Attn: Mrs. Rose Sharon

# A PROPOSAL FOR THE DEVELOPMENT OF AN ENTERPRISE RESOURCE PLANNING (ERP) SOFTWARE.

**INTRODUCTION**

We would like to thank you for meeting with us to discuss your company’s software project requirements.

Artificial Intelligence Technologies Ltd is one of the leading software development companies globally. We have the required expertise and track record in providing optimal performance and quality software products and services.

At Artificial Intelligence Technologies Ltd, we know that creating client-oriented software takes a mixture of technical excellence and clear communication. Our firm only hires the very best to ensure you receive both. We understand that every client is unique, so we strive to deliver an individual, innovative an affordable proposal every time, following through with an outstanding delivery that is both on time and within budget. We have more than 6 years’ experience of developing and deploying software solutions and Security Gadgets to our clients. To mention but a few:

* **Ministry of Youth and Sport**s: Development and maintenance of website with SEO optimization
* **Clearing Cost Agency**: An online shipping clearance cost service for importers and exporters
* **Telo Technologies**: A logistics marketplace that connects users to dispatch riders for pickup and drop-off service.

Please let us know if you would like to get in touch with these exciting clients to hear their testimonials about Artificial Intelligence Technologies Ltd. You can learn more from our website at [*www.aitechnologiesng.com.*](http://www.aitechnologiesng.com/)

We also pride ourselves on our after-sales client-care including our guarantees, staff training, and onsite and offsite support.

# NAME OF THE PRODUCT:

The name of the software system to be developed is yet to be discussed.

# OBJECTIVE OF THE PROJECT

The objective of this project is to develop a system that tracks and checkmates all the incomes and expenditures of certain government parastatals to ensure transparency of financial activities.

# PROJECT REQUIREMENTS

From the information we gathered from our conversation, the following are the requirements of the Stylemee platform.

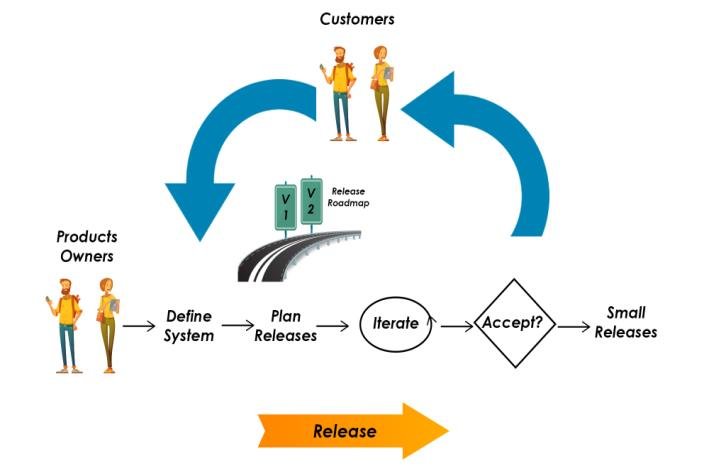
1. You want to be able to create three software products that will be integrated together for Skilled Professionals to create an account profile, add their products/services, receive orders (appointments) & push-up notifications.
2. End users should be able to get an organized, detailed categories of product/services that has been verified and updated from the skilled professional application database.
3. Users should have access to available product/services at that particular time the user tends to make an order. Users will be able to view a list of transactions they have performed.
4. Admin should have a control panel that maintains, monitors and controls the skilled professional’s software interface and the user interface. Admin should be able to view requests, receive orders, push-up notifications, configure databases and above all, having supreme privileges to tweak any functions & features within the software application.

Others are:

1. Use of Coding Standards
2. Use of Good Naming Conventions
3. File and Folder Naming and Organization
4. Adequate Commenting and Documenting
5. Application of Object-Oriented Programming Principles (Classes, Functions etc.)
6. Adequate Testing of the System

# METHODOLOGY

For the project, we would be using an internally developed agile methodology, which has proven useful over our years of experience to protect both our clients and our company.



**Diagram I: Client’s Involvement Using Agile Process**

The illustration above shows a basic demonstration of how the client would be involved in the development life cycle of the software. The client provides feedback at every stage of the life cycle, our development team receives the feedback and makes the appropriate changes as far as it aligns with the initial requirements of the software as specified in the User Interface design. This process continues till the full deployment of the product within **the specified project timeline**.

It should however be noted that after a feature has been mutually agreed to be satisfactory, any new changes made to it will attract extra charges. In addition, the core features of the product will not be changed without re-negotiation.

A more detailed illustration of the client’s involvement in the overall development life cycle of the software is shown below:

# 

**Diagram II: Software Development Life Cycle**

1. The client provides the functional requirements of the software, our development team analyzes it and sends back the attainable non-functional requirements, the client reviews it and sends feedback, the requirements are adjusted until an agreement is reached.
2. The planning and analysis stage takes place internally within our company.
3. The fully detailed design plan of how the software will be developed, the frameworks and technologies that will be used and other area of interest are sent to the client, the client gives feedback, the design plan is updated until both parties are satisfied.
4. Testing is carried out internally within our company. After the software is deployed, it is sent to the client for beta testing.

The feedback gotten from this stage is sent back to the development stage. The final product is then released to the public for use.

# CODE DEVELOPMENT LIFE CYCLE

# 

**Diagram III: Code Development Cycle**

The illustration explains the stages that comprises the actual code development life cycle.

* The codes are developed functionally in units. The units are then tested individually and then as a whole to make sure they integrate properly. Thereafter, the code is deployed in a staged environment within the company where alpha testing is performed. Feedback from the alpha test is sent back to the unit development stage for improvement, the cycle continues until the alpha test is successful.
* The full app is then deployed for beta testing by the client and finally released to the public. If the code fails any of the quality assurance tests, it falls back to the unit development stage.

# Each feature developed as a unit follows the plan –> build –> test –> review cycle shown below:

# 

# Diagram IV: Feature Development Cycle

# FUNCTIONAL REQUIREMENTS

Based on the above requirements, the Stylemee platform to be developed will consist of four functional modules and other sub-modules. This is to enhance modularity and flexibility in the development approach. They include but not limited to:

* 1. UI/UX (User Interface, User Experience) Template Design Module
  2. Admin Module
  3. User Controller Module
  4. Service Professionals Module
  5. Payment Controller Module

# UI/UX TEMPLATE DESIGN MODULE

A template specification which will form a fundamental design framework (if any) will be provided by us and the client is free to make adjustments based on the App designs, colors, and images. Once ascertained and approved, it will be converted into a unique User Interface design for the mobile application.

The template module will consist of:

* Content pages - (onboarding, login, signup, password recovery, categories, list of artisans with images (skilled professionals), artisan information, wallet, billing, invoice preview, transactions, admin/user dashboard, image slide shows, search bar and navigations)

# ADMIN MODULE

As a norm in mobile applications, the administrator will be empowered with the ability of having absolute and privileged control of the system. The administrator module will consist of the following sub-modules.

* Admin can register, login and also view all users
* Admin can add/edit/delete services
* Admin can add/edit/delete categories
* Admin can edit/delete the users’ data
* Admin can manage mobile app processes
* Admin can create a Coupon code for Discount
* Admin can add supported shipping states/countries
* Admin can add supported currencies
* Admin can add/edit/delete user’s comment and rating
* Generate Invoice in PDF
* Get Invoice copy on email

# Appointment and Reports

* Admin can view all appointments
* Admin can edit/delete appointments from the list
* Admin can manage the appointments of each artisan.
* Admin can accept/reject the order.
* Admin can filter by payment status and order number
* Admin can view the reports
* Admin can filter the reports for Date Range, Monthly, Weekly, Daily

# FAQ (Admin and User Functionality)

* Admin can view all the faqs
* Admin can add faqs
* Admin can edit/delete the existing faqs

# Client’s Added Features

# Admin can generate reports and export it in various file formats

# Admin can make special offers and discounts

# Admin can export data for analysis-pending any third-party onboarding AI for big data analytics in the open-source

# Admin can provide help online to the customers

# Admin can receive customer feedback and ratings for loyalty rewards

# USER MODULE

System user’s form an integral part of any mobile application and as such, top class user experience will be taken into cognizance during the development of this application in order to enhance their experience, business productivity and profitability. Hence, the user module will consist of the following sub-modules:

* Users can sign up for an account and Login
* Users can search for desired services or service professionals
* Users can book service professionals for an appointment and may apply coupon codes where necessary
* can edit place of service address and Information
* can track status of their appointments.
* can generate invoice of their appointments
* they can view/reschedule appointments.
* Users can make payment using either pay after service or any online payment options provided in-app
* can leave comments and ratings after service delivery

# SERVICE PROFESSIONAL MODULE

The service professional module will be developed to allow for service professionals to post their services on the Stylemee mobile application. As such, service professionals can:

* signup for an account
* Login to their account with credentials
* they can decline/view appointments.
* they can post the services they render with image and description, price and time allocation and other relevant information
* see clients' reviews
* withdraw their earnings

# PAYMENT CONTROLLER MODULE

We will be utilizing Paystack as the platforms’ online Payment gateway for wallet funding. With Paystack API, the following payment channels are available for your customers.

* + Debit and Credit Cards
  + Bank Account
  + Bank Transfer
  + USSD

To integrate this payment API for seamless and hitch free transactions on the website and Mobile App, we will take into cognizance, four programmable tools and services.

**Database** -This will store details of the transaction for easy retrieval and management **Webhook** – This will be used to run a check to the transaction end-point to verify that both servers of customer and that of the payment API returned success for a particular transaction. This is used to avoid double payments, debit of client without receipt of value etc. In a scenario whereby both end-points message don’t tally, the transaction is discontinued and marked as failed.

**SQL** – A set of queries to store, retrieve and commit transactions will be carried out in order to modularize and efficiently handle the transaction data.

**UI/UX** – This will be implemented using a front-end programming language. This is necessary to make the user interface responsive, attractive and painless.

**NON-FUNCTIONAL REQUIREMENTS**

All software modules explained above must meet the defined non-functional requirement limits at unit stage to ensure that only standardized units are integrated into the software. The non-functional requirements provided by our company are best implemented using cloud technologies. To make the most of this offer, we strongly advice you consider the cloud hosting option at the cost implementation section of this document.

**PERFORMANCE**

This is usually the first non-functional requirement we consider since it determines how fast the software returns result and how higher workloads affect that speed. We also put into consideration the quality of our code as it relates to memory and speed efficiency.

During development, our team of data structure analysts evaluates all the unit codes to ensure that they scale up to the highest attainable CPU and memory usage efficiency, with optimized speed as the primary target.

The system will be designed with requests per minute and bytes per request at unit stage. This means all our application API endpoints will be optimized to minimize bandwidth and optimize speed.

Our unit test strategy ensures that we can impose strict thresholds on these parameters and ensure quality at unit blocks.

Here are the performance metrics our company will provide for the software:

1. CPU usage
2. Memory usage
3. Requests per minute
4. Bytes per request

**SCALABILITY**

Our choice of architecture (microservices) provides a special scaling solution for the system. The architecture puts cost and scaling efficiency into consideration. With microservices, functionalities can be scaled individually, either vertically or/and horizontally. This means frequently and rarely used functionalities will not scale together (resulting in resource wastage), rather the frequently used functionalities will scale more than rarely used ones thereby saving cost.

Off the box, our solution will offer automatic vertical scaling (provision of resources) up to a preset maximum cost value that will be provided by the client. Once that value is exceeded, the client will have the option to initiate negotiations for the implementation of a horizontal scaling solution (provision of multiple servers) and *Kubernetes*.

The scalability metric our company will provide for the software is:

* Maximum number of simultaneous visits for each functionality

**AVAILABILITY**

Our choice of cloud deployment platform (Microsoft Azure) provides an off-the-box availability rate of 99.95% annually. However, we will deploy the system with the option of local redundancy which increases the rate to 99.99% annually. This nearly totally eliminates the possibility of downtime due to infrastructure failure.

**RELIABILITY**

Our solution will have the ability to recover from failures and continue to function as intended within a very short time. Our choice of deployment platform (Microsoft Azure) provides monitoring tools that will detect any failure that might arise while in production.

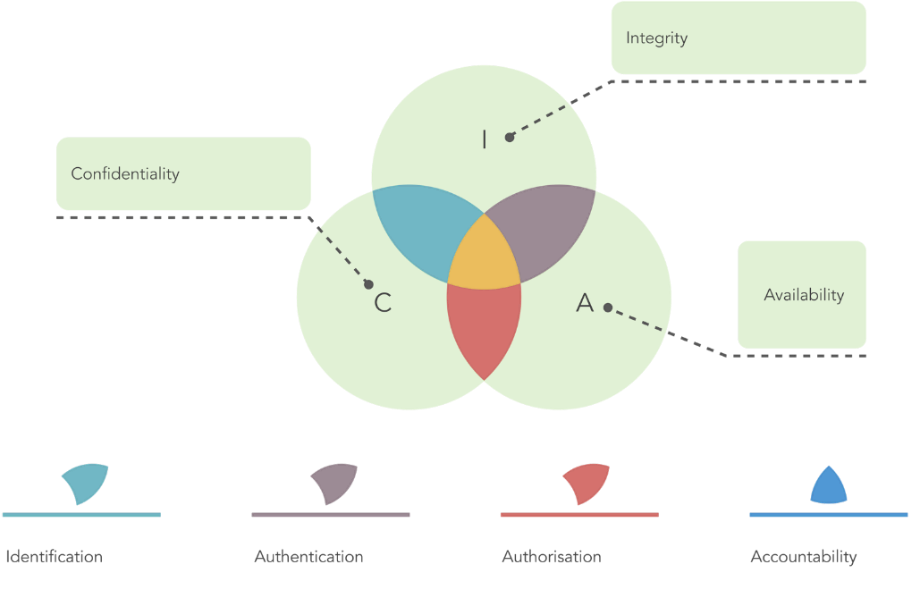
Also, during implementation, the application will be hosted separately from the storage to ensure that the data is not dependent on server.

The reliability metric our company will provide for the software is:

* Mean time to repair (MTTR)

**SECURITY**

Security is one of the most important non-functional requirements we provide to our client. Hence, we have put in a lot of consideration into the security of your application and have decided to use the CIA security triad.



**Diagram V: CIA Triad**

**CONFIDENTIALITY**

For confidentiality, our cloud service provider automatically provides security from elements outside the cloud. For security within the cloud, we use Identity and Access Management (IAM) to implement least privilege access principles. This ensures that unauthorized internal personnel cannot have access to information beyond the scope of what they need. Implementing this ensures data privacy.

The confidentiality non-functional requirement our company will provide for the client is:

* Hierarchical login tiers and roles for various technical staff

**INTEGRITY**

We will implement various strategies that ensure data cannot be altered or accessed by unauthorized personnel. We will implement generic security strategy like hashing, SSL, and others.

However, we will not stop there, we will also implement other advanced security feature that prevents bridge of data integrity.

The integrity non-functional requirements our company will provide for the client are:

* SSL protection
* Microsoft Azure firewall

**AVAILABILITY**

Distributed Denial of Service (DDOS) is one of the most basic yet dangerous attack that systems like this are naturally vulnerable to. Since we know this, our solution provides anti-DDOS**.**

The availability non-functional requirement our company will provide for the client is:

* Microsoft Azure DDOS protection

**DESIGN ARCHITECTURE**

We will use microservices as the design architecture for system; this is because of the various benefits the architecture provides. Some these benefits are functionality independence (upgrading of functions are performed independently without affecting the entire system), scalability (functions are scaled independently based on demand), easy maintainability, framework and language flexibility (languages are chosen based on performance on specific functionality).

The figure below is a similar architecture to our proposed architecture, we used it in a previous

application design. We will use it as a use case to explain how our proposed architecture should operate.

# 

# Diagram VI: Microservices Use Case

# The system will have three major parts similar to what is in the use case diagram;

# Mobile Application

# User Web Application

# Admin Web Application

These major parts will be interfaced with microservices architecture through an API gateway.

The design will be implemented to fully exploit microservices architecture by selecting the best database, the best language and framework as illustrated in our use case example above.

Consequently, for a start we have chosen to use two frameworks and two programming languages for backend development which are:

* Python – Django
* JavaScript – NodeJS
* For the frontend applications, we will use:
* Flutter – Dart (Mobile)
* ReactJS (Web)

**WHY DOCKER?**

We picked docker as our container service because moving forward, the need to scale our container instances with increase in demand will arise. Over our years of experience, we have come to realize that docker is one of the best and most efficient open-source container services for the job.

**SOFTWARE SPECIFICATIONS**

The Stylemee platform will be developed to run on the following platform:

**Operating System:** Android, iOS, and Web

**DATA PRIVACY**

Once negotiations are completed, you will be required to send in your Data Privacy \Policy as this will be needed when publishing the Apps to their respective stores.

**TIMELINE**

Our teams of software engineers have estimated that the project will be completed and delivered within 16 weeks (as detailed below) from date of receipt of initial payment.

|  |  |
| --- | --- |
| **Stage** | **Duration** |
| Planning | 1 week |
| Requirement Analysis | 1 week |
| Design and Prototyping | 2 weeks |
| Software Development | 7 weeks |
| Software Testing | 2 weeks |
| Implementation and Integration | 2 weeks |
| Operations and Management | 1 week |

**COST OF IMPLEMENTATION – Using Microsoft Azure Cloud Hosting (RECOMMENDED)**

|  |  |  |
| --- | --- | --- |
| S/N | DESCRIPTION | AMOUNT (N) |
| 1. | Dedication/Private Web Server Domain registration  Website Hosting on high-speed Microsoft Azure servers Unlimited Bandwidth  SSL for https:// secured connection Firewall/protection from DOS attacks Unlimited Disk Space  4GB RAM  Online Payment SDK Integration | To be provided by the client |
| 2. | Stylemee Mobile Application  En d U s e r s - (Android & iOS & Web)  Se r v i c e Pr o f e s s i o n a l s - (Android & iOS)  Upload to Google play store Upload to Apple App store | N2,500,000.00 |
| 3. | Implementation of Solution | N500,000.00 |
| 5. | Training | FREE |
| 6. | Support (3 MONTHS FREE) | FREE |
|  | GRAND TOTAL | ~~N~~3,000,000.0 |

**PAYMENT STRUCTURE**

The payment plan for the project will take the following troll:

50% before commencement of Development, 30% Five (5) Weeks after Commencement

20% before handover of project to client.

Kindly note that the price stated in the costing table is negotiable.

**YOUR ACTION:**

After review of proposal, if you need further details, our team of software engineers and developers will be readily available to answer your questions.

You can reach out to us via any of the following channels:

**WhatsApp/Phone Call** – +2348123181756, +2348163645640

**Email**: [info@aitechnologiesng.com,](mailto:info@aitechnologiesng.com) [mobileapps@aitechnologiesng.com,](mailto:mobileapps@aitechnologiesng.com)

aitechnologiesn [g@gmail.com](mailto:g@gmail.com)

Please, accept my assurances of highest regard.

Yours faithfully,

*For: Artificial Intelligence Technologies Limited*

**Joel G. Babalola**

**Director**