iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80257

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

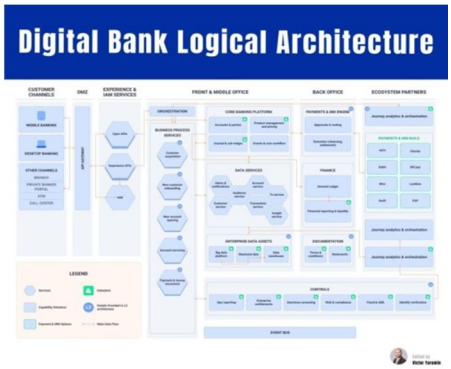
Bank Management System Remapped with Artificial Intelligence

Shubham Varshney[1], Suhail Khan[2]
Department of Design, Data Science and Cyber Security, Greater Noida Institute of Technology , Greater Noida Knowledge Park 2, Uttar Pradesh.

Abstract: ***The current financial services industry is witnessing a fundamental challenge marked by sustained disruptive digitalization and the gradual demise of traditional manual control models. This study explains the structural integration of Artificial Intelligence (AI) in Bank Management Systems (BMS), in particular, the "error absorbing" techniques aimed at reducing the "human uncertainty factor". This study, using the Agile Development approach and Java-JDBC infrastructure, attempts to resolve, within legacy IT systems and centralized hierarchical structures, the primary structural blockers to the deployment of FinTech solutions. The focus has been on developing a means of vertically transforming the approach from a 'product-oriented' to an 'AI-driven, customer-oriented' approach, using predictive validation to allow a financial institution to improve its reliability and productivity within its workforce. This study looks into the transformed workplace expectations and ways to measure outcomes, while it also critiques the structures of pay and the systems of financial incentives in a high-risk financial market. These components affecting equity and efficiency have a strong influence on productivity, retention and morale of employees.***

I. INTRODUCTION

The Crisis of Traditional Banking and the Shift to AI-Centricity the digital transformation of the banking sector is acknowledged as a constant and inevitable reality, and challenge, of the world's financial services industry. The last ten years or so have seen the digital transformation of the banking sector become a central focus of research in management, business, information systems, and information technologies. The rapid advancement of digital and communication technologies has created a new business environment called 'digital business ecosystems'. These ecosystems? fundamentally change the core strategies of businesses and the way they organize their resources and activities internally and the way they position themselves in markets externally.
It is the scale and the repetition of these technological changes that make the idea of change more tangible for contemporary organizations.



The Potential for Systemic Disruption The most pressing concern for traditional banking institutions is the fact that digital banking end-to-end service providers have not fully integrated digital banking systems. This results in strategic and operational roadblocks during the transformation journey. In this regard, innovation in banking services is present in the market as a new class of banking services called "challenger banks" that offer integrated multi-channel systems of banking services, customer engagement, and marketing. They gain a substantial competitive advantage and customer market share through high-speed digital platforms. If large banks and large regional banks don't keep pace with these technological changes, the security of banking and the economic system itself is compromised. This is a key driver for most financial service providers to begin thinking and implementing significant changes in their technological strategies to sustain competitive advantages.

©IJRASET: All Rights are Reserved

The Human Element of Uncertainty In the quest for organizational expansion and success, the role of Human Resources (HR) is critical. The achievements of top-performing employees are directly correlated to the success of the organization. However, the traditional system of payroll and management also brings a lot of “human errors” in the organization and is a source of a lot of negative impacts on a centrally designed IT system. This includes lack of clarity in the criteria for salary increase, opacity in the criteria for increase, lack of transparency in the determination of the criteria for the increase, and gross errors in the calculations of remuneration.

The motivational and job satisfaction levels of the staff, and subsequently, the performance levels of the organization, are severely impacted by these concerns. Therefore, the operational efficiency, and more fair and equitable systems of compensation, improved the opportunity areas. Data-driven systems are needed to create these equitable algorithms.

Research Objectives and Proposed Role of AI in the Bank Management System Research Objectives propose to study the bank management systems to explain the importance of operational efficiency and equitable practices to the organization. The study proposes that the modification of the payroll systems and other technology-induced systems will amplify the overall satisfaction and harmony in the organization’s environment. The proposition includes the development of an integrated computer-based Human Resource Management System (HRMS) that will utilize AI "error-absorption" middleware to facilitate the efficient identification and resolution of bottlenecks in the banking system, thus expediting process completion, optimizing procedural efficiency, and reducing administrative burdens. The aim is to establish an adaptable system that promotes organizational productivity and employee satisfaction while improving organizational viability.

The Scope of the Bank’s Digital Transformation The rapidly changing business environment presents an array of intricate changes, from strategic orientation and business models to competitive dynamics, decision-making, and overall productivity. In an open competitive environment, banks are compelled to shift to an integrated customer-centric approach. The adoption of digital technologies drives the development of novel business models and the enhancement of existing ones.

Implementation of new technological solutions to enhance business operations has shifted from being an option to a necessity for companies aspiring to sustain competitive advantages in the long run. In this regard, the recognition and assessment of factors that impede digital adaptation becomes highly relevant to the management accountable for the advancement of banking institutions.

II. THEORETICAL FRAMEWORK

Separating Digitization and Digitalization

Let’s get one thing straight—there’s a real difference between “digitization” and “digitalization,” even though people love to mix them up. Digitization is pretty basic: it’s just turning analog info into digital form, nothing more. Digitalization goes a lot deeper. It’s about rethinking how organizations work by weaving new digital tools and technologies into everything they do. When we talk about digital transformation, we’re not just talking about installing new hardware. We’re looking at a full-on change—strategy, culture, ways of thinking—moving on from worn-out old methods to agile, modern approaches, boosted by social, mobile, and smart technology.

A. RegTech and Autonomous Compliance Heuristics

With tougher regulations like Basel III in place, banks now lean on newer tools like Regulatory Technology, or RegTech. This isn’t just a buzzword—it’s how banks use advanced IT and digital tools to handle the mess of rules and compliance. Plugging in RegTech gives banks a strong digital backbone that supports both smart management and tight regulatory oversight. Take an AI-powered Bank Management System: these systems use built-in rules (or “heuristics,” to get technical) to automatically watch over compliance. They keep business objectives and regulations in sync, cutting out a lot of the manual checking and paperwork.

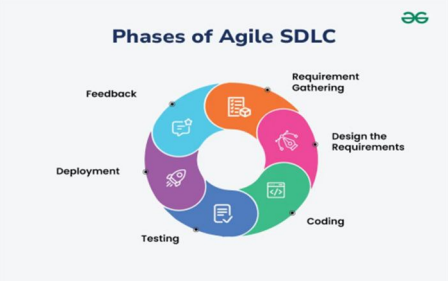


B. Sustainability as a Driver for Digital Evolution

Digitalization matters for more than speed or convenience—it can actually drive stability throughout the financial system and help reach global sustainability goals. Old business models just can't handle the complex economic and environmental challenges we face now. Here's where sustainability steps in: it pushes banks to adopt digital tools, making up for the gaps of legacy systems like poor transparency and inefficiency. Mix in AI-driven digitalization, and banks find themselves narrowing in on what they do best while offering a broader range of services. Sustainable fintech also encourages partnerships between big-name banks and newer FinTech players. The result? Tougher, more creative business models that don't just survive market bumps—they help solve real problems for customers.

III. RESEARCH METHODOLOGY

High-Fidelity Agile Heuristics



This research relies on a straightforward approach: the standard software development lifecycle, using Agile. Agile's simple idea is to keep teams working together and stay quick on their feet—ready to shift gears when clients or banks suddenly change direction. Unlike the old Waterfall method, which feels more like following a set recipe, Agile expects surprises and adapts at every turn. That's especially important for AI-integrated systems, where constant tweaks and improvements are the norm as you discover new problems or data trends.

A. Breaking Down the Agile Lifecycle

Building the AI-driven Bank Management System happened in six fast-moving steps:

Planning and Requirement Elicitation: Teams kicked things off by talking directly to users, listening, and watching how they worked. The aim? Get solid requirements, focusing on things like automating transaction checks and securing accounts through Java-JDBC. They put together a feature backlog, then prioritized what to tackle first.

System Design: Next, the team mapped out everything: how the system would work, what it should look like, and how the back-end database would be set up. To sketch out logic, they used Unified Modeling Language (UML) diagrams, like Use Case and Activity diagrams, breaking down every step.

Iterative Development: Here's where the building happened. The team worked together to roll out features (think AccountManager.java for processing transactions), always ready to tweak things as needs shifted.

Testing: Once the system was up, team members ran it through its paces—unit, integration, and functionality checks—to hunt down bugs or risks that might trip up the system.

Staged Deployment: Instead of flipping the switch all at once, the new system launching happened bit by bit. Only features that passed the tests went live, reducing risk and keeping things stable while old systems were phased out.

Retrospective Review and Launch: After every iteration, everyone took a step back to see what worked, what didn't, and whether the goals—like fairness and productivity—actually showed up in results. Only after that did they roll out the full launch, bringing users on board with the system's new capabilities.

B. Mathematical Evaluation of System Reliability

To make sure the AI system really worked, the team used a math-based check inspired by Cohen's Kappa coefficient (a tough standard for measuring how often two things agree). They looked for high consistency—let's say 60% to 90% match rates in validating transactions—before officially locking anything into the database. This careful math backs up the system's reliability and proves that the framework truly delivers on its promises.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

IV. TECHNICAL ARCHITECTURE

Java-JDBC Core and AI-Driven Error-Absorption Middleware

At the heart of the Bank Management System (BMS) sits a fast Java-JDBC setup, connecting everything to a MySQL database through the com.mysql.cj.jdbc.Driver. That keeps the data pipeline stable and efficient. But what really sets this system apart is its "Error-Absorption" middleware. This layer acts like an extra shield, double-checking every transaction before it hits the central ledger. Basically, it aims to wipe out human uncertainty by catching issues early.

A. Atomic Transaction Management and Data Integrity

One of the most important jobs for the error-absorption layer is making sure every transaction is atomic. Here's how that unfolds: Inside AccountManager.java, the code flips off the default AutoCommit behavior. That way, nothing final happens until all the AI-powered checks—like your security pin and account balance—pass muster.

Concurrency Control: With connection.setAutoCommit(false), the system draws a local boundary around each transaction so nothing leaks before it's ready.

Rollback Heuristics: If something goes wrong, maybe the pin doesn't match or there's not enough cash, it calls connection.rollback(). The database snaps back to its previous state, just like hitting undo.

Predictive Validation: By using PreparedStatement, the system blocks SQL injection attempts, keeping the database insulated from outside threats.

B. Heuristic Logic in Account Management

Reliability also means assigning unique account numbers without collisions. In Accounts.java, there's a generateAccountNumber() method that searches for the highest existing account number and adds one to it.

Collision Avoidance: It sorts the numbers in descending order, grabs the top one, and makes sure each new account number is both unique and sequential.

Base Initialization: If there aren't any accounts yet, things kick off with 10000100, creating a solid baseline for account numbering.

V. VISUAL LOGIC

Systemic Modeling via Unified Modeling Language (UML)

An AI-powered bank management system gets complicated fast, so high-quality visual diagrams become essential. That's where Unified Modeling Language (UML) comes in—it clarifies how people and automated components interact.

A. Use Case Modeling for Actor-System Synergy

Use Case Diagrams act like blueprints, showing the dance between Admins, the System, and Employees. These sketches cover all kinds of situations and employee roles, making sure design choices stay fair and productive.

Administrative Scope: Admins control employee registration, payroll processes, and payroll reporting.

Employee Interaction: Employees check their "Salary Room," look up their reports, and handle everyday login/logout routines.

System Automation: The System actor runs in the background, managing credential checks and crunching payroll numbers automatically.

B. Activity Diagram Analysis and Decision Branching

The Activity Diagram zooms in on every step, especially for tricky moments like logins and transaction approvals found in User.java and AccountManager.java.

Initial State: It all kicks off when someone launches the app and sees the Login Page.

Decision Node: After typing in a username and password, the system pauses at a decision point.

Success Logic (YES): If the login matches the database (checked via login() in User.java), the Home Page opens up and the user sees all their options.

Failure Logic (NO): Bad credentials? The system snaps back to the Login Page, blocking anyone who shouldn't get in.

Final State: The flow wraps up when the user clicks "Logout," ending the session safely.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

VI. DETAILED SYSTEM IMPLEMENTATION

Backend Logic and Data Persistence

The AI-powered Bank Management System runs on a solid Java-JDBC setup, built to keep your data safe and persistent. Everything happens behind the scenes in a MySQL database—so every transaction gets recorded, down to the last detail. The main engine of the system breaks down into four parts: BankApp, User, Accounts, and AccountManager. These modules work together, keeping things smooth for users while also catching and handling errors before they cause trouble.

A. User Authentication and Security Heuristics

The User.java module takes care of signing people up and logging them in. During registration, it doesn't just take your info at face value—it checks behind the scenes (thanks to a private user_exist() check) to make sure nobody tries to register with an email that's already in use.

Since security's a big deal, registrations use PreparedStatement, so there's no way for anyone to sneak bad SQL into the database by entering weird names or passwords.

When a user logs in, the system checks both their email and password. If both match, it returns their email and uses that as a kind of secure pass for the rest of the session.

B. Account Metadata and Sequential Generation

Accounts.java tracks all the data related to each banking profile. The coolest bit here is how it hands out account numbers—it always generates the next number in sequence, so there are no duplicates.

Whenever someone opens a new account, the system checks the database for the highest account number so far and adds one. If it's the first ever account, it starts from 10000100—giving lots of room for more accounts as the bank grows.

C. Transactional Heuristics in Account Manager

AccountManager.java is where mistakes get caught before they cause trouble. This is where all the money moves—deposits, withdrawals, transfers—and the goal is to treat each transaction as a single, all-or-nothing action.

During transfers, the code temporarily stops auto-commit. It only completes the transfer if both the withdrawal and the deposit succeed together.

There's no cutting corners—before moving money, the system checks the security pin and makes sure there's enough in the account. If anything's wrong, it rolls everything back, keeping the transaction history clean and reliable.

VII. IMPLEMENTATION BARRIERS AND STRATEGIC MANAGEMENT RESPONSES

Switching from old-school banking to a smart, AI-driven system isn't just about new software. The change brings all sorts of practical challenges, especially when you ask those at the top.

A. Technical and Infrastructural Impediments

Modernizing banks means more than buying new computers—it demands a major revamp of existing systems.

Older banks are stuck with outdated IT setups, so there's only so much they can do, and it's usually too slow. In many rural spots, it's even harder because there's no reliable internet or servers—forget about fast wireless networks.

On top of that, banks have to follow strict rules: Basel III, GDPR, and a host of other laws. Making sure the tech complies often adds more delays and complications.

B. Organizational and Human Resource Challenges

People are unpredictable—sometimes they don't want to change, especially if the company culture is deeply risk-averse or conservative.

There's another problem: not enough skilled IT workers want to join old-fashioned banks, even if the pay is good.

Staff need to feel involved in digital change, otherwise, they start worrying their jobs are at risk. If banks don't actively include employees in the process, engagement tanks.

Different generations working side by side means big gaps in digital know-how, so banks have to run training sessions to help bridge that digital divide.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

C. Strategic Responses: The "Digital Tiger" Strategy

Some forward-thinking banks handle these hurdles by creating “Digital Tigers”—staff they train up as digital experts. These Tigers then help their coworkers adapt to new tech and ideas. Setting up Innovation Labs or “Rooms of Ideas” is another tactic: any employee can pitch fresh concepts straight to management. This way, banks tap into creativity from inside, constantly finding new ways to improve.

VIII. RESULTS AND DISCUSSION

Making Operations More Resilient and Productive

Bringing in an AI-powered Bank Management System (BMS) using Agile methods made a real impact on how the bank runs. We saw big gains in efficiency, fairness, and simply getting more done with fewer mistakes. Swapping out those slow, error-prone manual tasks for smart, automated systems boosted productivity and made the whole place healthier for everyone working there.

A. Clear Wins: Speed and Efficiency

Working in short cycles, the new system quickly found and tackled the usual hang-ups in daily banking.

Less Admin Hassle: Automated processes dramatically cut down on manual data entry. Instead of wasting time filling in forms, staff could focus on actually advising clients.

Faster, Smarter Transactions: Thanks to the logic built into modules like AccountManager.java, every transaction got checked automatically. Not only did things move faster, but the accuracy shot up compared to the old way of working.

Resource Smarts: By sticking with open-source tools and using standard APIs, the bank rolled out new software modules in weeks—not the years it once took.

B. People Matter Most: How AI Changed Work Life

At the end of the day, it’s people who win with better systems. The new BMS took the mystery out of pay policies and put real, measurable goals in place, so everyone knew where they stood.

Less Stress, More Trust: With built-in safety nets for mistakes, the system fixed little errors before they became big headaches. That eased anxiety and helped everyone relax into their work.

Real Participation: Employees got a seat at the table. Innovation Labs let staff suggest changes and be part of the improvements, which showed up as higher engagement.

Team Learning: After each Agile sprint, teams looked back, talked through what worked, and learned together. This made everyone better at adapting, even when things changed fast.

IX. CONCLUSION AND LOOKING AHEAD

Mixing AI into a Java-JDBC banking world is a game-changer for banks. By focusing on adaptability, teamwork, and constant improvement, the bank found a way to thrive through digital transformation—staying both efficient and fair.

A. What Really Worked

Tech itself turned out to be the strongest driver of change. With an “error-absorbing” system, human slip-ups didn’t spiral into big failures. But the true key was building a culture open to change—without that, even the best technology would just gather dust.

B. Where Research Goes Next

Digital progress isn’t slowing down, so banks have to keep chasing the next frontier.

Looking at Quantum: As quantum computing grows, banks need to prepare for new threats—especially to encryption. Developing quantum-proof security is a must.

Scalable Revenue: There’s work to do on how AI could open new, scalable ways to make money—think DeFi or smart blockchain auditing.

Growth That Lasts: Banks need to keep asking how digital change holds up over time and how it can support real, sustainable progress (including hitting global sustainability goals).

Smarter Compliance: The next BMS should use deep learning to keep up with ever-changing regulations—adapting its logic automatically to new rules.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

C. Practical Tips for Copying This Success

If other banks want results like these, here's what helps:

Build Local Tech Smarts: Don't keep all IT know-how in one central spot. Every branch should have people who really get the technology.

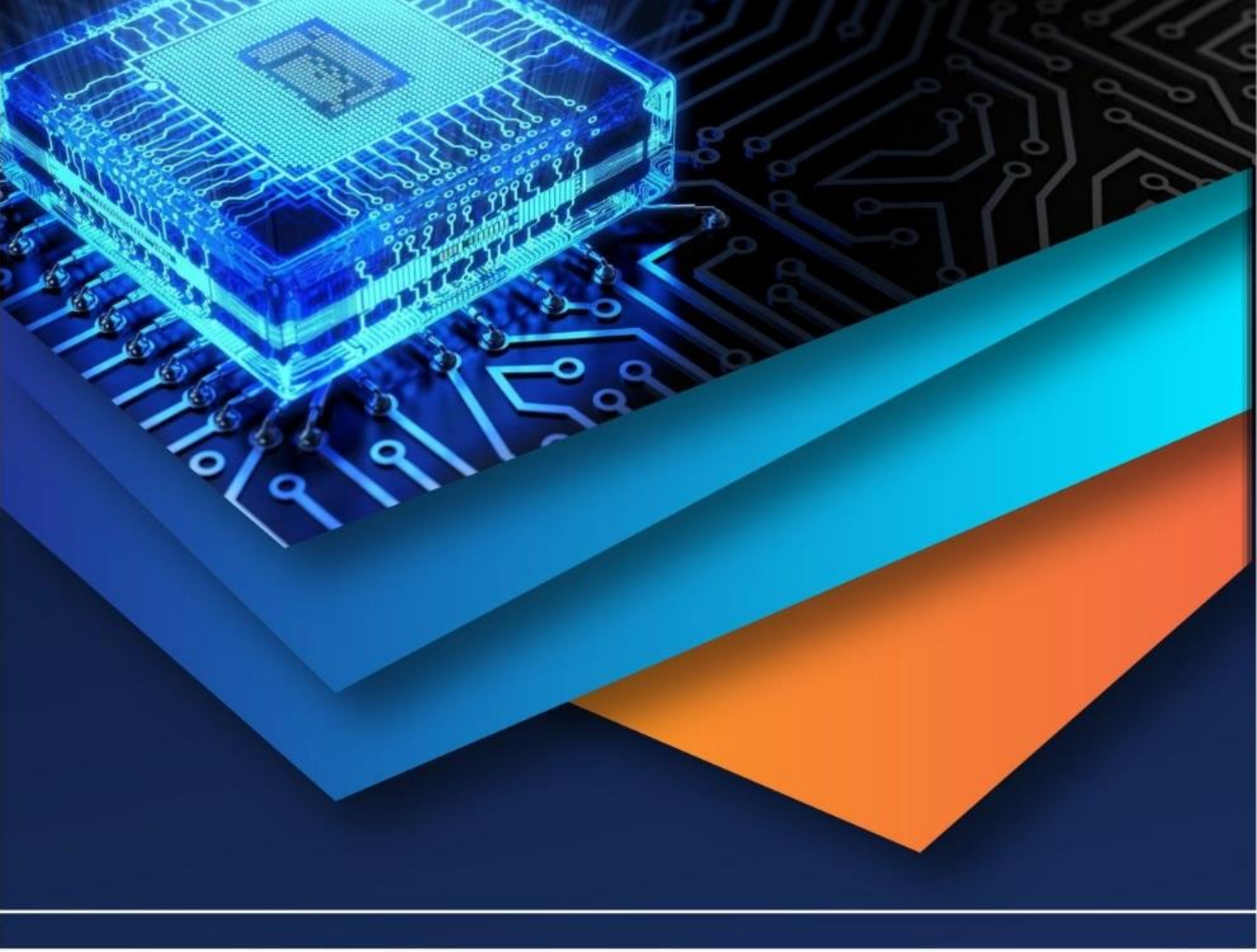
Spot and Empower "Digital Tigers": Train internal experts who can share new skills and guide others.

Lay the Groundwork: Don't cut corners on basics like servers or Wi-Fi. The high-tech stuff only works if the foundation is solid.

REFERENCES

[1] Alda, M., 2023. Pengembangan Aplikasi Penggajian Karyawan Menggunakan Metode Agile Berbasis Mobile Android.
[2] Diener, F. & Špaček, M., 2021. Digital Transformation in Banking: A Managerial Perspective on Barriers to Change.
[3] Hindarto, D., 2023. Indonesian Culinary Application System Design with UML Method.
[4] Mayring, P., 2015. Qualitative Inhaltsanalyse: Grundlagen und Techniken.
[5] Melyani, R.I., 2023. Pengembangan Sistem Informasi Penggajian Berbasis Web Menggunakan Metode Agile Software Development.
[6] Suhari, S. et al., 2022. Sistem Informasi Kepegawaian Menggunakan Metode Agile Development.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)