2020). These conventional approaches can also disrupt valuable instructional time, impacting both teaching and learning experiences (Chowdhury et al., 2019).

In recent years, advancements in technology have paved the way for more efficient and reliable solutions. One such innovation is the use of facial recognition technology, which offers a seamless and automated alternative to traditional attendance tracking methods (Jain et al., 2022). Facial recognition systems utilize sophisticated algorithms to identify and verify individuals based on their unique facial features, providing a highly accurate and efficient means of recording attendance (Zhao et al., 2020).

The Face Recognition Student Attendance System aims to harness the power of this cutting-edge technology to transform attendance management in educational institutions. By automating the attendance process, the system not only streamlines operations but also enhances data security and accuracy (Gupta et al., 2021). Real-time attendance reporting and alerts further enable administrators to monitor student presence, identify patterns, and address issues promptly (Li & Lu, 2019).

This research proposal focuses on the development of a user-friendly, secure, and reliable Face Recognition Student Attendance System. The proposed system will address common challenges associated with traditional attendance methods, such as time consumption, human error, and fraudulent sign-ins, offering a modern solution that benefits both students and educators. The following sections will detail the objectives, methodology, and expected outcomes of this research, providing a comprehensive roadmap for the successful implementation of the system.

## **1.2. Background to the Study**

The accurate tracking of student attendance is crucial for educational institutions, impacting not only administrative efficiency but also student performance and institutional accountability (Deka et al., 2018). Traditional attendance methods, such as roll calls and sign-in sheets, have been in use for decades. However, these methods are increasingly viewed as outdated due to their susceptibility to errors, time consumption, and potential for manipulation (Sharma et al., 2019).

Recent advancements in biometric technologies have introduced more sophisticated and reliable methods for attendance tracking. Among these, facial recognition technology stands out due to its non-intrusive nature and high accuracy. Unlike fingerprint or retina scans, facial recognition does not require physical contact, making it more hygienic and user-friendly, particularly in the context of a pandemic or other public health concerns (Nguyen et al., 2020).

Facial recognition systems work by capturing an image of an individual's face and comparing it to a database of stored images to find a match. This technology has been widely adopted in various sectors, including security, banking, and retail, due to its effectiveness and efficiency (Jain et al., 2021). In education, implementing such systems for attendance tracking can significantly reduce administrative burdens, eliminate human errors, and prevent fraudulent sign-ins (Zhang & Liu, 2017).

Despite its potential, the adoption of facial recognition technology in educational settings raises several concerns. Privacy and data security are primary issues, as the system involves collecting and storing sensitive biometric information. Ensuring that this data is handled responsibly and securely is paramount to gaining trust and compliance from students, parents, and educational staff (Rahman et al., 2019). Additionally, the system must be user-friendly and seamlessly integrate into existing school infrastructure to maximize its effectiveness and acceptance (Kumar et al., 2020).

This study aims to address these concerns by developing a robust and secure Face Recognition Student Attendance System tailored for educational institutions. The system will incorporate advanced encryption methods to protect biometric data and user-friendly interfaces to facilitate ease of use. By tackling the challenges of traditional attendance methods and leveraging the benefits of modern technology, this study seeks to contribute to the enhancement of administrative processes and overall educational experience.

## **1.3. Problem Statement**

Businesses often struggle with limited resources to manage software development projects, leading to missed opportunities and inefficiencies (Bryman, 2020). Concurrently, freelancers face challenges such as inconsistent work, delayed payments, and difficulties in finding clients who match their skill sets (Bryman, 2020). According to a survey by Upwork, 7.3 million Ghanaians freelanced in 2019, contributing over $1 trillion (about $3,100 per person in Ghana) to the economy, yet many report struggles with job security and client acquisition (Upwork, 2023).

Despite the existence of various platforms, such as FreeLance-Me, Upwork, and MeetDev, gaps remain in effectively connecting businesses with qualified freelancers in a way that ensures both quality and satisfaction (Davidson, 2023).

This research proposes the design and implementation of SwiftCode, a web-based platform aimed at optimizing short-term software development collaborations between freelancers and businesses. SwiftCode will leverage advanced technologies and user-centered design principles to create a seamless and efficient experience for matchmaking, communication, and project management.

## **1.4. Aims and Objectives**

### **1.4.1. Aim**

The primary aim of this research is to develop **SwiftCode**, a tech-social platform that effectively connects businesses with skilled freelancers for short-term software development projects, addressing the existing challenges of reliability, quality control, and cost-effectiveness.

### **1.4.2. Objectives**

To achieve the stated aim above, this research pursues the following objectives:

* Conduct a comprehensive analysis of the specific software development needs of businesses, particularly SMEs.
* Investigate the challenges faced by businesses in hiring qualified freelancers for short-term projects.
* Examine the issues freelancers encounter in finding relevant job opportunities that match their skill sets and offer fair compensation.
* Implement robust quality control measures, such as freelance verification, ratings, and reviews, to ensure reliable and experienced professionals are matched with projects.
* Design a fee structure that is affordable for SMEs, reducing the financial barriers to accessing skilled freelancers.
* Offer flexible payment options and project management tools that help businesses manage budgets effectively.
* Provide businesses with detailed freelancer profiles and project histories to make informed hiring decisions.
* Create a marketing strategy to attract businesses and freelancers to the platform, highlighting the benefits and unique features of **SwiftCode**.
* Establish partnerships with industry associations, trade groups, and educational institutions to broaden the platform's reach and credibility.
* Implement mechanisms for continuous feedback from users to identify areas for improvement.
* Regularly update the platform based on user feedback and technological advancements to maintain its relevance and effectiveness in meeting the needs of both businesses and freelancers.

By addressing these objectives, this research aims to provide a holistic understanding of the specific needs and challenges faced by businesses and freelancers in the short-term software development market. It also aims to offer practical solutions through the development of **SwiftCode**, a platform designed to foster an efficient, reliable, and cost-effective connection between businesses and freelancers.

## **1.5. Research Questions**

To guide this research, the following questions will be explored:

* What specific software development needs do businesses, particularly SMEs, have for short-term projects?
* What challenges do businesses face in hiring qualified freelancers for these projects?
* What issues do freelancers encounter when seeking short-term job opportunities that align with their skills and provide fair compensation?
* How can **SwiftCode** implement effective quality control measures to ensure the reliability and competence of freelancers?
* What fee structures can be designed to make **SwiftCode** affordable for SMEs while ensuring fair compensation for freelancers?
* What flexible payment options and project management tools can **SwiftCode** offer to help businesses manage their budgets effectively?
* How can detailed freelancer profiles and project histories be presented to assist businesses in making informed hiring decisions?
* What marketing strategies can attract both businesses and freelancers to use **SwiftCode**?
* How can partnerships with industry associations, trade groups, and educational institutions enhance the platform's reach and credibility?
* What mechanisms can be established for continuous user feedback to improve the platform?
* How can **SwiftCode** be regularly updated based on user feedback and technological advancements to remain relevant and effective?

## **1.6. Significance**

The development of **SwiftCode** holds significant importance for multiple stakeholders in the software development ecosystem. This section highlights the key contributions and potential impacts of the proposed platform:

* **For Businesses:**

**Enhanced Resource Management: SwiftCode** will provide businesses, especially SMEs, with a reliable platform to source skilled freelancers, thereby optimizing their resource management. This can lead to more efficient project completions and reduced operational costs.

**Quality Assurance:** The platform's robust quality control measures, including freelancer verification, ratings, and reviews, will ensure businesses can hire competent professionals. This will improve project outcomes and business satisfaction.

**Cost-Effectiveness:** By offering an affordable fee structure and flexible payment options, **SwiftCode** will lower the financial barriers for businesses to access skilled freelancers. This can lead to increased competitiveness and growth for SMEs.

* **For Freelancers:**

**Consistent Work Opportunities: SwiftCode** will address the issue of inconsistent work by providing freelancers with a steady stream of relevant job opportunities that match their skills and experience.

**Timely Payments:** The platform will ensure freelancers receive timely payments, mitigating the common issue of delayed compensation.

**Professional Growth:** Detailed profiles and project histories will help freelancers build a reputable online presence, enhancing their career prospects and job security.

* **For the Economy:**

**Economic Contribution:** By facilitating more efficient and productive collaborations between businesses and freelancers, **SwiftCode** can contribute to economic growth. This is particularly relevant in contexts like Ghana, where freelancing is a significant economic activity.

**Job Creation:** The platform will create new job opportunities for freelancers, thereby contributing to employment and economic stability.

* **For the Technology Sector:**

**Innovation: SwiftCode** will leverage advanced technologies and user-centered design principles, contributing to innovation in the tech-social platform space.

**Best Practices:** The research and development of **SwiftCode** will establish best practices for quality control, project management, and user engagement in online freelancing platforms.

* **For Academic Research:**

**New Insights:** This research will provide new insights into the specific needs and challenges faced by businesses and freelancers in the short-term software development market.

**Practical Solutions:** The findings and solutions proposed in this research can inform future studies and developments in similar fields, bridging gaps in existing literature and practice.

By addressing these areas, SwiftCode aims to create a more efficient, reliable, and equitable ecosystem for short-term software development projects, benefiting businesses, freelancers, and the broader economy.

## **1.7. Justification**

The development of **SwiftCode** is justified by several compelling factors that highlight the need for an improved platform to connect businesses with freelancers for short-term software development projects. These factors are grounded in current market trends, challenges faced by both businesses and freelancers, and the potential benefits of a dedicated solution.

**Market Trends**

* **Growing Freelance Economy:**

Freelancers now constitute a sizable portion of the global workforce, with an estimated 1.57 billion people, or 46.7% of the world population, freelancing or being self-employed as of February 2024. This trend is projected to grow at an annual rate of 2.60% by 2031. The increasing number of freelancers necessitates a platform that can effectively connect them with suitable job opportunities.

* **Shift Towards Project-Based Work:**

Many businesses, particularly SMEs, are turning to freelance platforms to hire skilled professionals on a project basis. This shift allows businesses to access a diverse talent pool and achieve cost-effectiveness when scaling up without the commitment of hiring full-time employees.

**Challenges Faced**

* **Difficulty in Finding Qualified Freelancers:**

A study by the Web-District Manpower Group reveals that 69% of employers report difficulty filling technical positions, particularly for skilled roles like software development and data analysis. This indicates a significant gap in the market that **SwiftCode** can address by providing a reliable source of qualified freelancers.

* **High Fees and Quality Control Issues on Existing Platforms:**

Existing freelance platforms often charge high fees, which can be a deterrent for SMEs with tight budgets. Additionally, the lack of effective quality control measures means businesses risk hiring freelancers who may not deliver the desired results, leading to wasted time and resources.

**Potential Benefits**

* **Efficient and Reliable Connections:**

**SwiftCode** aims to streamline the connection process between businesses and freelancers, ensuring that both parties can efficiently and reliably find suitable matches. This reduces the time and effort spent on recruitment and enhances the likelihood of successful project outcomes.

* **Cost-Effectiveness for SMEs:**

By offering an affordable fee structure and flexible payment options, **SwiftCode** makes it feasible for SMEs to engage skilled freelancers without exceeding their budgets. This supports their ability to remain competitive and responsive to market demands.

* **Enhanced Employment Opportunities for Freelancers:**

Freelancers benefit from a platform that offers relevant job opportunities, fair compensation, and tools to display their skills. This can lead to increased job satisfaction, financial stability, and professional growth.

* **Promotion of Global Collaboration:**

Leveraging the rise of remote work and digital communication tools, **SwiftCode** facilitates collaboration between businesses and freelancers across geographical boundaries. This expands the talent pool available to businesses and provides freelancers with access to a wider array of job opportunities.

In conclusion, the justification for developing **SwiftCode** is clear: it addresses significant market needs and challenges, provides substantial benefits to both businesses and freelancers, and leverages current trends in the freelance economy and remote work. By creating a more efficient, reliable, and cost-effective platform, **SwiftCode** has the potential to transform the short-term software development market, fostering innovation and economic growth.

## **1.8. Limitation and Delimitation**

### **1.8.1. Limitations**

* **Limited Initial User Base:**

At the launch, **SwiftCode** might have a limited number of users, both businesses and freelancers, which can affect the diversity and availability of job opportunities. Building a robust user base will take time and strategic marketing efforts.

* **Market Competition:**

Existing platforms like **Upwork** and **Freelancer** have established user bases and brand recognition. Competing against these giants may pose significant challenges in attracting users to **SwiftCode**.

* **Quality Control Challenges:**

Despite implementing verification and review systems, ensuring consistent quality and reliability of freelancers can be challenging. Misalignment between freelancer capabilities and project requirements may still occur.

* **Technological Barriers:**

Developing a technologically advanced and user-friendly platform requires substantial resources and expertise. Potential technical issues, such as platform downtime or bugs, can affect user experience and trust.

* **Regulatory and Legal Issues:**

Navigating different regulatory environments and legal frameworks, especially in a global context, can be complex. Ensuring compliance with various laws regarding freelance work and data protection is essential but challenging.

* **User Adoption and Retention:**

Attracting users to the platform is only the first step; retaining them is equally crucial. Continuous engagement and satisfaction require ongoing updates, improvements, and customer support.

### **1.8.2. Delimitations**

* **Focus on Short-Term Projects:**

**SwiftCode** will primarily focus on short-term software development projects rather than long-term or permanent hiring solutions. This specialization helps streamline the platform’s offerings but limits its scope to specific project types.

* **Targeted User Demographics:**

The platform will initially target SMEs and freelancers in the software development field. While this focus ensures a tailored experience, it excludes other types of businesses and freelancers from different industries.

* **Geographic Scope:**

In its initial phase, **SwiftCode** may focus on specific regions or countries to manage growth effectively and ensure regulatory compliance. This geographical limitation helps in addressing local market needs but may restrict global accessibility initially.

* **Feature Prioritization:**

The development of **SwiftCode** will prioritize essential features like user profiles, project listings, communication tools, and payment systems. Advanced features and integrations will be introduced gradually, which may limit the platform’s functionality at launch.

* **Fee Structure:**

To attract SMEs, **SwiftCode** will implement an affordable fee structure. However, this may limit revenue generation in the short term, affecting initial profitability and investment capacity for further platform development.

* **Resource Allocation:**

The development team will focus resources on creating a robust, scalable platform and building a user base. Marketing, customer support, and additional services will be developed progressively, which may initially affect the platform’s reach and user experience.

By acknowledging these limitations and delimitations, the development of **SwiftCode** aims to provide a balanced and realistic analysis of the current freelance marketplace for short-term software development projects. The platform’s design, implementation, and strategic focus are intended to be practical and actionable, contributing to a more efficient, reliable, and inclusive solution for connecting businesses with skilled freelancers.

## **1.9. Preliminary Literature Review**

The literature on freelance platforms and the gig economy provides valuable insights into the challenges and opportunities within this sector. Studies have highlighted several key issues faced by businesses and freelancers when using existing platforms. For instance, Shevchuk & Trebor (2023) discuss the high costs associated with hiring full-time developers for short-term projects and the need for more flexible solutions. Zappa, P., et al., (2023) emphasize the difficulties businesses face in finding reliable freelancers and the inefficiencies in the current matching processes. According to Muhammad et al., (2022) freelancers often struggle to find job opportunities that match their skill sets and offer fair compensation, leading to underemployment and job dissatisfaction.

The "*Global Impact of Freelancing*" report by Upwork (2024) provides a comprehensive overview of the freelance workforce, noting the significant portion of the global population engaged in freelancing and the expected growth in this sector (Sean Cope, 2024). Mobil-Sans (2024) project a continued increase in the freelance population, driven by the rise of remote work and digital tools. Studies by the Web-District Manpower Group (MPG, 2024) reveal that a substantial percentage of employers' report difficulty in filling technical positions, particularly in software development and data analysis. This underscores the need for a platform like **SwiftCode** that can effectively connect businesses with skilled freelancers.

Zappa, P., et al., (2023) discuss the benefits of using freelance platforms for SMEs, highlighting the flexibility and cost-effectiveness these platforms offer. However, they also point out the challenges associated with high platform fees and the lack of effective quality control measures. Meredith & Rosenbaum, (2024) further elaborate on the risks businesses face when hiring freelancers from existing platforms, including project delays and subpar results due to inadequate vetting processes.

Lehdonvirta, (2018) emphasizes the growing demand for platforms that leverage digital communication tools to facilitate remote work and global collaboration. They argue that such platforms can broaden the talent pool available to businesses and provide freelancers with access to a wider array of job opportunities. This aligns with the objectives of **SwiftCode**, which aims to promote remote work and enhance global collaboration.

In summary, the literature highlights the significant challenges and opportunities within the freelance economy. By addressing these issues, **SwiftCode** has the potential to transform the short-term software development market, providing a more efficient, reliable, and cost-effective solution for connecting businesses with skilled freelancers.

## **1.10. Research Methodology**

The research methodology section outlines the tools, techniques, procedures, and processes applied in the development of **SwiftCode**. This section also elaborates on the research design, materials and methods, system requirements, and testing tools necessary for developing and evaluating the platform.

The development of **SwiftCode** will utilize various tools and existing solutions to ensure efficient and effective platform creation. The primary development tools include Integrated Development Environments (IDEs) such as Visual Studio Code and IntelliJ IDEA, which facilitate code writing and debugging. The programming languages chosen for this project are JavaScript, Python, HTML, and CSS, providing a robust and flexible foundation for both front-end and back-end development. Frameworks and libraries like React.js, Node.js, and Django will be employed to streamline the development process and enhance the platform's functionality. For database management, MySQL and MongoDB will be used to store and manage data securely and efficiently. Version control will be handled using Git and GitHub, ensuring collaborative development and efficient tracking of code changes. Project management tools like Jira and Trello will be used to organize tasks, track progress, and facilitate team collaboration.

In terms of techniques and procedures, the Agile Methodology will be applied to ensure iterative development and continuous feedback, allowing for regular updates and improvements based on user input. User-Centered Design (UCD) principles will be followed to create a user-friendly interface, ensuring that the platform meets the needs and expectations of its users. Quality Assurance (QA) measures, including verification, ratings, and reviews, will be implemented to maintain high standards and reliability, ensuring that businesses and freelancers can trust the platform for their short-term software development projects.

### **1.10.1. Research Design**

The research design employed for developing **SwiftCode** is **Descriptive**. This design was chosen to provide a comprehensive analysis of the current market needs, user requirements, and the functionality of the platform. Descriptive research is suitable for understanding the characteristics and features that **SwiftCode** must incorporate to address the problems faced by businesses and freelancers.

### **1.10.2. Research Methods**

* **Design Science**

The Design Science approach focuses on the development and performance of a software product. This involves creating innovative artifacts and processes that effectively solve an identified problem or set of problems. The iterative design and evaluation process ensures that the product meets its intended purpose and provides value to users.

* **System Development Life Cycle (SDLC)**

SDLC would be used in the development of **SwiftCode.** The chosen SDLC for developing **SwiftCode** is the Agile Model. This model emphasizes iterative development, allowing for regular updates and feedback incorporation, which is crucial for creating a user-centered platform.

**Reason for Choosing Agile:**

* Allows for flexibility and iterative improvements.
* Encourages continuous stakeholder engagement.
* Adapts to changing requirements effectively.

**Procedure:**

1. **Requirement Analysis:** Gather requirements from stakeholders through surveys and interviews.
2. **Design:** Create wireframes and prototypes for the user interface.
3. **Development:** Implement features in iterative sprints.
4. **Testing:** Conduct unit testing, integration testing, and user acceptance testing (UAT).
5. **Deployment:** Deploy the platform to a staging environment for final testing.
6. **Maintenance:** Regularly update the platform based on user feedback and technological advancements.

**Mixed Method**

A Mixed Method approach will also be used, combining quantitative and qualitative data to gain a comprehensive understanding of user needs and platform performance.

**Quantitative Methods:**

* Surveys to gather numerical data on user preferences and satisfaction.
* Analytics to track platform usage and performance metrics.

**Qualitative Methods:**

* Interviews and focus groups to gather in-depth feedback from users.
* Usability testing sessions to observe user interactions with the platform.

### **1.10.3. Research Population**

The research population for this study consists of two main groups: businesses and freelancers. Each group plays a crucial role in the development and validation of **SwiftCode**, ensuring that the platform effectively meets their respective needs and requirements.

**Businesses**

1. **Small and Medium-Sized Enterprises (SMEs):**

**Definition:** SMEs are businesses with a limited number of employees and low revenue compared to larger enterprises.

**Reason for Inclusion:** SMEs often have short-term software development needs that do not justify hiring full-time developers. They are the primary target users of **SwiftCode** due to their need for cost-effective and flexible solutions.

1. **Startups:**

**Definition:** Startups are newly established businesses in their preliminary stages of operation.

**Reason for Inclusion:** Startups typically require quick development cycles and agile responses to market demands. They benefit from a platform like **SwiftCode** that can connect them with skilled freelancers for rapid development.

1. **Technology Firms:**

**Definition:** Established technology firms that may require additional development resources for specific projects.

**Reason for Inclusion:** Even established firms may face temporary spikes in development workload that their in-house teams cannot handle alone. **SwiftCode** provides these firms with access to additional skilled resources on a project basis.

**Freelancers**

1. **Software Developers:**

**Definition:** Freelancers with skills in various programming languages, frameworks, and development tools.

**Reason for Inclusion:** Software developers are the core service providers on the platform. Their skills and availability are crucial for the success of **SwiftCode**.

1. **UI/UX Designers:**

**Definition:** Freelancers specializing in user interface and user experience design.

**Reason for Inclusion:** UI/UX designers contribute to creating visually appealing and user-friendly applications, which are often part of short-term software development projects.

1. **Quality Assurance Testers:**

**Definition:** Freelancers who specialize in testing software for bugs, performance issues, and overall quality.

**Reason for Inclusion:** Quality assurance testers ensure that the developed software meets the required standards and performs as expected, making them essential for project success.

1. **Project Managers:**

**Definition:** Freelancers with expertise in managing and coordinating software development projects.

**Reason for Inclusion:** Project managers help ensure that projects are completed on time, within scope, and to the desired quality, making them valuable assets for businesses using **SwiftCode**.

### **1.10.4. Research Sample**

**Sampling Strategy**

1. **Businesses:**

**Sampling Method:** Stratified sampling will be used to ensure representation across different business types (SMEs, startups, technology firms) and industries.

**Sample Size:** A minimum of 50 businesses will be surveyed, with an equal distribution among the different business types to ensure a comprehensive understanding of their needs and challenges.

1. **Freelancers:**

**Sampling Method:** Purposive sampling will be employed to select freelancers with varying levels of experience and expertise in different areas of software development.

**Sample Size:** At least 100 freelancers will be included, ensuring diversity in skills, experience levels, and geographical locations.

### **1.10.5. Research Instrument**

**Data Collection Methods**

1. **Surveys:**

**Purpose:** To gather quantitative data on the needs, preferences, and challenges faced by both businesses and freelancers.

**Distribution:** Online surveys will be distributed via email and social media platforms.

1. **Interviews:**

**Purpose:** To collect qualitative insights and in-depth feedback from selected participants.

**Format:** Semi-structured interviews conducted via video conferencing.

1. **Focus Groups:**

**Purpose:** To facilitate discussions among groups of businesses and freelancers, providing a platform for idea exchange and collaborative feedback.

**Format:** Online focus group sessions with 5-10 participants each.

1. **Usability Testing:**

**Purpose:** To observe user interactions with the platform and identify areas for improvement.

**Format:** Remote usability testing sessions with screen recording and think-aloud protocols.

By carefully selecting and engaging a diverse research population, this study aims to ensure that **SwiftCode** is designed and developed to effectively address the real-world needs and challenges of its users.

### **1.10.6. Functional and Non-Functional Requirements**

The development of **SwiftCode** necessitates a clear delineation of both functional and non-functional requirements to ensure the platform effectively meets the needs of its users.

Functional requirements focus on the specific behaviors and functionalities the system must possess, such as user authentication, project posting, and freelance search capabilities. Non-functional requirements, on the other hand, encompass the quality attributes that determine the performance and user experience of the platform, including security, reliability, and user interface friendliness. Together, these requirements form the foundation for creating a robust, secure, and user-centric platform that seamlessly connects businesses with skilled freelancers for short-term software development projects.

**Functional Requirements**

1. **User Registration and Authentication:**

* Users (both businesses and freelancers) will be able to create accounts using email or social media credentials.
* Users will complete profile verification through email confirmation or two-factor authentication (2FA).

1. **Profile Management:**

* Freelancers will be able to create and manage profiles detailing their skills, experience, portfolio, and availability.
* Businesses will be able to create profiles specifying their industry, size, and project requirements.

1. **Project Posting and Bidding:**

* Businesses will be able to post project descriptions, requirements, timelines, and budgets.
* Freelancers will be able to search for projects, submit bids, and negotiate terms.

1. **Matching and Recommendations:**

* The platform will provide automated recommendations of freelancers to businesses based on project requirements and freelancer profiles.
* The platform will notify freelancers of projects that match their skills and interests.

1. **Communication Tools:**

* Users will have access to a secure messaging system for real-time communication.
* The platform will support video conferencing for virtual meetings between businesses and freelancers.

1. **Payment Processing:**

* The platform will facilitate secure payment transactions, including escrow services to hold funds until project milestones are completed.
* The platform will support multiple payment methods, including credit/debit cards, bank transfers, and digital wallets.

1. **Rating and Review System:**

* Users will be able to rate and review each other upon project completion.
* The platform will display ratings and reviews on user profiles to maintain transparency and trust.

1. **Project Management Tools:**

* The platform will provide project management features such as task assignment, progress tracking, and milestone setting.
* Businesses and freelancers will be able to upload and share project-related documents and files.

**Non-Functional Requirements**

1. **Security:**

* Data Encryption: All user data will be encrypted in transit and at rest using industry-standard encryption protocols.
* Access Control: Role-based access control (RBAC) will be implemented to restrict access to sensitive information based on user roles.
* Data Privacy: The platform will comply with data protection regulations such as GDPR to safeguard user privacy.

1. **Reliability and Performance:**

* Uptime: The platform will guarantee a minimum of 99.9% uptime to ensure availability.
* Scalability: The platform will be able to handle increased user traffic and data load as the user base grows.
* Load Balancing: Load balancing mechanisms will be implemented to distribute traffic evenly across servers.

1. **User Interface Friendliness:**

* Intuitive Design: The platform will have a user-friendly interface with easy navigation, clear instructions, and consistent design elements.
* Accessibility: The platform is accessible to users with disabilities by adhering to WCAG (Web Content Accessibility Guidelines).
* Responsiveness: The platform will be responsive and perform well across various devices and screen sizes, including desktops, tablets, and smartphones.

1. **Performance:**

* Response Time: The platform will have fast response times, with pages loading within 3 seconds under normal conditions.
* Latency: Latency is minimized in data processing and communication features to ensure real-time interactions.
* Resource Efficiency: The platform will be optimized to use server and database resources efficiently, reducing overhead and operational costs.

1. **Quality Assurance:**

* Testing: Comprehensive testing will be frequently conducted, including unit testing, integration testing, and user acceptance testing (UAT), to identify and fix bugs.
* Continuous Improvement: Mechanisms will be implemented for continuous feedback and regular updates to address user needs and technological advancements.
* Documentation: Thorough documentation would be provided for users, including FAQs, tutorials, and customer support contact information.

By addressing these functional and non-functional requirements, the development of **SwiftCode** aims to create a secure, reliable, and user-friendly platform that meets the needs of businesses and freelancers in the short-term software development market.

### **1.10.7. System Design**

**Use Cases**

The primary users of **SwiftCode** are businesses seeking software development services and freelancers offering their skills. The processes involved include user registration, job posting, freelancer search, job application, project management, and payment processing. Here are the main use cases:

1. **User Registration:**

Both businesses and freelancers need to register on the platform.

1. **Profile Management:**

Users can update their profiles, including skills for freelancers and requirements for businesses.

1. **Job Posting:**

Businesses post job requirements.

1. **Freelancer Search:**

Businesses search for freelancers based on skills, ratings, and availability.

1. **Job Application:**

Freelancers apply for posted jobs.

1. **Project Management:**

Includes communication, milestone tracking, and file sharing.

1. **Payment Processing:**

Handling payments from businesses to freelancers.

1. **Review and Rating:**

Both parties can review and rate each other after the project is completed.

**Context-Level Diagram**

The context-level diagram provides a high-level overview of the system, showing the interactions between the system and external entities.

The context-level diagram provides a high-level overview of the interactions within the **SwiftCode** system. The system has three main external entities: businesses, freelancers, and the payment gateway. Businesses interact with **SwiftCode** by posting jobs, searching for freelancers, managing ongoing projects, and making payments. Freelancers use the system to register, apply for jobs, manage projects, receive payments, and provide reviews. The payment gateway facilitates financial transactions between businesses and freelancers. This diagram helps to visualize the system boundaries and the primary interactions between the users and the system.

**Entity-Relationship Diagram (ERD)**

The ERD illustrates the data model for **SwiftCode**, showing the relationships between key entities.

**Entities:**

1. User (UserID, Name, Email, Password, UserType)
2. Profile (ProfileID, UserID, Skills, Rating)
3. Job (JobID, BusinessID, Title, Description, Budget, Deadline)
4. Application (ApplicationID, JobID, FreelancerID, Proposal, Status)
5. Project (ProjectID, JobID, FreelancerID, Status, Milestones)
6. Payment (PaymentID, ProjectID, Amount, Date)

**Relationships:**

1. A User has one Profile.
2. A Business (User) posts many Jobs.
3. A Freelancer (User) applies for many Jobs.
4. A Job has many Applications.
5. A Job can be associated with one Project.
6. A Project has many Milestones.
7. A Project has many Payments.

The Entity-Relationship Diagram (ERD) illustrates the data model for the **SwiftCode** platform. It defines the entities involved and their relationships.

* The **User** entity represents all users on the platform, which can either be businesses or freelancers. Each user has a unique **Profile** represented by the Profile entity, containing details like skills and ratings.
* The **Job** entity represents the job postings created by businesses. A business can post multiple jobs, but each job is linked to a single business.
* Freelancers apply to jobs through the **Application** entity. Each job can receive multiple applications from different freelancers.
* Once a job is awarded to a freelancer, it is tracked as a **Project**. A project can have multiple milestones and involves both the business and the freelancer.
* Financial transactions related to the projects are represented by the **Payment** entity, which records payments made from businesses to freelancers.

This ERD helps in understanding the data structure and the relationships between various entities, which is crucial for database design and ensuring data integrity.

**Data Flow Diagram (DFD)**

The Data Flow Diagram (DFD) visualizes how data flows within the **SwiftCode** system.

**Level 0 DFD:**

* At the highest level, businesses and freelancers interact with the **SwiftCode** platform. Businesses perform activities like posting jobs, searching for freelancers, managing projects, and making payments. Freelancers register, apply for jobs, manage projects, receive payments, and leave reviews.
* **SwiftCode** communicates with the user database to store and retrieve user-related data, including job postings, applications, profiles, and project details.
* For payment processing, **SwiftCode** interacts with an external payment gateway to handle transactions between businesses and freelancers.

**Level 1 DFD:**

1. **User Registration:**

Businesses and freelancers provide their details, which are processed by **SwiftCode** and stored in the user database.

1. **Job Posting:**

Businesses submit job details, which are saved in the job database.

1. **Freelancer Search:**

Businesses search for freelancers using specific criteria, with results retrieved from the user database.

1. **Job Application:**

Freelancers apply for jobs, and their applications are stored in the application database.

1. **Project Management:**

Both parties manage project-related activities, which are tracked in the project database.

1. **Payment Processing:**

Businesses initiate payments through **SwiftCode**, which are processed by the payment gateway, and the transaction details are stored in the payment database.

This DFD helps in understanding the movement of data within the system, identifying potential bottlenecks, and ensuring efficient data processing.

**Sequence Diagrams**

Sequence diagrams illustrate the interaction between different entities over time for specific use cases.

**User Registration Sequence Diagram:**

This diagram illustrates the steps involved when a business or freelancer registers on the **SwiftCode** platform:

1. The user (business or freelancer) accesses the registration page on **SwiftCode**.
2. The user submits the registration form with their details.
3. **SwiftCode** processes the registration data and stores it in the user database.
4. **SwiftCode** sends a confirmation message back to the user, indicating successful registration.

**Job Posting Sequence Diagram:**

This diagram explains the process of a business posting a job on **SwiftCode**:

1. The business opens the job posting page on **SwiftCode**.
2. The business submits the job details through the form provided.
3. **SwiftCode** saves the job data in the job database.
4. **SwiftCode** sends a confirmation message to the business, confirming that the job has been posted successfully.

**Job Application Sequence Diagram:**

This diagram shows the steps involved when a freelancer applies for a job:

1. The freelancer accesses the job details page on **SwiftCode**.
2. The freelancer submits their application for the job.
3. **SwiftCode** processes the application and stores the application data in the application database.
4. **SwiftCode** sends a confirmation message to the freelancer, confirming that their application has been received.

**Payment Processing Sequence Diagram:**

This diagram details the steps involved in processing a payment from a business to a freelancer:

1. The business initiates a payment through **SwiftCode**.
2. **SwiftCode** forwards the payment details to the external payment gateway for processing.
3. The payment gateway processes the transaction and sends a confirmation back to **SwiftCode**.
4. **SwiftCode** stores the payment details in the payment database.
5. **SwiftCode** notifies the freelancer that the payment has been made.

These sequence diagrams help in understanding the interactions and the flow of operations within the system for specific use cases.

**User Interface Designs**

The user interface designs aim to provide an intuitive and user-friendly experience for both businesses and freelancers.

1. **Home Page:**

* The home page includes a navigation bar with links to key sections: Home, Jobs, Freelancers, Projects, Profile, and Payments.
* It features highlighted job postings and top-rated freelancers, making it easy for users to find relevant information quickly.

1. **Registration Page:**

* The registration page allows users to select their user type (business or freelancer) and fill out a registration form with their details.
* A submit button at the end of the form facilitates easy registration.

1. **Job Posting Page:**

* This page provides a form for businesses to enter job details such as title, description, budget, and deadline.
* A submit button ensures the job is posted after entering the details.

1. **Freelancer Search Page:**

* The freelancer search page includes a search bar for keywords and skills, along with filter options like rating, availability, and budget.
* The search results list displays freelancers matching the criteria, making it easy for businesses to find suitable candidates.

1. **Profile Page:**

* The profile page displays user information such as name, skills, and ratings.
* An edit profile button allows users to update their profile information as needed.

1. **Project Management Page:**

* This page provides details of ongoing projects, including a milestone tracker and a communication section for messages and file sharing.
* It ensures that both businesses and freelancers can manage projects efficiently.

1. **Payment Page:**

* The payment page includes a form for entering payment details and a history section to view past transactions.
* This ensures smooth and transparent financial transactions within the platform.

These UI designs ensure that the platform is easy to navigate, provides all necessary functionalities, and offers a seamless user experience for both businesses and freelancers.

By integrating these elements, **SwiftCode** aims to provide a comprehensive, efficient, and user-friendly platform that addresses the specific needs of the freelance software development market.

### **1.10.8. Minimum System Requirements**

1. **Hardware Requirements**

* **Processor:** Intel Core i3 or equivalent

The Intel Core i3 processor, or its equivalent, provides sufficient computational power for handling typical operations on the SwiftCode platform, such as browsing job listings, uploading profiles, and processing transactions. This ensures smooth performance without overburdening the system.

* **RAM:** 4 GB minimum

4 GB of RAM is the minimum required to run modern web applications efficiently. It allows for smooth multitasking and handling of dynamic content, which is common in a web-based freelancing platform where users might interact with various components simultaneously.

* **Storage:** 10 GB free disk space

At least 10 GB free disk space ensures there is ample space for storing essential application files, cache, and temporary data. This is important for maintaining performance and ensuring quick access to frequently used resources.

* **Display:** 1280 x 720 resolution

A display resolution of 1280 x 720 ensures that the user interface is adequately rendered, providing a clear and usable interface for users. This resolution is commonly supported by most modern displays, ensuring accessibility for a wide user base.

1. **Software Requirements**

* **Operating System:** Windows 10 or higher, macOS 10.14 or higher, Linux

Supporting Windows 10 or higher, macOS 10.14 or higher, and Linux ensures compatibility with the most widely used operating systems, providing flexibility and accessibility to a broad range of users.

* **Browser:** Latest version of Google Chrome, Mozilla Firefox, or Microsoft Edge

Ensuring users have the latest version of their browsers guarantees that they benefit from the latest security updates, performance improvements, and compatibility features. Modern browsers also support the latest web technologies essential for a rich user experience on the SwiftCode platform.

1. **Network Requirements**

* **Bandwidth:** Minimum 1 Mbps internet connection for smooth operation

A minimum bandwidth of 1 Mbps is sufficient for basic interactions with the platform, such as browsing job listings, updating profiles, and handling text-based communications. This ensures that users with average internet connections can use the platform without significant lag or interruptions.

* **Network Speed:** At least 5 Mbps for optimal performance

A network speed of at least 5 Mbps is recommended for optimal performance, ensuring that users can smoothly handle more data-intensive operations like video calls, large file uploads, and real-time collaboration without experiencing delays.

1. **User Requirements**

* **Age:** Minimum age of 18 years

Setting a minimum age of 18 years ensures that users are legally adults, capable of entering into contracts, and responsible for their own financial transactions. This is important for legal and ethical considerations, as the platform involves job contracts and payments.

* **Certification:** Relevant certifications or portfolio for freelancers to verify skills

Requiring relevant certifications or portfolios from freelancers ensures a certain level of skill and professionalism. This helps maintain the quality of services offered on the platform, providing businesses with confidence in the freelancers they hire and ensuring that freelancers are adequately qualified for the jobs they apply for.

These requirements are designed to ensure that the SwiftCode platform is accessible, functional, and efficient for its intended user base, providing a reliable and high-quality experience.

## **1.11. Research Ethics**

The proposed research will adhere to the highest ethical standards to ensure the protection, dignity, and rights of all participants. All participants will be provided with comprehensive information about the research, including its purpose, procedures, potential risks and benefits, and their rights as participants. Informed consent will be obtained from each participant before their involvement in the study. This process will include following; Participants will receive a detailed explanation of the research objectives, methods, and what their participation entails, Participation will be entirely voluntary, and participants will have the right to withdraw from the study at any time without any negative consequences, and a written consent forms will be used to document participants' agreement to take part in the research.

The confidentiality and anonymity of participants will also be strictly maintained throughout the research process. Measures to protect participants' privacy including data protection, anonymization, and confidentiality will be securely preserved. The research will be conducted with respect for the rights, dignity, and well-being of all participants while upholding transparency and honesty in all interactions with participants and stakeholders. The research will comply with relevant ethical standards and guidelines, including those set by institutional review boards (IRBs) and ethics committees of businesses that will be involved in this research.

## **References**

Deka, B., Biswas, R., & Bhattacharjee, A. (2018). An Efficient Attendance Management System based on Face Recognition Using Raspberry Pi and GSM Module. International Journal of Computer Applications, 182(25).

Jain, A., Sharma, P., & Kumar, N. (2021). Automated Attendance System Using Facial Recognition. International Journal of Computer Applications, 175(24).

Kumar, R., Aggarwal, R., & Singh, S. (2020). Development of Student Attendance Management System Using Face Recognition Technology. Journal of Engineering Technology, 9(3).

Nguyen, K., Dang, C., & Tran, D. (2020). Non-intrusive attendance management system using face recognition and machine learning. International Journal of Advanced Computer Science and Applications, 11(5).

Rahman, M. S., Islam, M. K., & Haque, M. (2019). Security and Privacy Concerns in the Use of Face Recognition Technologies: Literature Review and Technical Insights. International Journal of Information Management, 49, 101-118.

Sharma, S., Jindal, R., & Jain, P. (2019). Analysis of Traditional and Biometric Based Attendance System: A Comparative Study. International Journal of Engineering and Technology, 7(2).

Zhang, Z., & Liu, X. (2017). Design and Implementation of Smart Classroom System for Ubiquitous Learning. Journal of Information Technology and Application, 11(1).