**VIAHALE: TRANSPORT NETWORK VEHICLE SERVICE**

**ADMINISTRATIVE WITH QR CODE-BASED  
VISITOR REGISTRATION AND FACIAL   
RECOGNITION POWERED BY OPENCV**

A Capstone

Presented to the Faculty of

The College of Computer Studies

Bestlink College of the Philippines

In Partial Fulfilment

Of the Requirements for the Degree of

Bachelor of Science in Information Technology

**GLEN B. HONRADO**

**DANICA C. LUMBAO**

**ERICA MAE S. MADRIGAL**

**LANCE STEVEN A. RAMOS**

**CARL JUSTINE R. SOLIS**

July 2025

# **DECLARATION**

# **APPROVAL SHEET**

# **CERTIFICATE OF ORIGINALITY**

# **ACKNOWLEDGEMENT**

# **DEDICATION**

# **ABSTRACT**

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# **TABLE OF CONTENTS**

[**DECLARATION**](#_wwl1diyc37yz) **ⅰ**

[**APPROVAL SHEET**](#_4czc4fde03qu) **ⅱ**

[**CERTIFICATE OF ORIGINALITY**](#_11v8nqbk0jvz) **ⅲ**

[**ACKNOWLEDGEMENT**](#_3ql783ho3c9g) **ⅳ**

[**DEDICATION**](#_d0tdyy3naoi0) **ⅴ**

[**ABSTRACT**](#_hdbj9ee4mm88) **ⅵ**

[**TABLE OF CONTENTS**](#_k0fvgkjz8seo) **ⅶ**

[**LIST OF TABLES**](#_qxfie9q3su8c) **ⅷ**

[**LIST OF FIGURES**](#_begzg432zwpb) **ⅸ**

[**Chapter Ⅰ**](#_yjnkt24crmtl) **ⅹ**

[**INTRODUCTION 1**](#_cc7ade4o8usv)

[1.1 BACKGROUND OF THE CAPSTONE PROJECT 1](#_kzifbayp9m5y)

[1.2 CONTEXT AND SCOPE 3](#_of9hdcqmj513)

[1.3 PROBLEM STATEMENT 4](#_dowht4ahfkwl)

[1.4 GOALS AND OBJECTIVES 5](#_q50hy0rzma5m)

[1.5 SIGNIFICANCE OF THE STUDY 7](#_97jyyif8neip)

[1.6 Document Structure 8](#_ib7mrgg371dd)

[2.1 OVERVIEW OF THE AGILE SCRUM METHODOLOGY 9](#_n9szq49zuip)

[2.2 CONCEPTS OF ENTERPRISE ARCHITECTURE (EA) 13](#_x19dvec2j592)

[2.3 Microservices Architecture 19](#_2rvtde8yjdlz)

[2.4 DevOps and CI/CD 21](#_308ad2p2l8hc)

[2.5 RELEVANT STUDIES AND RESEARCH 23](#_eqg547lns4az)

[2.6 INTEGRATION OF INFORMATION SYSTEMS 25](#_67usbn8khn70)

[**METHODOLOGY 32**](#_hp6ig7l7avsq)

[3.1 AGILE SCRUM METHODOLOGY IN THE PROJECT 32](#_62ul45jmavna)

[3.2 ROLES 36](#_s6upxoiaz0ki)

[3.3 Sprint Cycles 37](#_bsz8x2snoqxe)

[3.4 SCRUM ARTIFACTS 38](#_1x2dgu8y4plr)

[3.5 Microservices Architecture 39](#_da59i6kmm0hx)

[3.6 DevOps Implementation 44](#_azx6ui3fnizj)

[3.7 Integration Approach for Information System 45](#_jkwwfkenkgr4)

[3.8 Introduction to TOGAF and the Four Architectural Domains 45](#_qp5n6ohohvl5)

[**Requirement Analysis 48**](#_ityk5771b2a9)

[4.1 Stakeholder Identification 48](#_rgwuhdip6lxh)

[4.2 Requirements Gathering Techniques 48](#_r1kxxieew896)

[4.3 User Stories and Use Cases 48](#_ifiygsqqsy43)

[4.4 Functional Requirements for Integration 57](#_p0ev9wtk11mb)

[**CHAPTER Ⅱ 65**](#_a0e2snbi1960)

[**BUSINESS PROCESS ARCHITECTURE 65**](#_r2mxgtwfpo91)

[5.1 Identification of Business Processes 65](#_kcs7jxjy43vc)

[5.2 Business Process Diagrams 67](#_8ch43wdcna53)

[5.3 Alignment of Integrated System with Business Processes 67](#_1hs3czlutrj0)

[5.4 Business Process Improvements 67](#_jnjs5k09obb)

[**APPLICATION ARCHITECTURE 71**](#_aig6zo7fc6zp)

[6.2 Application Architecture Diagrams 72](#_xb62msudxvm4)

[6.3 Integration of Software Modules 73](#_t8yay5kktma2)

[6.4 Communication and Interaction Patterns 74](#_1kb2w2v3r9bu)

[**DATA ARCHITECTURE 75**](#_fs8m1q4vnghl)

[7.1 Data Sources and Types 75](#_dvbsctt00uq5)

[7.2 Data Flow Diagrams 75](#_a83heyd2vwkz)

[7.3 Data Storage and Management 77](#_ptyf028p7b35)

[7.4 Data Synchronization Across System 78](#_g99u1dxif7db)

[**TECHNOLOGY ARCHITECTURE 78**](#_8dsofzpsvwtq)

[8.1 Technology Stack and Infrastructure 78](#_ku5l7ryu8zsp)

[8.2 Software Technologies 82](#_tqcqytk392mu)

[8.3 Scalability and Performance Considerations 84](#_wxepihnnhu8o)

[**CHAPTER Ⅲ 86**](#_bggh3lh6vpma)

[**DEVELOPMENT PROCESS 86**](#_6htdamaqrw72)

[9.1 Agile Scrum Roles and Responsibilities 86](#_xfouq0ji0352)

[9.2 Sprint Planning and Backlog Management 87](#_an2b0u162ql8)

[9.3 Sprint Execution and Deliverables 87](#_h6icddckkrb8)

[9.4 Challenges Faced in the Development Process 87](#_a6ctv4xquh05)

[**References 88**](#_psv9annmij8q)

# 

# **LIST OF TABLES**

# 

# 

# 

# 

# 

# 

# 

# 

# **LIST OF FIGURES**

# 

# 

# 

# **CHAPTER Ⅰ**

# **1. INTRODUCTION**

## **1.1 BACKGROUND OF THE CAPSTONE PROJECT**

In the past few years, Transport Network Vehicle Service has rapidly grown in the Philippines, and a lot of commuters rely on ride-hailing platforms like Grab, Uber, and Angkas for convenience and accessibility. These services reduce dependencies on traditional taxis and public utility vehicles. In this transformation the administrative areas of TNVS operations have grown increasingly complex such as Facility Reservation, Visitor Management, and Data monitoring.

Administrative areas of TNVS companies are frequented by a diverse mix of visitors: drivers, applicants, delivery personnel, clients, and business partners. Managing this growth securely and efficiently is vital not only for facility protection but also for maintaining a smooth human resource and compliance workflows. Many of these offices still rely on manual logbooks and basic id checks. The traditional manual logbooks for physical management systems used in TNVS offices have shown limitations, from delays in check in, inaccurate tracking, unauthorized access to inefficient record keeping. With the increased demand for security, accountability, and real-time data in administrative areas, there's a drawing need for smart solutions.

In contrast, institutions such as local government units and corporate buildings have already adopted modern visitor tracking systems integrating QR code scanners and Facial Recognition. These technologies offer fast, contactless, and verifiable ways of managing people entering and exiting a facility. However, these advancements remaina noticeable gap in customized specifically for the administrative operations of TNVS providers where real-time monitoring, mobility, and data integration are essential.

This capstone project, titled ViaHale: Transport Network Vehicle Service Administrative with QR Code-Based Visitor Registration and Facial Recognition Powered by OpenCV, aims to address these gaps. It proposes a smart, centralized, and automated platform tailored to TNVS administrative functions. By integrating QR code check in, facial recognition, and real time visitor monitoring, the system will modernize the visitor entry process while improving data accuracy, operational visibility, and overall security for TNVS facilities.

## **1.2 CONTEXT AND SCOPE**

This project is set in the context of the growing Transport Network Vehicle Services (TNVS) industry in the Philippines. As more people rely on apps like Grab and Angkas for daily transport, the internal operations of TNVS companies especially in handling visitors are also getting more demanding. Offices often receive walk-ins from drivers, job applicants, partners, and other guests, making visitor tracking an essential part of day-to-day operations.

At present, many companies still use manual logbooks for visitor entry. This method can be slow, disorganized, and prone to errors, especially during busy hours. It’s hard to keep track of who's coming in and out, and this can affect both security and efficiency in the workplace.

This project aims to develop ViaHale, a smart administrative system focused on visitor monitoring. The main goal is to make the check-in process faster and more organized through the use of Facial Recognition and QR codes. Instead of writing on paper, visitors can check in quickly using a contactless method. The system will also let admins monitor visitor flow in real time and access logs whenever needed, all in one place.

The system is focused solely on visitor management. It does not cover ride-booking, driver or fleet tracking, customer services, or mobile app functions. The focus is on improving the internal check-in process in TNVS offices through a smart, efficient, and easy-to-use system.

## **1.3 PROBLEM STATEMENT**

The current administrative processes in Transport Network Vehicle Services (TNVS) suffer from inefficient manual check-ins, unsecured identity verification, and disconnected booking and visitor data systems resulting in security vulnerabilities, delayed operations, and poor coordination that affect staff, drivers, and clients alike.

**Why This Problem Matters:**

1. Manual check-in inefficiency slows down daily operations, especially during peak hours, causing frustration among visitors and administrative staff.
2. Security gaps in identity verification, due to lack of biometric or digital validation, allow risks such as unauthorized access and identity fraud a serious concern for safety and compliance.
3. Limited real-time oversight makes it difficult for administrators to monitor visitor flow, facility usage, and legal documentation accurately.
4. Disjointed systems for bookings and visitor management result in miscommunication between departments, scheduling conflicts, and delays in coordination.

According to recent studies, over 70% of administrative departments in service industries report significant inefficiencies due to fragmented data systems, while security breaches linked to poor identity verification cost organizations millions annually in operational and reputational damage.

This project seeks to resolve these challenges by creating an AI-powered TNVS administrative system that integrates smart visitor monitoring using Facial Recognition and QR check-in, streamlines document and legal management, enables facility reservation automation, and provides administrators with a real-time dashboard to optimize oversight and decision-making across operations.

## **1.4 GOALS AND OBJECTIVES**

**The primary objectives and goals of ViaHale: Transport Network Vehicle Service Administrative System project are as follows:**

1. Develop a centralized TNVS administrative platform that integrates visitor management, facility reservations, legal case tracking, and document management to streamline internal operations.
2. Implement a smart visitor monitoring system utilizing QR code check-in/out, and facial recognition authentication to enhance security and efficiency in guest and personnel identification.
3. Design an administrative dashboard with real-time analytics and data visualization for improved decision-making and operational oversight.
4. Ensure data privacy and security compliance by applying secure protocols for managing personal, and legal data in accordance with the Data Privacy Act.
5. Provide comprehensive technical documentation and user manuals to support system deployment, onboarding, and long-term maintenance.
6. Gather feedback from administrative staff and stakeholders through surveys and testing to refine system functionality and improve user experience.

## **1.5 SIGNIFICANCE OF THE STUDY**

With Transport Network Vehicle Service (TNVS) such as Grab and Angkas booming in the Philippines, administrative offices are facing increased visitor traffic drivers, applicants, partners, and more while still relying on manual logbooks that slow down operations and pose security risks. This project is important because it modernizes this administrative process by using Facial Recognition, QR codes, and real-time monitoring, resulting in faster check-ins, safer access control, and more accurate records. This transformation aligns with industry-wide shifts toward contactless, automated solutions. For instance, a 2022 McKinsey study found that organizations integrating automation into workflows can boost productivity by up to 30%. Moreover, robust identity verification measures like multifactor authentication (MFA) can reduce unauthorized access by over 99%, enhancing security for admin systems. By addressing real-world concerns from operational efficiency to facility security, This project benefits administrative staff, companies, visitors, security personnel, and future researchers.

Who benefits from the project:

* + - 1. Admin Staff – Less paperwork and faster check-in process
      2. Visitors – A quicker and smoother experience
      3. Companies – Better security and accurate visitor logs
      4. Security Teams – Easier monitoring of who enters and leaves
      5. Future Researchers – A helpful reference for building smart admin systems

In today’s world, where digital and contactless systems are becoming the norm, this project is relevant and timely.

## **1.6 Document Structure**

This document is organized into several key sections that guide the reader through the development of the ViaHale: Transport Network Vehicle Service Administrative System with Qr-Based Visitor Registration and Facial Recognition. It begins with the introduction and background of the study, followed by a review of related literature and research that supports the project. The methodology section explains the development process and tools used. The design and implementation section showcases the system’s features, including QR code-based visitor registration and facial recognition. Finally, the document concludes with findings, evaluation, and recommendations. Together, these sections provide a complete understanding of the system's purpose, development, and potential impact on administrative operations.

# **CHAPTER II**

1. **LITERATURE REVIEW**

## **2.1 OVERVIEW OF THE AGILE SCRUM METHODOLOGY**

Agile Scrum is a framework within the Agile methodology that is widely used in software development projects. It emphasizes iterative and incremental development, allowing teams to deliver high-quality software while adapting to changing requirements. Scrum promotes collaboration, flexibility, and customer involvement throughout the development process.

**Principles of Agile Scrum**

Delivering valuable software to customers early and continuously is a primary goal of agile methodology, as emphasized by Schwaber and Sutherland (2020) in *The Scrum Guide*. This principle is particularly relevant for VIAHALE: Transport Network Vehicle Service Administrative system, which aims to enhance transportation network services in administrative areas through innovative technology. Daily collaboration between business stakeholders and developers is essential; for system development, this means engaging with transportation authorities, users, and technical teams to ensure that the QR code-based visitor registration and facial recognition features meet real-world needs. The most effective communication is face-to-face conversation, which can facilitate quick feedback and adjustments to the system.

Agile Scrum welcomes changing requirements, even late in development, allowing VIAHALE: VIAHALE: Transport Network Vehicle Service Administrative system to adapt to new regulations or user feedback regarding visitor registration processes. This flexibility provides a competitive advantage in the rapidly evolving transportation sector. As noted in *The Scrum Guide*, agile processes promote sustainable development, maintaining a constant pace indefinitely, which is crucial for the ongoing updates and maintenance of the system.

Working software is delivered frequently, with a preference for shorter timescales; this approach allows development to roll out features incrementally, such as initial QR code functionalities followed by advanced facial recognition capabilities. Continuous attention to technical excellence and good design enhances agility, ensuring that the VIAHALE: Transport Network Vehicle Service Administrative system remains robust and user-friendly. The primary measure of progress is working software, which for VIAHALE translates to the successful deployment of its features that improve the efficiency and security of transportation services (Schwaber & Sutherland, 2020).

**Practices of Agile Scrum**

According to Haigh, A.J. (2025) Agile Scrum methodology development is structured around time-boxed iterations known as sprints, typically lasting 2-4 weeks. For the VIAHALE Transport Network Vehicle Service Administrative system, each sprint represents a critical opportunity to advance the development of its QR code-based visitor registration and facial recognition system. At the beginning of each sprint, the team collaborates to plan the specific features and enhancements to be completed, ensuring alignment with the project’s goals and user needs.

Short daily meetings, often referred to as stand-ups, are essential for maintaining momentum within the team. During these meetings, team members discuss their progress, outline plans for the day, and identify any obstacles they may be facing. This practice fosters open communication and quick problem-solving, which is vital for addressing challenges that may arise in the development of complex features like facial recognition technology.

At the end of each sprint, the team demonstrates the completed work to stakeholders and end-users, to gather valuable feedback. This iterative review process allows the team to ensure that the system is evolving in a way that meets real-world requirements and expectations, enabling timely adjustments based on user input.

The **Product Backlog** serves as a prioritized list of features, enhancements, and bug fixes that the team will work on. This dynamic list is continuously refined based on stakeholder feedback and changing requirements, ensuring that the most critical functionalities such as the QR code registration process and facial recognition capabilities are prioritized for development. By maintaining a clear and prioritized backlog, the team can effectively manage its resources and focus on delivering the most valuable features to enhance administrative areas.

**Iterative and Incremental Nature**

Agile Scrum is characterized by its iterative and incremental approach. Teams work in cycles (sprints) to refine and improve the product based on feedback and changing requirements. Each sprint results in a potentially shippable product increment, allowing for gradual enhancement of the software.

According to Magnussen, E. F., Havnelid, E., & Molléri, J. S. (2024) customer collaboration is a cornerstone of the Agile Scrum framework. The Product Owner plays a crucial role in representing stakeholders, defining the product vision, and prioritizing the backlog to ensure that the development team aligns with user needs. Regular feedback from customers is essential, as it allows the team to adjust priorities and refine features, such as the QR code registration and facial recognition functionalities, to better serve users. Additionally, Scrum's adaptability to changing requirements is particularly relevant in this project. The iterative nature of the framework enables the team to reassess priorities and make necessary adjustments based on stakeholder feedback, market dynamics, or new insights gained during development. This flexibility ensures that the project remains responsive to user needs and can effectively integrate advancements in technology, ultimately enhancing the overall user experience.

## **2.2 CONCEPTS OF ENTERPRISE ARCHITECTURE (EA)**

In the context of the ViaHale administrative system, Enterprise Architecture (EA) plays a vital role in aligning the project's technological components with the operational goals of the TNVS platform. By applying EA principles, the system ensures that the four key administrative modules such as, Facilities Reservation, Document Management, Legal Management, and Visitor Management, are developed in a unified and scalable structure that supports overall service efficiency. This structured alignment supports digital transformation and efficient governance Ettinger, A. (2025).

**Enterprise Architecture’s Role in Aligning Business and Technology**

In developing the ViaHale TNVS administrative system, Enterprise Architecture ensures that the system's technological components are aligned with the platform's operational structure and workflows. Each administrative function has been designed to complement one another through well-coordinated architecture. The connection between the Document and Legal Management modules, for instance, enables streamlined access to case-related documents, minimizing manual handling, and ensuring proper documentation flow across departments. This structured alignment across modules supports greater efficiency, reduces duplication, and promotes a more responsive and organized administrative environment within the TVS system.

Moreover, EA supports long-term sustainability by creating a scalable and adaptable system structure. It defines how each component interacts through well-planned data flows, application design, and supporting technology infrastructure. According to O’Higgins, D. (2023), effective business architecture significantly influences an organization's digital transformation by improving coordination, reducing complexity, and enabling strategic alignment between IT and operational goals. This directly applies to ViaHale, where every design decision from database schemas to user interface functionality contributes to broader institutional objectives. EA thus ensures that ViaHale’s administrative system is not only efficient in its current state but also future-ready, capable of integrating innovations such as biometric verification or AI-driven analytics without the need for major overhauls. Ultimately, EA ensures that technology serves as a driver of transparency, innovation, and operational excellence, reinforcing the TNVS system’s ability to evolve with emerging demands while maintaining structural integrity.

**Key Enterprise Architecture’s (EA) Frameworks and Standards**

The Key EA frameworks, such as TOGAF, offer standardized methodologies for developing and managing enterprise architectures. TOGAF's Architecture Development Method (ADM) provides a structured approach for designing, implementing, and maintaining enterprise systems that support both current and future business needs. These frameworks are highly relevant to project architecture as they ensure consistency, reduce redundancy, and support decision-making across the enterprise.

According to van de Wetering (2021), a critical advantage of using standardized Enterprise Architecture (EA) frameworks is their alignment with essential architectural elements governance, stakeholder engagement, and technology standardization. Van de Wetering’s quantitative study involving 299 CIOs, IT managers, and lead architects found that dynamic EA capabilities (realized through frameworks like TOGAF) significantly enhance business IT alignment and process innovation, delivering tangible organizational benefits. This supports the argument that adopting TOGAF in complex systems like ViaHale not only improves operational alignment but also positions the enterprise for sustainable growth and innovation.

**Areas of Architecture within Enterprise Architecture (EA)**

**Business Architecture** in the ViaHale system models operational flows such as Facility Reservation, Visitor Check-In, and Document Handling. These processes align with institutional policies and strategic goals, ensuring functional coordination across administrative modules Bernard, S. A. (2020).

**Application Architecture** includes the QR code scanning module, facial recognition system, and document tracking software. These applications function cohesively to streamline workflows and improve system usability. Rehman et al. (2023) note the growing importance of intelligent components within enterprise systems to support dynamic user needs.

**Data Architecture** supports structured data handling across the modules such as secure Visitor logs, audit trails for legal case documentation, and metadata tagging for digital files ensuring traceability and compliance Ettinger, A. (2025).

**Technology Architecture** integrates tools such as OpenCV for facial recognition, cloud-based storage for centralized access, secure APIs for inter-module communication, and responsive mobile/web platforms. Gartner Research (2021) emphasizes that such integrations reduce fragmentation, enhance scalability, and support enterprise-wide agility.

**Importance of Coherent Architecture**

A coherent Enterprise Architecture is essential in ensuring that the ViaHale TNVS administrative system functions as a fully integrated, scalable, and future-ready platform. Given the nature of ViaHale's operations which involve real-time facility bookings, secure handling of legal documents, and cross-module coordination among administrative departments, a unified architectural structure is critical Sararuch, S., Palaiahnakote, P., & Kim, K. (2023). This structure prevents the development of isolated modules and supports interoperability, future enhancements, and cost-efficiency.

This unified structure supports interoperability, allowing smooth data flow and consistent communication between system modules. For instance, legal documents stored in the Document Management module can be automatically referenced in the Legal Management module without duplication or manual transfer. Additionally, a coherent architecture positions ViaHale for future enhancements, such as the integration of biometric verification or Al-driven analytics, without requiring major system overhauls. By minimizing complexity, eliminating redundancies, and reducing maintenance costs, the system's coherent architecture directly contributes to operational efficiency and supports the organization's long-term digital transformation goals.

## **2.3 MICROSERVICES ARCHITECTURE**

Microservice is an innovative architectural style that is becoming more and more popular. Numerous scholars support microservices as a solution to the outdated methods for monolithic architecture style. However, there is no comprehensive research study on the shift from monolithic architecture to MSA. Micro Service Architecture has made it necessary to create software development processes and architecture design is the continuous development of MSA to meet the needs of autonomous deployment, scalability, and maintenance Razzaq et al. (2022).

A study by Bushon et al. (2021) Microservices Architecture enhances scalability, reusability, and independent deployment essential features for administrative systems where several workflows must run simultaneously and independently.

Calp and Karakose (2023) highlighted the growing popularity of AI-assisted service deconstruction and dynamic orchestration in a recent review on artificial intelligence based microservice architecture. This is particularly useful in complex administrative systems where processes such as legal compliance and document approval can be modularized and automated.

Microservices have also been found to be effective in edge computing environments, where real-time visitor data needs to be processed efficiently. Gharavi and Hu (2023) investigated this in the context of distributed this in the context of distributed microservices across cloud and edge, highlighting latency reduction and improved user responsiveness key factors in Visitor Management systems.

Zia et al. (2024) showed the benefits of Microservice Architecture in building responsive and scalable reservation systems. The framework allowed real-time analytics and predictive processing, improving both administrative planning and user experience, an ideal fit for the Facility Reservation module.

## **2.4 DEVOPS AND CI/CD**

DevOps is a modern approach that bridges development and operations by integrating people, processes, and technology to streamline the application lifecycle. It breaks down silos between development, IT operations, quality assurance, and security teams, fostering continuous collaboration and faster delivery cycles (Chintale, 2024). This is especially critical in administrative systems like TNVS, where rapid updates, consistent performance, and secure operations are paramount.

Continuous Integration (CI) and Continuous Delivery (CD) are core DevOps practices that automate testing and deployment workflows, enabling frequent, reliable releases. Soares et al. (2021) analyzed 101 empirical studies and found that CI enhances system integration, team collaboration, and code quality—benefits directly applicable to complex admin systems that integrate visitor management, driver profiling, reservation modules, and real-time access logs.

Rongali (2025) emphasized that CI/CD pipelines improve operational efficiency and scalability in AI workflows—principles that can be extended to TNVS systems that may later integrate predictive analytics (e.g., for driver performance or facility usage). Automation of critical tasks such as version control, database migrations, and security patching through CI/CD pipelines ensures that releases are not only faster but more reliable, minimizing downtime and manual errors.

As administrative systems scale, CI/CD pipelines play a key role in maintaining consistency across environments—development, staging, and production. These pipelines ensure that all updates to the TNVS system, whether a new module or a hotfix, are deployed uniformly using version-controlled rollouts that support traceability and rollback if needed. This stability is crucial for systems that deal with sensitive data such as visitor logs and driver profiles.

Moreover, successful DevOps implementation is not purely technical—it demands cultural and organizational readiness. Gwangwadza and Hanslo (2022) identified collaboration, toolchain integration, and continuous improvement as critical success factors. In line with this, Al-Debagy and Martinek (2023) outlined nearly 100 success metrics, reinforcing that effective DevOps adoption requires structured processes and team alignment—both of which are essential when building a TNVS admin platform with cross-functional development teams.

In the context of the TNVS administrative system, adopting DevOps practices ensures faster release cycles, better testing coverage, and the ability to respond rapidly to user feedback and operational requirements. Whether rolling out features like real-time check-in dashboards, QR/RFID-based access, or analytics reporting, CI/CD pipelines provide the foundation for scalable and resilient system deployment.

## **2.5 RELEVANT STUDIES AND RESEARCH**

This section presents a review of previous studies, scholarly articles, and research related to QR code-based systems, facial recognition, and administrative technologies. It aims to summarize key findings, methodologies, and conclusions that are relevant to the proposed capstone project. By analyzing these studies, this section highlights best practices, identifies existing gaps, and explores opportunities for innovation. The insights gathered provide a solid foundation for the project and ensure it builds upon validated research and real-world applications.

**RELATED STUDIES AND RESEARCH**

**Gallera (2023)** authored *"Designing and Evaluating a QR Code-Based Monitoring System for School Visitor Logs"*, where a school portal and mobile app allowed visitors to check in via QR codes, with positive usability, security, and maintainability results.

**Nguyen‑Tat et al. (2024)** authored *"Automating Attendance Management in Human Resources: A Design Science Approach Using Computer Vision and Facial Recognition"*, where facial recognition powered by OpenCV and Jetson Nano was used to automate attendance in workplace settings, resulting in high accuracy and cost-efficiency, particularly for organizations with limited infrastructure.

**Bugingo et al. (2025)** authored *"Enhancing Face Recognition Attendance System Utilizing Real-Time Face Tracking"*, where OpenCV-based facial tracking technology was implemented in an attendance system, significantly improving recognition speed and reliability in real-time environments such as schools and small offices.

**Bhat, Nithin & Pranav (2023)** authored *"Enhancing Room Security and Automating Class Attendance Using ID Cards"*, where the integration of facial recognition, Arduino, and IoT components improved automation in classrooms, demonstrating real-time monitoring and access control relevant to visitor check-ins.

**Face2QR Authors (2024)** authored *"Face2QR: A Unified Framework for Aesthetic, Face-Preserving, and Scannable QR Code Generation"*, where a novel approach was introduced to embed facial images into QR codes using Stable Diffusion, enabling visual recognition while preserving scannability, with implications for enhanced visitor identity verification systems.

## **2.6 INTEGRATION OF INFORMATION SYSTEMS**

Integrating information systems into the business environment is a critical endeavor for administrative areas, which seeks to enhance its operational capabilities. This involves ensuring that various information systems, such as QR code-based visitor registration systems, facial recognition technology powered by OpenCV, and backend databases, work together seamlessly. This integration can improve efficiency, consistency, and decision-making, benefiting both transportation authorities and visitors. By combining these systems, VIAHALE: Transport Network Vehicle Service Administrative System with Qr-Code Based Visitor Registration and Facial Recognition for access control security powered by OpenCV can create a single platform that makes visitor registration easier and improves security. Challenges include data compatibility, tracking system updates, and security issues. However, successful integration can lead to real-time data sharing, accurate visitor identification, and improved reporting capabilities. Common integration tools like Application Programming Interfaces (APIs) and middleware solutions are essential for building a strong infrastructure that can handle changing transportation network needs. These integration efforts align with Enterprise Architecture (EA) ideas, ensuring that VIAHALE's information systems can adapt to future technology and rules changes. Kohansal, M. A. (2024).

**Challenges of Integrating Information System**

Complexity of Systems*:* VIAHALE Administrative system may need to operate with a mix of legacy systems and modern applications, which can lead to compatibility issues. McKinsey & Company. (2021). Integrating these diverse systems such as existing visitor management systems with the new QR code registration and facial recognition technologies can be technically challenging and resource-intensive. This complexity requires careful planning and execution to ensure a smooth integration process.   
*Data Silos:* Different departments within the transportation network may utilize separate systems that do not communicate with each other, resulting in data silos. For the VIAHALE Administrative system, this fragmentation can hinder collaboration among stakeholders, such as transportation authorities and security personnel, and lead to inconsistent data regarding visitor registrations and identifications. Breaking down these silos is essential for achieving a unified and efficient operational framework. Kusumawati, R. (2025).  
*Resistance to Change:* Employees and stakeholders may be accustomed to existing processes and systems, making them resistant to adopting new integrated solutions. Ramadhani, W., Khuzaini, K., & Shaddiq, S. (2024). For the VIAHALE Administrative system, effective change management strategies are crucial to address this resistance, ensuring that all users understand the benefits of the new QR code-based visitor registration and facial recognition system and are trained to use it effectively.

*Security Risks:* Integrating systems can expose vulnerabilities, especially when sensitive data, such as personal identification information, is involved. Herath, H. M. S. S., Herath, H. M. K. K. M. B., Madhusanka, B. G. D. A., & Guruge, L. G. P. K. (2024) The VIAHALE Administrative system must implement robust security measures to protect against data breaches and ensure compliance with privacy regulations. This includes securing data transmission between the QR code system and facial recognition software, as well as safeguarding stored data.   
*Interoperability Issues:* Different systems may use varying data formats and protocols, complicating the integration process. Sadeghi, M., Carenini, A., Corcho, O., Rossi, M., Santoro, R., & Vogelsang, A. (2024). For the VIAHALE Administrative system, ensuring interoperability between the QR code registration system, facial recognition technology, and existing databases will require additional development work. This may involve creating APIs or middleware solutions to facilitate seamless communication between systems.

**Benefits of Integrating Information Systems**

Integrating information systems offers numerous advantages that can significantly enhance operational efficiency and effectiveness. According to the Striped Giraffe Team (2025), these benefits include improved efficiency, data consistency, better decision-making, enhanced collaboration, and scalability. *Improved Efficiency:* Integration automates the data flow between systems, significantly reducing manual data entry and minimizing errors. For this project, this streamlining of processes means that visitor registrations can be processed more quickly and accurately, leading to faster operations and lower operational costs. The automation of tasks such as data capture from QR codes and facial recognition enhances overall service delivery in the transportation network. *Data Consistency:* A unified information system ensures that all departments within VIAHALE Transport Network Vehicle Service access the same data, reducing discrepancies and improving the reliability of information. This consistency is crucial for accurate reporting and analysis, particularly when it comes to tracking visitor patterns and security metrics. Reliable data supports better decision-making and enhances the effectiveness of the transportation network.   
*Better Decision Making:* Integrated systems provide decision-makers with real-time access to comprehensive data and analytics. For the VIAHALE Administrative system, this means that transportation authorities can quickly assess visitor traffic, identify trends, and make informed decisions regarding resource allocation and operational adjustments. The ability to analyze data from both the QR code registration and facial recognition systems enables more effective management of transportation services. *Enhanced Collaboration:* Integration fosters collaboration among departments by providing a shared platform for communication and data sharing. This project leads to improved teamwork among security personnel, administrative staff, and transportation authorities, ultimately resulting in better project outcomes. A collaborative environment enhances the ability to respond to visitor needs and operational challenges effectively. *Scalability:* Integrated systems can be more easily scaled to accommodate growth, as they provide a cohesive framework for adding new functionalities or systems without disrupting existing operations. For this project, this means that as the transportation network expands or as new technologies emerge, the system can adapt and integrate these advancements seamlessly, ensuring continued efficiency and effectiveness.

**Integration Techniques and Technologies**

According to Haider K. (2025), data integration encompasses various techniques that facilitate the seamless exchange and management of data across different systems.Middleware Solutions: Middleware serves as a communication bridge between different components of the VIAHALE system. For example, it connects the visitor registration module with the facial recognition engine powered by OpenCV, ensuring real time data flow and responsiveness.  
API Integration: The use of Application Programming Interfaces (APIs) is crucial for allowing the QR code scanner, facial recognition software, and administrative dashboard to interact efficiently. APIs ensure that data captured from visitors such as check-in timestamps and identity matches can be shared instantly across the system modules. Data Warehousing: VIAHALE benefits from a centralized data repository where all administrative logs, visitor records, facial recognition matches, and system usage analytics are stored. This centralized structure ensures consistent and accessible reporting for security audits and administrative decision-making. ETL Processes: Extract, Transform, Load (ETL) procedures are implemented to gather data from facial recognition results, visitor logs, and QR scan events. These processes clean and standardize the data, making it reliable for further analysis and enhancing the accuracy of security monitoring.Cloud Integration: VIAHALE may leverage cloud integration to ensure that visitor data, facial recognition logs, and administrative reports are securely stored and accessible remotely. This setup allows the system to operate in a hybrid environment where real-time visitor verification can occur on-site while data management and analytics can be handled off-site through cloud services.

# **CHAPTER III**

# **METHODOLOGY**

## **3.1 AGILE SCRUM METHODOLOGY IN THE PROJECT**

In line with the project's goals of flexibility, rapid adaptation, and stakeholder-driven development, researchers have chosen to implement the Agile Scrum methodology. Agile Scrum is a collaborative and iterative framework that emphasizes regular communication, teamwork, and responsiveness to change. This methodology is particularly well-suited for software development projects in higher education, which require close coordination between multiple functional modules and emerging technologies such as QR code-based check-ins and facial recognition powered by OpenCV.

\

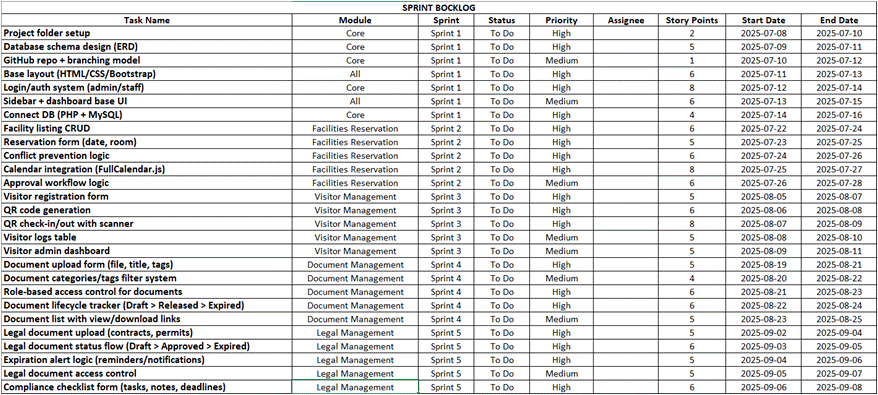
*Figure 1. System Development Life Cycle Diagram of Transport Network Vehicle Service Administrative System*

## **3.2 ROLES**

The project team is structured according to Scrum best practices:

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Responsibility** |
|  | Product Owner | Defines requirements, prioritizes features in the product backlog, and ensures the team delivers value. |
| Honrado, Glen | Scrum Master | Ensures Scrum process is followed, facilitates events, coaches team, and removes impediments. |
| Ramos, Lance Steven  Honrado, Glen | Development Team | Responsible for designing, coding, testing, and delivering product features collaboratively. |
|  | Document Specialist | Creates, maintains, and manages project documentation, ensuring accuracy and accessibility for communication |
|  | Quality Assurance (QA) | Ensures product meets quality standards, develops and executes test plans, and verifies feature functionality. |

## **3.3 SPRINT CYCLES**

The Figure 2. show Sprint Backlog for a project system development. A list of tasks needs to be accomplished by submodules (like Facilities Reservation, Visitor Management, Legal Management, And Document Management) and scheduled across five sprints. Each task includes its priority, estimated effort (story points), and target dates.

*Figure 2. Sprint Backlog: A planning document used in Agile project management to track tasks across different sprints.*

## **3.4 SCRUM ARTIFACTS**

The Scrum artifacts were crucial in directing the development process and guaranteeing alignment with stakeholder needs in the study of VIAHALE: Transport Network Vehicle Service Administrative with QR Code-Based Visitor Registration and Facial Recognition Powered by OpenCV. All necessary features, enhancements, and technical tasks to the project were documented in the Product Backlog, which functioned as a dynamic document. This backlog was updated and improved on a regular basis by the Product Owner in response to feedback from stakeholders and users. This allowed them to prioritize important tasks like integrating facial recognition and QR code registration. The team was able to stay focused on providing value and adjusting to any new needs or insights to this flexible strategy.

Additionally, the team's work was organized according to the Sprint Backlog for each sprint. By selecting specific items from the Product Backlog. The Development Team pledged to complete a set of user stories and tasks that were required to advance the project. Throughout the sprint, the team actively managed the Sprint Backlog, monitoring progress and making the required changes. This adaptability was important, particularly when new challenges emerged or stakeholder input suggested that adjustments to functionality were necessary. In terms of Scrum artifacts, the backlog duration is estimated to span between 2 to 8 weeks. This timeframe indicates that the team is prepared to work on the project over multiple sprints, allowing for flexibility and responsiveness to emerging stakeholder needs. By using these Scrum artifacts, the VIAHALE project was able to maintain a clear focus on offering a dependable and user-friendly service while also ensuring that the development process remained flexible and responsive to the evolving needs of its customers.

## **3.5 MICROSERVICES ARCHITECTURE**

To enhance scalability, modularity, and system resilience, the proposed system adopts a Microservices Architecture. Each core function of the system is broken down into independently deployable services, communicating via REST APIs. Below is a breakdown of each service, their purpose, key endpoints, and interactions.

**Facility Reservation Service**

Purpose: Manages booking of rooms, equipment, and facility spaces.

|  |  |
| --- | --- |
| **Function** | **Endpoint** |
| Create reservation | POST /api/reservations/create |
| View availability | GET /api/facilities/availability |
| Cancel reservation | DELETE /api/reservations/cancel/:id |
| Approve/reject request | PATCH /api/reservations/approve/:id |

**Visitor Management Service**

Purpose: Handles pre-registration, check-ins, QR code generation, and visitor notifications.

|  |  |
| --- | --- |
| **Function** | **Endpoint** |
| Register visitor | POST /api/visitors/register |
| Generate QR code | GET /api/visitors/qr/:id |
| Visitor check-in/out | POST /api/visitors/checkin/:qr\_id |
| Host notification | POST /api/visitors/notify-host/:visitor\_id |

**Document Management Service**

Purpose: Secure storage, retrieval, sharing, and archiving of files.

|  |  |
| --- | --- |
| **Function** | **Endpoint** |
| Upload document | POST /api/documents/upload |
| View/download document | GET /api/documents/:id/view |
| Assign document access | POST /api/documents/share |
| Track document history | GET /api/documents/audit/:id |

**Legal Management Service**

Purpose: Handles contracts, permits, licensing, and compliance notifications.

|  |  |
| --- | --- |
| **Function** | **Endpoint** |
| Create contract | POST /api/legal/contracts/create |
| Track renewal | GET /api/legal/compliance/track |
| Notify expiry | GET /api/legal/reminder/:contract\_id |
| Upload permits | POST /api/legal/permits/upload |

**User & Role Management Service**

Purpose: Centralized user login, authentication, and role control.

|  |  |
| --- | --- |
| **Function** | **Endpoint** |
| User login/register | POST /api/auth/login |
| Assign role | PATCH /api/auth/roles/assign |
| Reset password | POST /api/auth/reset |
| User profile | GET /api/users/:id/profile |

**Notification Service**

Purpose: Handles all system notifications (email, SMS, push, in-app).

|  |  |
| --- | --- |
| **Function** | **Endpoint** |
| Send email | POST /api/notify/email |
| SMS blast | POST /api/notify/sms |
| Notify on approval | POST /api/notify/reservation/:id |

**Audit Trail / Logging Service**

Purpose: Logs user/system actions for traceability and legal auditing.

|  |  |
| --- | --- |
| **Function** | **Endpoint** |
| Log action | POST /api/logs/create |
| View logs | GET /api/logs?user=123 |
| Export logs | GET /api/logs/export |

## **3.6 DEVOPS IMPLEMENTATION**

This section outlines the tools and practices involved in each phase of the CI/CD pipeline for the system.

|  |  |  |
| --- | --- | --- |
| **Stage** | **Tools** | **Description** |
| Version Control | Git + GitHub | All code pushed and reviewed in branches via PRs. |
| CI – Continuous Integration | GitHub Actions | Run unit tests and syntax checks on code push/merge. |
| CD – Continuous Delivery | GitHub Actions | Push Docker images to a container registry. |
| Monitoring/Logging | Prometheus + Grafana | Monitor system health and logs. |

## **3.7 INTEGRATION APPROACH FOR INFORMATION SYSTEM**

The integration of ViaHale’s administrative modules such as Visitor Management, Facilities Reservation, Document Management, and Legal Management is achieved through a modular, API-driven approach. Each service communicates using RESTful APIs and standardized JSON formats to ensure smooth data exchange. Key integration techniques include data mapping for aligning fields like visitor ID and timestamps, and data transformation for formatting consistency. The system uses asynchronous communication protocols and a shared cloud database with role-based access control to maintain real-time synchronization and data integrity. This setup allows for seamless collaboration among microservices, improves operational visibility, and supports scalable deployment through CI/CD pipelines in a DevOps environment.

## **3.8 INTRODUCTION TO TOGAF AND THE FOUR ARCHITECTURAL DOMAINS**

As part of the enterprise architecture framework for the project, adherence to The Open Group Architecture Framework (TOGAF) is essential. TOGAF provides a comprehensive and widely adopted approach to enterprise architecture, emphasizing the four architectural domains. This structure ensures a unified architectural vision for ViaHale.

1. Business Process Architecture: The focus is on defining the operational workflows that support the ViaHale system. This includes visitor registration processes utilizing QR codes, vehicle tracking, service management procedures, and governance around biometric data capture. The aim is to ensure that these processes align with operational objectives such as efficiency, security, and user satisfaction.
2. Application Architecture: outlines the software systems and service interfaces that support business operations. For ViaHale, this encompasses the applications for QR code issuance and scanning, the OpenCV-powered facial recognition engine, the administrative dashboard, and integrations with backend systems and vehicle tracking modules. This architecture ensures interoperability and alignment with business workflows
3. Data Architecture: The focus is on how data is structured, stored, integrated, and secured. This involves designing data schemas for visitor and vehicle information, managing QR-token logs, biometric facial embeddings, and implementing encryption mechanisms. The architecture also includes lifecycle policies for data retention and access control, ensuring data integrity, consistency, and regulatory compliance.
4. Technology Architecture: Specifies the infrastructure required to support applications and data processes. In the context of ViaHale, this includes the computing infrastructure necessary for running the OpenCV facial recognition system, QR code scanners, backend servers, and network topology. It also addresses deployment environments, high-availability configurations, and integration with access control systems, ensuring a robust and scalable technology foundation.

By integrating these four architectural domains, the ViaHale project ensures that business objectives, system functionality, data governance, and technology infrastructure are fully aligned. This integrated approach provides a strong foundation for scalability, adaptability, regulatory compliance, and operational effectiveness, while remaining compatible with Agile Scrum practices.

# **CHAPTER IV**

# **REQUIREMENT ANALYSIS**

## **4.1 STAKEHOLDER IDENTIFICATION**

This section identifies all individuals or groups who are directly affected by the system or play a role in its development and implementation. Understanding these stakeholders is essential to ensure that the system meets the needs of its users and functions effectively within the organizational environment.

**Stakeholders and Descriptions:**

1. Administrative Staff – End-users who will operate the system to manage visitor logs, approve entries, and monitor security.
2. Visitors – External individuals who will use the QR code and facial recognition features to check in securely.
3. Security Personnel – Use the system to verify the identity of visitors and ensure safety protocols are followed.
4. System Developers – The capstone team responsible for designing, developing, and testing the system.
5. Project Manager – Oversees project progress, stakeholder communication, and ensures that all requirements are met.
6. IT Support – Handles potential integration, maintenance, and troubleshooting during or after system deployment.
7. Advisers and Panelists – Faculty members who evaluate the system’s feasibility, usability, and overall effectiveness.
8. Organization Head/Dean – Decision-maker who authorizes implementation and ensures it aligns with institutional goals.

## **4.2 REQUIREMENTS GATHERING TECHNIQUES**

## **4.3 USER STORIES AND USE CASES**

In line with Agile development practices, we used user stories to describe system functionalities from the perspective of end users. These stories guided the creation of features that deliver real value. Additionally, we modeled use case diagrams to visually represent how users interact with the system and its core modules.

**Facilities Reservation**

|  |  |
| --- | --- |
| **User ID** | **User Story** |
| US01 | As a Requestor, I want to submit a facility reservation request online so that I can secure a space without going to the office. |
| US02 | As a Requestor, I want to view available facilities so I can choose a suitable room for my activity. |
| US03 | As a Requestor, I want to check my reservation history so I can track past and upcoming bookings. |
| US04 | As a Facility Admin, I want to view submitted reservation requests so I can manage room bookings. |
| US05 | As a Facility Admin, I want to approve or reject requests so I can avoid double bookings or conflicts. |
| US06 | As a Facility Admin, I want to notify the requestor of the reservation status so they are updated. |
| US07 | As a Facility Admin, I want to cancel reservations when necessary so I can free up the facility for others. |

**Document Management**

|  |  |
| --- | --- |
| **User ID** | **User Story** |
| US01 | As a Legal Admin, I want to upload documents so that I can store legal files in the system. |
| US02 | As a Legal Admin, I want to edit or annotate documents so I can update or comment on content when needed. |
| US03 | As a Legal Admin, I want to search and retrieve documents so I can access specific files quickly. |
| US04 | As a Legal Admin, I want to view or download documents so I can review and use them offline. |
| US05 | As a Legal Admin, I want to track document versioning so I can monitor document changes and history. |
| US06 | As a Legal Admin, I want to assign user permissions so I can control who can access or edit files. |
| US07 | As a Legal Admin, I want to approve or reject documents so I can ensure only valid documents are processed. |
| US08 | As a Legal Admin, I want to archive documents so I can store inactive files properly. |
| US09 | As a Member of Other Department, I want to upload documents so I can share files with the legal team. |
| US10 | As a Member of Other Department, I want to request access to documents so I can view files that are restricted. |
| US11 | As a Member of Other Department, I want to view or download documents so I can read and use them. |
| US12 | As a Legal Officer, I want to track document versioning so I can review how a document has changed over time. |
| US13 | As for the System, I want to automatically classify documents so that all files are organized efficiently. |

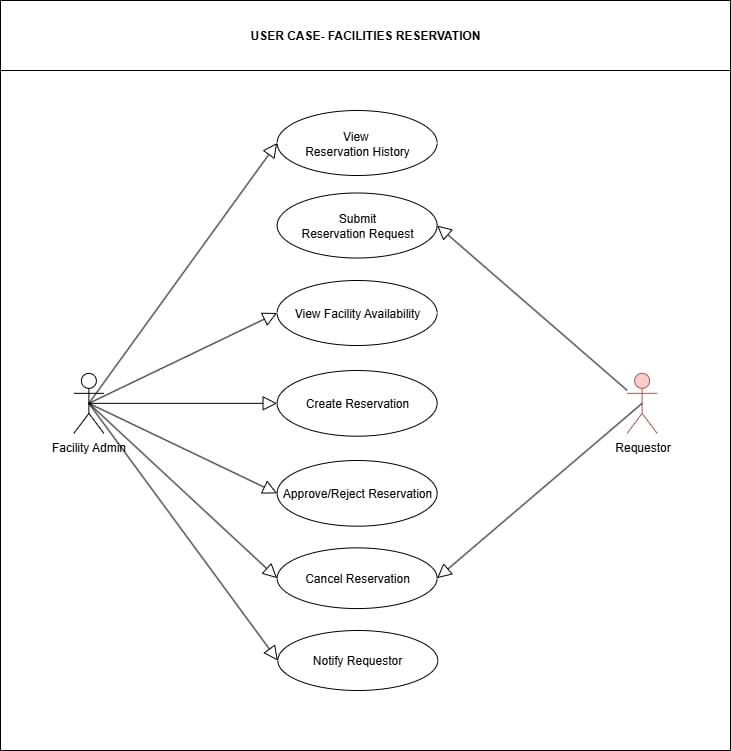
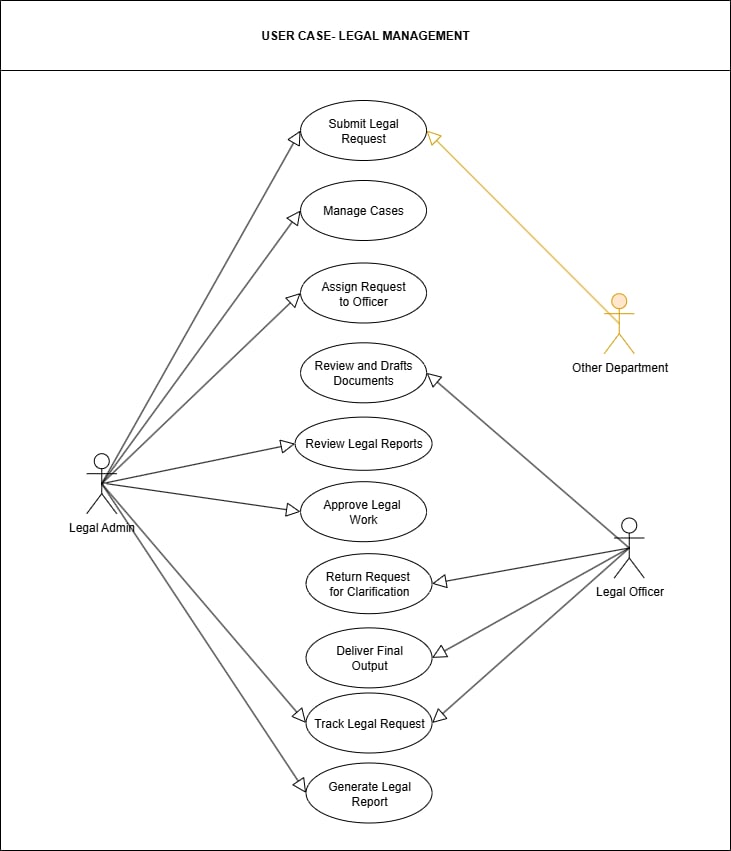
**LEGAL MANAGEMENT**

| **User ID** | **User Story** |
| --- | --- |
| US01 | As a Legal Admin, I want to submit legal requests so that I can initiate legal processes when needed. |
| US02 | As a Legal Admin, I want to manage legal cases so that all related documents and actions are organized. |
| US03 | As a Legal Admin, I want to assign legal requests to officers so that tasks are properly delegated. |
| US04 | As a Legal Admin, I want to review legal reports so that I can monitor the quality and progress of legal work. |
| US05 | As a Legal Admin, I want to approve legal work so that only validated outputs are finalized. |
| US06 | As a Legal Admin, I want to track legal requests so that I can monitor case status and progress. |
| US07 | As a Legal Admin, I want to generate legal reports so that I can provide summaries and insights to management. |
| US08 | As a Legal Officer, I want to review and draft legal documents so that I can provide accurate legal outputs. |
| US09 | As a Legal Officer, I want to return a request for clarification so that missing or unclear details can be corrected. |
| US10 | As a Legal Officer, I want to deliver final legal outputs so that the requestor receives the needed legal document. |
| US11 | As a Legal Officer, I want to track legal requests so that I can manage my assigned tasks effectively. |
| US12 | As a Member of Other Department, I want to submit legal requests so that I can get legal support or review for our documents. |

**Visitor Management**

|  |  |
| --- | --- |
| **User ID** | **User Story** |
| US01 | As a Visitor Management Admin, I want to register visitors so that their information is properly recorded. |
| US02 | As a Visitor Management Admin, I want to approve or reject visit requests so I can control access to the facility. |
| US03 | As a Visitor Management Admin, I want to monitor the visit log so I can track who enters and exits the premises. |
| US04 | As a Visitor Management Admin, I want to manage blacklisted visitors so that restricted individuals are denied entry. |
| US05 | As a Visitor Management Admin, I want to view scheduled visits so I can prepare for upcoming guests. |
| US06 | As the System, I want to generate a visitor registration link so visitors can pre-register online. |
| US07 | As the System, I want to issue an access pass so that authorized visitors can enter securely. |
| US08 | As the System, I want to check in visitors automatically so their arrival is logged in real-time. |
| US09 | As a Requestor, I want to register a visitor so I can invite someone to visit our facility. |
| US10 | As a Visitor, I want to self-register as a visitor so I can enter the building with proper authorization. |

**Use Case Diagram Overview**

To visualize how users interact with the system, we developed a Use Case Diagram that illustrates the relationship between user roles and the primary system functions.

## 

## 

## 

## 

## 

## **4.4 FUNCTIONAL REQUIREMENTS FOR INTEGRATION**

**FACILITIES RESERVATION**

| **Req ID** | **Description** | **Input** | **Output** | **Notes** |
| --- | --- | --- | --- | --- |
| FR-01 | The system shall send reservation payload via HTTPS (JSON) to notification service within 1 second of submission. | reservation\_id, facility\_id, timeslot | HTTP 200 ACK | Validate fields; implement retry logic up to 3 times on errors. |
| FR-02 | The system shall query facility availability via API and return status within 2 sec. | facility\_id, timeslot | available/unavailable | Handle concurrent queries; cache results for 5 minutes. |
| FR-03 | The system shall update backend and trigger confirmation notification within 2 sec on approval. | reservation\_id, status | backend update + notification | Implement retry mechanism (up to 2 attempts) upon failure; log errors comprehensively |
| FR-04 | The system shall log rejection reason and notify requestor within 30 sec via notification API. | reservation\_id, rejection\_reason | rejection notification | Use HTTPS/OAuth2; log detailed error messages. |
| FR-05 | The system shall release facility timeslot and send cancellation notice within 30 sec. | reservation\_id | availability release + notification | Log all activities; ensure rollback capability if updates fail. |

**DOCUMENT MANAGEMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req ID** | **Description** | **Input** | **Output** | **Notes** |
| DM-01 | The system shall allow uploading documents through HTTPS API within 2 seconds and return status. | file\_data, metadata, user\_id | HTTP 200 ACK / error | Validate file type/size; implement retry on failure. |
| DM-02 | The system shall provide an API for document retrieval with role-based access validation. | document\_id, user\_token | document\_data or 403 | Ensure JWT/OAuth2 token validation. |
| DM-03 | The system shall support annotation updates to existing documents via JSON payload. | document\_id, annotation\_data | update status | Use versioning to track changes; audit trail required. |
| DM-04 | The system shall allow internal users to request access, triggering approval workflow. | user\_id, document\_id | access\_request\_log | Notify Legal Officer; await approval. |
| DM-05 | The system shall notify the Legal Officer of new document uploads needing classification or approval. | document\_id, uploader\_id | notification (email/in-app) | Use event-based messaging (e.g., webhook or message queue). |
| DM-06 | The system shall automatically assign version numbers for edited or re-uploaded documents. | document\_id, version\_meta | new\_version\_id | Triggered by edit or re-upload; maintain history. |
| DM-07 | The system shall allow Legal Admin to assign user permissions via secure API. | user\_id, permission\_type | permission\_updated | Log actions for security audit. |
| DM-08 | The system shall return document classification labels from AI engine or manual input. | document\_id | classification\_label | Allow integration with AI classifier API. |
| DM-09 | The system shall archive old versions of documents to a storage service and update metadata. | document\_id, version\_id | archive\_status | Use secure storage; log operation. |
| DM-10 | The system shall log all document access and changes for auditing. | action\_type, user\_id, timestamp | log\_entry | |  | | --- | |  |  |  | | --- | | Store in audit trail DB; tamper-proof logging required. | |

**LEGAL MANAGEMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req ID** | **Description** | **Input** | **Output** | **Notes** |
| LM-01 | The system shall receive legal request submissions from other departments via REST API. | request\_details, user\_id | request\_id, status | Validate required fields; log submission. |
| LM-02 | The system shall allow the Legal Admin to assign legal requests to officers using a role-based access API. | request\_id, officer\_id | assignment\_confirmation | Notify the assigned officer via email/in-app. |
| LM-03 | The system shall support collaborative drafting by syncing document changes between Legal Admin and Officer. | doc\_id, content\_changes | updated\_document | Use WebSocket or polling; maintain version history. |
| LM-04 | The system shall retrieve and present legal report drafts for review with comments API. | report\_id | report\_data, comments | Allow inline commenting and thread tracking. |
| LM-05 | The system shall trigger a legal work approval API upon final review completion. | report\_id, approval\_status | approval\_confirmation | Store timestamps; send notification to the requestor. |
| LM-06 | The system shall return legal requests to the originator with clarification notes via feedback API | request\_id, notes | return\_status | Include detailed feedback log |
| LM-07 | The system shall deliver final approved legal output to the originating department via secure delivery API. | request\_id, final\_output | delivery\_acknowledgment | Encrypt documents; log delivery status. |
| LM-08 | The system shall track the status and lifecycle of legal requests through integration with the case tracking system. | request\_id | status\_log   |  | | --- | |  |  |  | | --- | |  | | Sync with task manager or dashboard. |
| LM-09 | The system shall auto-generate legal reports using template integration and metadata population. | case\_data, template\_id | formatted\_report | Connect to document generation service. |

## 

**VISITOR MANAGEMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req ID** | **Description** | **Input** | **Output** | **Notes** |
| VM-01 | The system shall generate a unique visitor registration link upon request via secure API. | requestor\_id, visit\_purpose, date | registration\_link | Auto-expire link after configurable time (e.g., 48 hrs). |
| VM-02 | The system shall allow visitors to self-register through the generated registration portal. | visitor\_details, visit\_info | registration\_confirmation | Validate fields; prevent duplicate entries; use CAPTCHA. |
| VM-03 | The system shall notify the front desk/admin for every new visitor registration via webhook or notification API. | registration\_id | notification\_payload | Include visitor name, schedule, and purpose. |
| VM-04 | The system shall allow the admin to approve or reject visit requests via secure API. | registration\_id, decision, reason | approval\_status | Log decision and timestamp; notify requestor. |
| VM-05 | The system shall issue a temporary access pass with QR code upon visit approval. | registration\_id, visit\_schedule | QR\_code\_pass | Integrate with the physical access control system. |
| VM-06 | The system shall check in visitors by scanning their QR code and logging entry time. | QR\_code | checkin\_status | Allow retry; log check-in with timestamp and location. |
| VM-07 | The system shall allow admins to monitor live visit logs through an API feed or dashboard integration. | filters (date, visitor, status) | visit\_log\_data | Provide real-time updates; include sorting/filtering. |
| VM-08 | The system shall maintain a blacklist database and cross-check new registrations. | visitor\_name, ID\_number | blacklist\_match | Block auto-approval; flag for manual review. |
| VM-09 | The system shall enable admins to schedule and view upcoming visits via calendar API integration. | date\_range, filters | scheduled\_visits | Sync with calendar tools (e.g., Google/Outlook Calendar). |

# **BUSINESS PROCESS ARCHITECTURE**

## **5.1 IDENTIFICATION OF BUSINESS PROCESSES**

This section outlines the key processes the system supports or automates across various departments.

1. **Facilities Reservation: Room/Facilities Booking**

Other departments request a room/facility to facility admin for meetings or events via a centralized facilities reservation system. This system supports availability tracking, booking conflict resolution, automated notifications, and automated approval workflows.

1. **Visitor Management: Guest Registration and Access Control**

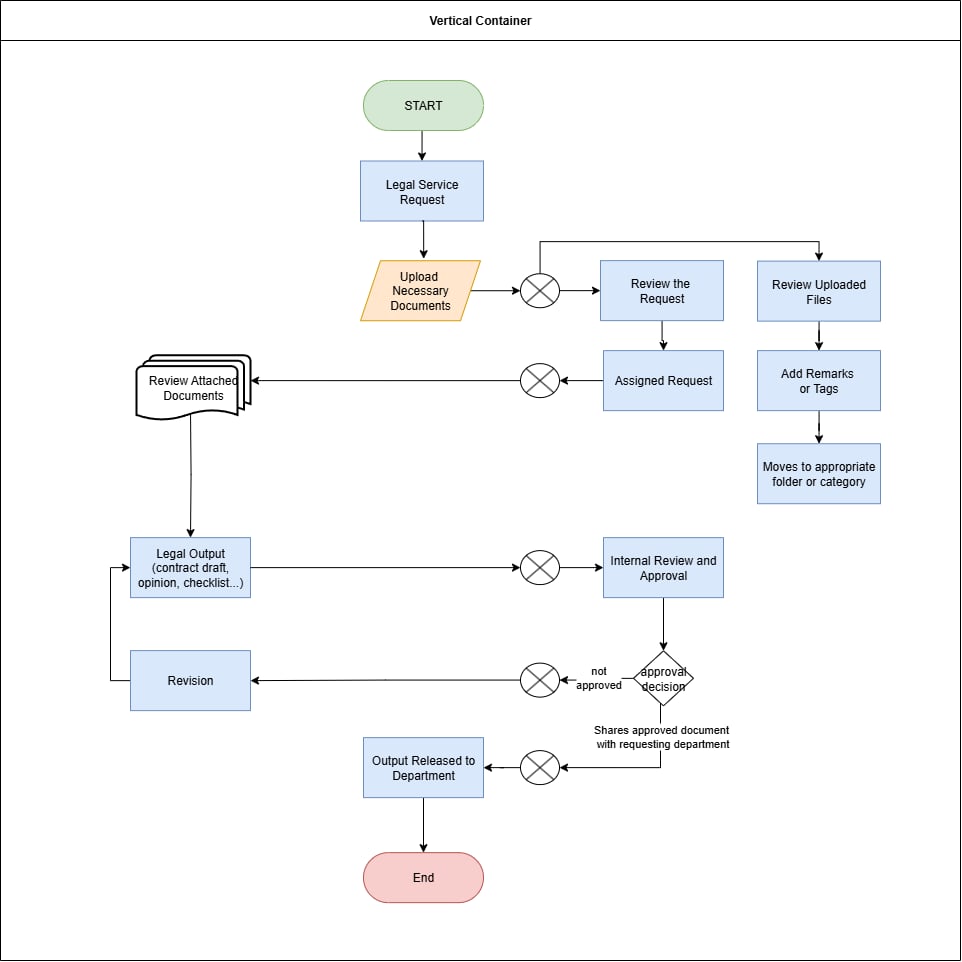
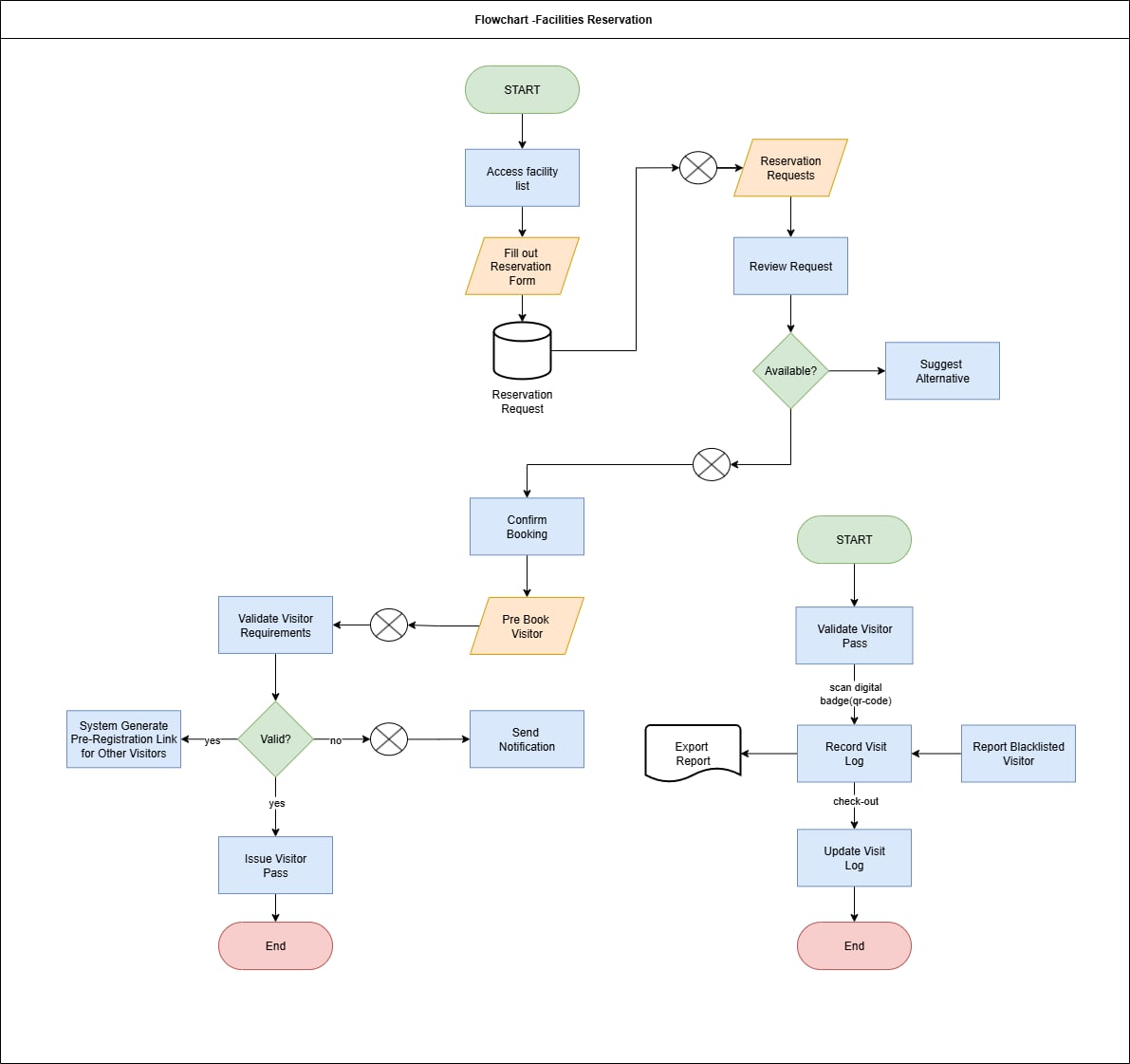
Visitors are pre-registered with facial recognition and given digital badges (Qr-code) as their visitor passes with limited permissions. They monitor via Facial recognition powered by OpenCV for access control. This system supports digital form registration, real-time check-ins/check-out, and integration with security systems.

1. **Legal Management: Legal Matter Lifecycle Management**

Legal Matters are created, reviewed, approved, and archived. This system supports digital signatures, automated reminders, and compliance checks.

1. **Document Management: Document Storage, Retrieval and Approval**

Users upload documents, categorize, search, and request approvals for internal documents. This system supports metadata tagging, version control, access rights, audit logs, and approval routing.

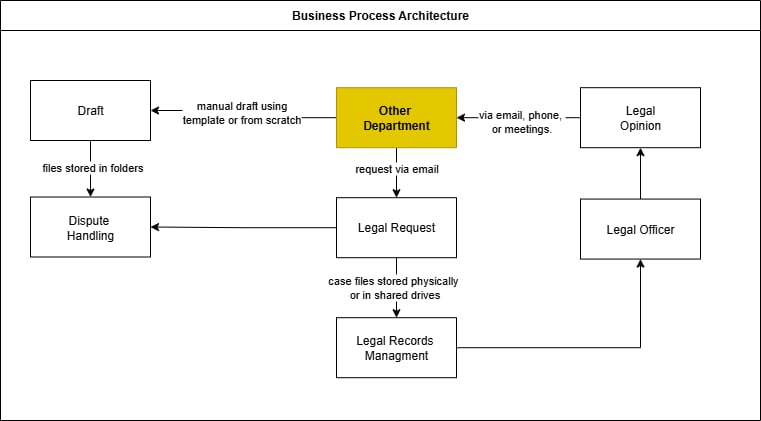
**5.2 BUSINESS PROCESS DIAGRAMS**

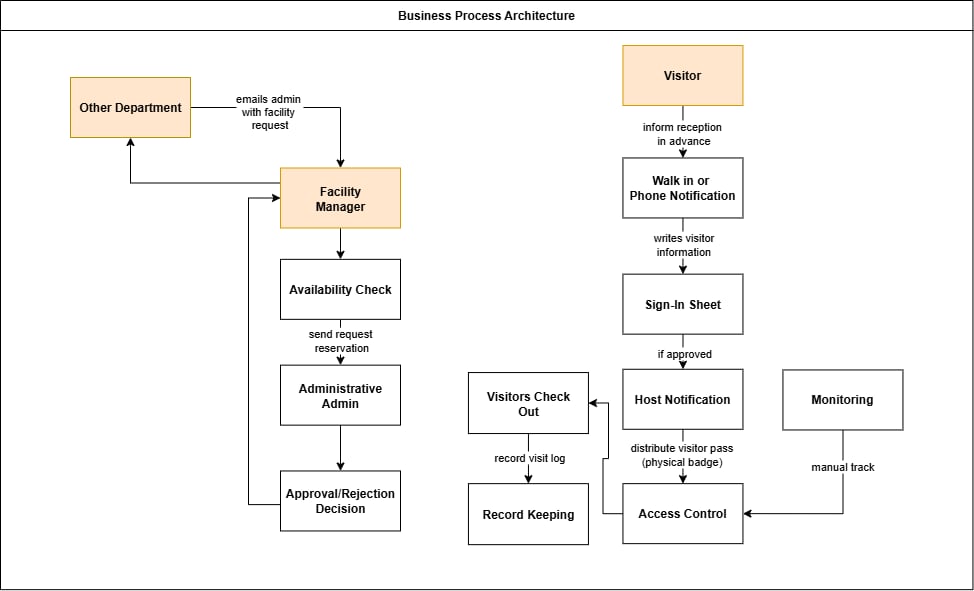
## 

## **5.3 ALIGNMENT OF INTEGRATED SYSTEM WITH BUSINESS PROCESSES**

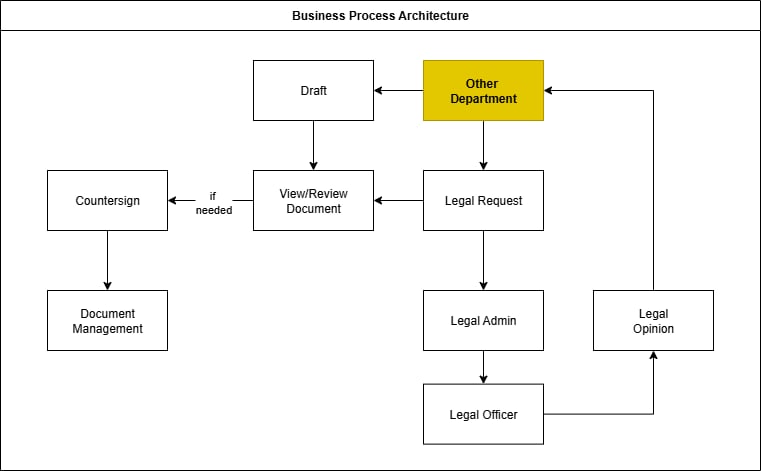
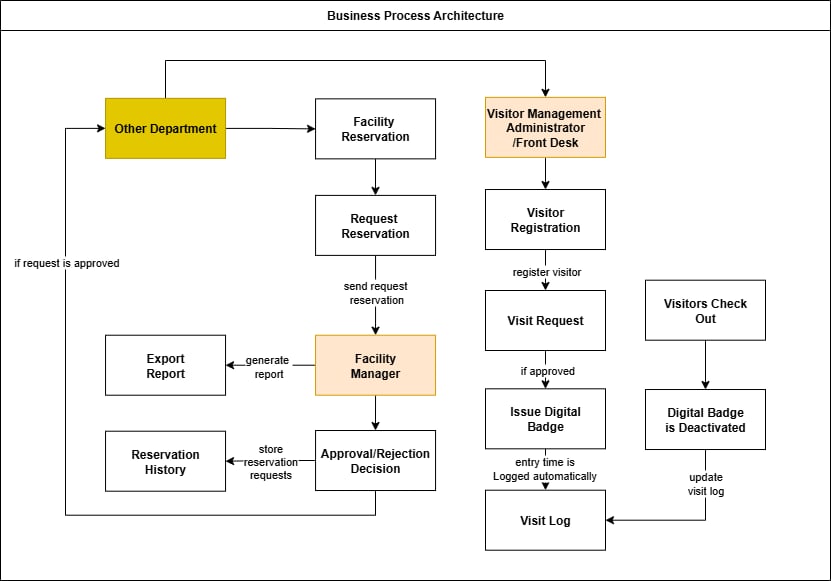
## **5.4 BUSINESS PROCESS IMPROVEMENTS**

The transition from the As-Is process to the To-Be system brings significant enhancements in administrative operations, particularly in the following areas:

**As-Is**



**To-Be**



# 

# **APPLICATION ARCHITECTURE**

## 

**6.1 COMPONENTS OF APPLICATION ARCHITECTURE** This outlines the system’s modular design and layered structure, mapping responsibilities across key components.

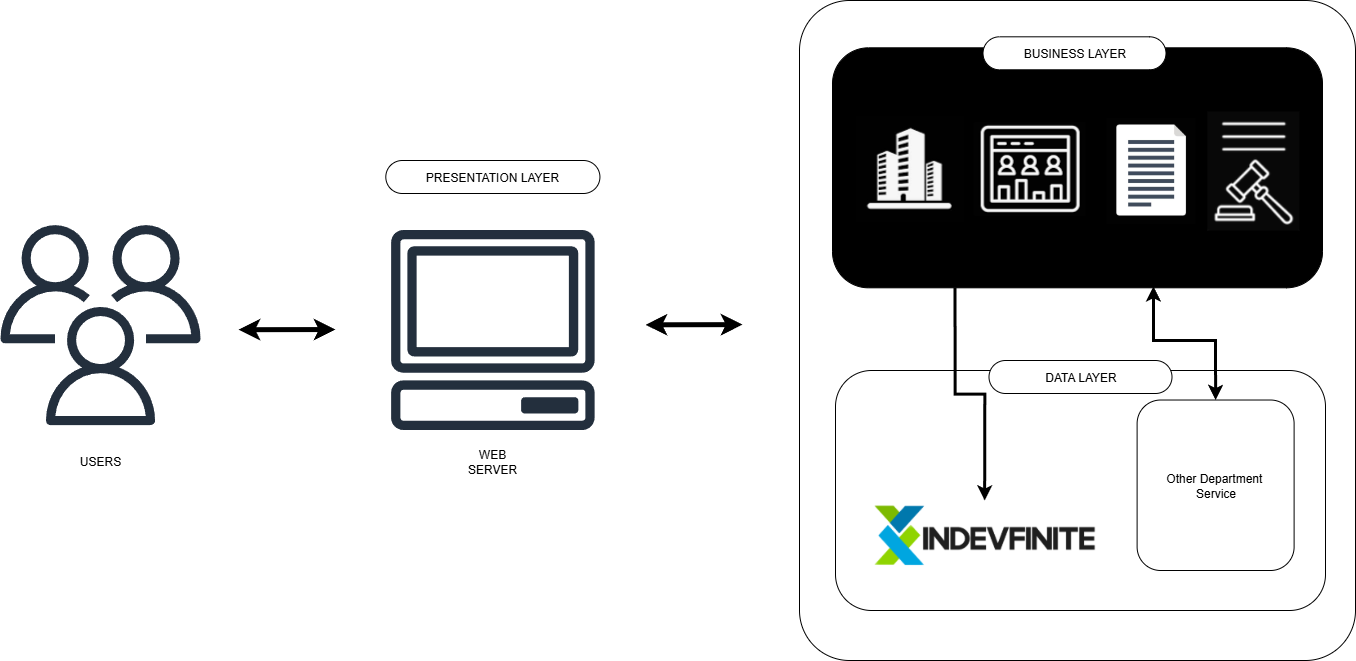
|  |  |  |
| --- | --- | --- |
| SubModules | Description | Primary Users |
| Facilities Reservation | Bookings of rooms/resources, approval workflows | Admin, Facility Admin |
| Visitor Management | Pre-Registration, badge issuance, access tracking | Visitor Management Administrator/Front Desk Admin |
| Document Management | File Storage, Version Control, Approvals | All Departments |
| Legal Management | Signing, Compliance, Contract | Legal Admin, Legal Officer |

|  |  |
| --- | --- |
| Layer | Responsibility |
| Presentation Layer | UI components (Web), ensures user-friendly interaction. |
| Business Logic Layer | Core rules and workflows: facilities reservation management, visitor management, document management, and legal management and cross module interactions. |
| Service Layer | API endpoints for internal/external access(REST), act as the interface between frontend/backend and enables integration with third parties. |
| Data Access Layer | Handles DB queries, ORM mappings, caching logic, and secure reads/writes. |
| Integration Layer | Manages communication across modules via event brokers, message queues, or API gateways. Enables async flow and cross department automation. |
| Security Layer | Authentication access control per role/submodule, encryption, logging, and monitoring. |
| Infrastructure Layer | Cloud deployment logic, logging, and observability |

## 

## **6.2 APPLICATION ARCHITECTURE DIAGRAMS**

This diagram presents the structural layout of the enterprise application, outlining both functional modules (Facilities Reservation, Visitor Management, Document Management, and Legal Management) and architectural layers (Presentation, Business Logic, Integration). It helps clarify data flow, module responsibilities, and deployment consideration.



*Figure #: Application Architecture depicting submodules and layered responsibilities.*

## **6.3 INTEGRATION OF SOFTWARE MODULES**

This section explains how individual modules including Facilities Reservation, Visitor Management, Document Management, and Legal Management interact across architectural layers to deliver end to end functionality.

|  |  |  |  |
| --- | --- | --- | --- |
| Sub Modules | Interacts With | Integration Method | Purpose |
| Facilities Reservation | Visitor Management, | RESTful APIs + Even Notifications | Bookings trigger visitor pre-registration. |
| Visitor Management | Facilities Reservation |  |  |
| Document Management | Legal Management |  | Stores signed documents. |
| Legal Management | Document Management |  | Legal actions initiate document workflows. |

## 

## **6.4 COMMUNICATION AND INTERACTION PATTERNS**

This section explains how software components communicate, the protocols in use, and the architectural patterns that govern interaction flow across modules and layers.

|  |  |  |
| --- | --- | --- |
| Protocol | Use Case | Modules Involved |
| HTTP/HTTPS | Web UI interactions, external API access | Presentation Layer, External Services. |
| REST | CRUD operations across microservices | All modules via API Gateway |

# **DATA ARCHITECTURE**

## 

## **7.1 DATA SOURCES AND TYPES**

## 

## **7.2 DATA FLOW DIAGRAMS**

## 

## 

## **7.3 DATA STORAGE AND MANAGEMENT**

To detail the underlying database technologies, storage strategies, and access methods used across the system modules for secure, efficient, and scalable data management.

|  |  |
| --- | --- |
| Aspect | Details |
| Database Engine | MySQL 5.2.1 |
| Hosting Platform | IndevFinite Web Hosting - managed DB instance or custom deployment |
| Storage Type |  |
| Structure | Relational Schema - normalized tables, foreign keys, indexes. |
| Access Method | Direct SQL queries |

All module data (reservation, visitor logs, documents, and legal entries reside in MySQL tables with clearly defined relationships. Access granted via API gateways and application layer controls limited by user role and module permissions.

## **7.4 DATA SYNCHRONIZATION ACROSS SYSTEM**

To explain how the system maintains data consistency and ensures real time updates across interconnected modules deployed within the IndevFinite cloud ecosystem.

|  |  |
| --- | --- |
| Component | Role |
| Central MySQL DB | Single source of truth for all business records. |
| IndevFinite API Layer | Unified access point for modules; handles validation and formatting. |

# **8. TECHNOLOGY ARCHITECTURE**

## **8.1 TECHNOLOGY STACK AND INFRASTRUCTURE**

## 

|  |  |
| --- | --- |
| **LOGO** | **DESCRIPTION** |
|  | PHP is an open-source server-side scripting language used to implement backend logic and RESTful API endpoints in our system. We chose PHP because it is widely supported, easy to deploy, and integrates seamlessly with our hosting environment, enabling efficient server-side processing and maintenance. |
|  | MySQL is a relational database management system that stores structured information such as user accounts, transactions, and system logs. It pairs reliably with PHP, offering ACID compliance, efficient querying, and well-established tooling. MySQL suits our needs for data consistency, scalability, and cost-effectiveness. |
|  | REST (Representational State Transfer) defines a stateless, HTTP-based interface that enables our frontend and backend to communicate via standardized CRUD operations. We adopt REST because it simplifies integration, decouples components, and supports interoperability across platforms and clients. |
|  | HTML5 is the standardized markup language used to structure webpages and applications. It includes modern APIs like web storage. We use HTML5 to build semantic, accessible, and responsive content ensuring consistent rendering across browsers, faster loading, and compatibility for both mobile and desktop users. |
|  | CSS (Cascading Style Sheets) defines the presentation and layout of web pages, including styling rules, colors, fonts, and responsive breaks. In our system, CSS is used to create visually appealing and adaptive interfaces that work smoothly across different devices and screen sizes, enhancing user experience and consistency. |
|  | JavaScript is a high-level, dynamic, and interpreted programming language that is widely used for web development. It was originally created to enable interactive web pages and is an essential part of web applications. |
|  | API Gateway acts as the centralized entry point for all client requests. It routes traffic to appropriate backend services, enforces authentication, rate limiting, caching, and transforms requests or responses as needed. This layer centralizes management of security, scalability, and monitoring, simplifying service interactions and improving observability across the system |
|  | GitHub hosts our source code repositories and enables collaborative development through pull requests, branching, and code reviews. It provides a stable version control system and supports multiple workflows, promoting organized and transparent software development. |
|  | GitHub Actions is our CI/CD platform. It allows us to automate build, test, and deployment workflows triggered by repository events like push or pull requests. Its native integration with GitHub, marketplace of reusable actions, and hosted runners reduce setup complexity while enhancing development speed and reliability. |
|  | XAMPP is a free, cross-platform development bundle including Apache, MariaDB (MySQL), PHP, and Perl. This environment mirrors production locally for faster debugging and prototyping. Because it bundles all required services in a single install, it enables rapid setup and consistent development workflows. We use XAMPP during development to simulate the server environment before deployment. While not recommended for production due to its default insecure configuration, it’s ideal for local testing and alignment with our live setup. |
|  |  |
|  | Monday.com is a cloud-based Work Operating System designed for planning, tracking, and collaborating on work through boards, dashboards, automations, and integrations. We use monday.com to coordinate tasks, visualize workflows, automate notifications, and maintain transparency across the team boosting productivity and ensuring project clarity. |

## **8.2 SOFTWARE TECHNOLOGIES**

The technological foundation of this project is built upon a comprehensive selection of programming languages and libraries, strategically chosen to deliver a high-performance and scalable application. This technology stack combines proven web development tools with specialized libraries to create a robust system capable of handling complex functionality while maintaining optimal user experience.

**BACKEND TECHNOLOGIES**

At the heart of the backend lies PHP, a robust server-side scripting language that acts as the engine driving the application's logic. PHP handles the intricate processing of user requests, interacts seamlessly with the database, and orchestrates the flow of information between the user interface and the data store. Working in tandem with PHP is MySQL, a powerful relational database management system (RDBMS) that efficiently organizes, stores, and retrieves the data that fuels the application, meticulously safeguarding the application's valuable information.

**Frontend Technologies**

The user interface is built upon the foundational trio of web technologies. HTML provides the structural scaffolding, defining the content, layout, and basic elements that form the visual landscape. CSS brings style, elegance, and visual appeal to the structure, while Bootstrap 5 serves as a comprehensive UI toolkit that simplifies the creation of responsive and visually appealing web pages. Bootstrap 5 ensures that the application looks and functions flawlessly across multiple devices and screen sizes.

JavaScript breathes life into the static HTML structure, adding dynamism and interactivity to the user interface. This powerful scripting language empowers the application to respond to user actions, update page content dynamically without requiring full page reloads, and create a more engaging and intuitive user experience.

**Specialized Libraries**

Extending the application's capabilities beyond traditional web development is the OpenCV library. This powerful computer vision library enables the application to incorporate sophisticated image processing and analysis functionalities, opening up possibilities for advanced image-based interactions and data analysis.

**Technology Integration**

These technologies work cohesively as an integrated system where PHP processes server-side logic and database interactions, HTML/CSS/JavaScript handle client-side presentation and user interactions, MySQL manages data persistence, and OpenCV provides specialized computer vision capabilities. This combination delivers a robust, efficient, and feature-rich application while maintaining simplicity and direct control over each component.

## 

## **8.3 SCALABILITY AND PERFORMANCE CONSIDERATIONS**

To ensure the Transport Network Vehicle Service Administrative System remains reliable and responsive as usage and data volume grow, the architecture incorporates several scalability and performance focused design strategies.

1. **Cloud Native Design (IndevFinite):** Hosted on scalable cloud infrastructure capable of vertical scaling (CPU/RAM) and horizontal scaling through service replication
2. **Optimized MySQL Schema Design:** Normalization balanced with performance indexing to support high read/write throughput.
3. **Connection Pooling:** Minimizes overhead from repeated database connections under traffic loads.
4. **Real-Time Metrics Collection**: System health and performance indicators tracked using Prometheus and Grafana.
5. **API First Approach:** External integrations are supported through scalable RESTful APIs.

# **DEVELOPMENT PROCESS**

## **9.1 AGILE SCRUM ROLES AND RESPONSIBILITIES**

## 

## **9.2 SPRINT PLANNING AND BACKLOG MANAGEMENT**

## 

## **9.3 SPRINT EXECUTION AND DELIVERABLES**

## 

## **9.4 CHALLENGES FACED IN THE DEVELOPMENT PROCESS**

Throughout the development of the Transport Network Vehicle Service Administrative System, the team encountered several key challenges. The table below summarizes these obstacles along with the strategies used to address them.

|  |  |  |
| --- | --- | --- |
| **Challenge** | **Description** | **Resolution/Mitigation Strategy** |
| Topic Selection |  | Conduct preliminary literature reviews; consult instructor for guidance. |
| Research Skills Gap | Inexperience with methodologies, academic writing and technical skills needed for project development. | Attend seminars and use online tutorials; seek help from research advisors. |
| Time Management | Balancing research with personal commitments. | Create a realistic timeline. Use productivity tool([Monday.com](http://monday.com)) to break tasks into manageable chunks. |
| Group Dynamics | Uneven contribution or communication issues in group research. | Set clear expectations; assign roles early. |
| Documentation and Formatting | Struggles with formatting and academic standards. | Follow Bestlink College of the Philippines templates and guides. |
| Data Collection Challenges | Difficulty accessing participants or gathering reliable data. | Use online surveys; adjust scope; consider alternative data sources or simulated datasets |

## 

## 

# **References**

# 

**AGILE SCRUM METHODOLOGY OVERVIEW**

1. Haigh, A. J. (2025). Agile Culture as Experienced by Scrum Masters in the Financial Sector (Doctoral dissertation, Walden University).
2. Kohansal, M. A. (2024). Navigating Enterprise Architecture (EA) Institutionalization: The Interplay of EA and Agile.
3. Magnussen, E. F., Havnelid, E., & Molléri, J. S. (2024). The Product Owner and its Impact on Success and Challenges in Agile Scrum Projects: An Interview Study. *Procedia Computer Science*, *239*, 1157-1164.
4. Schwaber, K., & Sutherland, J. (2020). The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org.
5. Kohansal, M. A. (2024). Navigating Enterprise Architecture (EA) Institutionalization: The Interplay of EA and Agile.

**Enterprise Architecture Concepts**

*1. Ettinger, A. (2025). Enterprise architecture as a dynamic capability for scalable and sustainable generative AI adoption: Bridging innovation and governance in large organisations. Sustainability, 16(20), 8865.*

*2. O’Higgins, D. (2023). Impacts of business architecture in the context of digital transformation: An empirical study using PLS-SEM approach. arXiv.*

3. Sararuch, S., Palaiahnakote, P., & Kim, K. (2023). A review of enterprise architecture frameworks for digital transformation in public administration. *Journal of Enterprise Information Management*, 36(3), 729–751.

*4. Rehman, Z., Tariq, N., Moqurrab, S. A., Yoo, J., & Srivastava, G. (2023). Machine learning and Internet of Things applications in enterprise architectures: Solutions, challenges, and open issues. Expert Systems, 40(6), e13447*

5. Bernard, S. A. (2020). *An introduction to enterprise architecture* (4th ed.). AuthorHouse.

6. Gartner Research. (2021). *Best practices for enterprise architecture: Achieving business outcomes*. Gartner, Inc.

*7. van de Wetering, R. (2021). Dynamic enterprise architecture capabilities and organizational benefits: An empirical mediation study. arXiv.*

**INTEGRATION OF INFORMATION SYSTEMS**

1. McKinsey & Company. (2021). *The digital transformation of industries: How to integrate legacy systems with modern applications*. <https://www.mckinsey.com/industries>
2. Kusumawati, R. (2025). Integrating big data analytics into supply chain management: Overcoming data silos to improve real-time decision-making. *International Journal of Advanced Computational Methodologies and Emerging Technologies*, *15*(2), 17-26.
3. Ramadhani, W., Khuzaini, K., & Shaddiq, S. (2024). Resistance to change: Human resources issues in the implementation of Industry 4.0 technology. *Proceeding: Islamic University of Kalimantan*.
4. Herath, H. M. S. S., Herath, H. M. K. K. M. B., Madhusanka, B. G. D. A., & Guruge, L. G. P. K. (2024). Data protection challenges in the processing of sensitive data. In *Data Protection: The Wake of AI and Machine Learning* (pp. 155-179). Cham: Springer Nature Switzerland.
5. Sadeghi, M., Carenini, A., Corcho, O., Rossi, M., Santoro, R., & Vogelsang, A. (2024). Interoperability of heterogeneous Systems of Systems: from requirements to a reference architecture. *The Journal of Supercomputing*, *80*(7), 8954-8987.
6. Striped Giraffe Team. (2025). *IT systems integration: Benefits, challenges, solutions for scalable business growth*. https://www.striped-giraffe.com/en/blog/it-systems-integration-benefits-challenges-solutions-for-scalable-business-growth/
7. Haider, K. (2025, July 24). What is data integration? Definition, best practices, and use cases. https://www.astera.com/type/blog/data-integration/