**Department of Computing and Information Systems**

**INDT 4216 - Industrial Training Program**

**Industrial Research Report**

Details of the Student:

|  |  |
| --- | --- |
| Registration Number | 192100 |
| Name | S.H.M.Chathuri Sathya Piyarathne |
| Supervisor’s Name | Prof.V.G.T.N.Vidanagama |
|  |  |

**Table Of Content**

[**Chapter 1: Introduction 3**](#_74ck692pgupw)

[**1.1 Background of the Study 3**](#_im853dkdlekw)

[1.2 Objectives of the Research 3](#_fxapop10tvnt)

[1.3 Scope of Work 4](#_7ie51pnwzth9)

[**Chapter 2: – Problem Specification and Literature Survey 5**](#_fka1fj468p0c)

[2.1 Problem Statement 5](#_xab9h6lj0sn8)

[2.2 Research Questions 5](#_xab9h6lj0sn8)

[2.3 Literature Review 6](#_xab9h6lj0sn8)

[2.3.1 Previous Work and Studies 6](#_8467ehfautb3)

[2.3.2 Gaps in the Existing Research 6](#_8wvw17xtl4v6)

[**Chapter 3: Design Methodology 7**](#_lnp0l6ul1evs)

[3.1 Proposed System Design 7](#_gx5d9lx083wa)

[3.3 Tools and Technologies Used 10](#_kjesbu1m2rnj)

[**Chapter 4: Implementation 14**](#_rxyw5hcvfq5i)

[4.1 Development Process 14](#_bjz2i7llivnm)

[**Chapter 5:Evaluation 19**](#_wisgpouwhz9k)

[5.1 Testing Methods 19](#_dim5bubzeatn)

[5.2 Performance Metrics 24](#_g983cicw6mv7)

[5.3 Comparison with Existing Systems 24](#_2vpzmqkjcig0)

[**Chapter 6: Discussion 25**](#_8ofeqeft05q6)

[6.1 Key Findings 25](#_dmnix9pl7iul)

[6.2 Interpretation of Results 25](#_s00wd3tapuj)

[**Chapter 7: Conclusion 27**](#_c3q00dxw3un4)

[7.1 Summary of Findings 27](#_bz9ip18cggh1)

[7.2 Contributions to the Field 28](#_vkl2j1wg96jr)

**Table of Figures**

[Figure 3.1: Work flow chart of Request New Asset 10](#_8jkrf6623q74)

[Figure 3.2 : MySQL Frame 12](#_3stm8cezz45)

[Figure 3.3 : Android Studio Frame 13](#_earlx8bjo7h9)

[Figure 3.4 : Flutter frame 13](#_b6xpvbruzvgh)

[Figure 3.5: Dart Frame 13](#_b9o1lin7vvga)

[Figure 3.6: Gradle Frame 14](#_k2fd5f2hp4f)

[Figure 4.1 : Request New Asset Form - Initial Part 17](#_r4rk8c2kfc5)

[Figure 4.2 : Request New Asset Form - Latter Part 18](#_19avs6kikc49)

[Figure 5.1 : Filled Request New Asset Form - Initial Part 20](#_njdvlvddydbm)

[Figure 5.2 : Filled Request New Asset Form - Latter Part 21](#_mtkgw48bybj9)

[Figure 5.3 : All Request Details Page 22](#_9nvpo8syh7wy)

[Figure 5.4 : Card Details 23](#_q8y8wr14h0ch)

[Figure 5.5 : Search Region wise 24](#_w9mf81dj970r)

# Table of Tables

# 

[Table 6.1 : Quantitative comparison between manual request and mobile app request 28](#_ktthubl8p20m)

# Acronyms

# 

# ERP Enterprise Resource Planning

# IoT Internet of Things

# AI Artificial Intelligence

# UAT User Acceptance Testing

# CRUD Create, Read, Update, Delete

# IDE Integrated Development Environment

# API - Application Programming Interface

# SDK Software Development Kit

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Chapter 1: Introduction

# 1.1 Background of the Study

In today’s fast-paced business environment, the efficient management of asset requests is crucial for maintaining operational continuity and maximizing resource utilization. New asset requests ranging from equipment and technology to infrastructure are pivotal in ensuring that organizations can adapt swiftly to changing demands and sustain their competitive edge. However, traditional asset request management systems often suffer from inefficiencies, such as prolonged approval processes, lack of transparency, and misalignment between departments. These challenges can lead to delayed asset acquisition, increased operational costs, and reduced productivity. This research aims to explore and develop strategies for optimizing the new asset request management process. By leveraging modern technologies and process re-engineering, the study will identify key areas for improvement and propose a framework that enhances the efficiency, accuracy, and responsiveness of asset request management. The ultimate goal is to create a streamlined, transparent, and agile system that aligns with organizational objectives, minimizes delays, and fosters a more productive work environment

## 1.2 Objectives of the Research

* **To Identify Key Challenges in Current Asset Request Processes:**

This research will first seek to identify and analyze the common pain points and inefficiencies in existing asset request management systems across different industries.

* ·**To Develop a Framework for Process Optimization**:

Based on the challenges identified, the study will propose a comprehensive framework for optimizing new asset request processes. This framework will incorporate best practices, automation technologies, and enhanced communication strategies to streamline workflows and reduce approval times.

* **To Assess the Impact of an Optimized Asset Request System on Operational Efficiency**:

The research will evaluate how the implementation of an optimized asset request management system can improve overall operational efficiency, including metrics such as request fulfillment time, resource utilization, and organizational responsiveness.

* **To Provide Strategic Recommendations for Organizations**:

Finally, the research will offer actionable recommendations for organizations seeking to modernize their asset request management processes. These recommendations will be tailored to different organizational sizes and structures, ensuring broad applicability.

## 1.3 Scope of Work

The scope of this research is to thoroughly investigate and improve the process involved in requesting new assets within industrial environments. The primary objective of this study is to develop a comprehensive and efficient asset request management framework that can streamline the approval process, reduce delays, and enhance overall asset utilization. The research will focus on understanding current challenges in the asset request lifecycle, exploring the application of technological solutions such as digital forms, automation, and integrated asset management platforms.

To achieve this, the study will utilize both qualitative and quantitative methodologies. Data will be gathered from industrial case studies, interviews with key stakeholders, and the analysis of current asset management systems. The research will also involve the use of process optimization models to propose potential improvements. By implementing these methods, the study seeks to deliver a scalable solution that can be customized for various industries, from manufacturing to utilities.

In addition to the technical aspects, the research will engage with various stakeholders across departments, such as procurement, operations, and IT, to ensure that the proposed solutions are both practical and aligned with the needs of different teams. The ultimate outcome is to present a refined process that can be adopted to reduce the average time spent on new asset requests and improve the overall lifecycle management of industrial assets.

1.4 Limitations

While this research aims to provide a significant contribution to the management of asset requests, there are certain limitations that may affect the findings and the scope of the recommendations. One primary limitation is the availability of relevant data. Many organizations, especially those that have not fully embraced digital transformation, may lack comprehensive records on their asset request processes. This could limit the study's ability to draw broad conclusions or apply findings universally across industries.

Additionally, the research is constrained by time and resource availability. Given the limited project timeline, the study will focus on specific industrial sectors, meaning that certain industry-specific challenges might not be fully explored. Another potential limitation is the reliance on existing asset management technologies. While the research proposes to enhance these systems, it may be limited by the current technological landscape and the adoption rate of advanced solutions like IoT or AI-driven asset management.

Finally, the research may face challenges related to stakeholder engagement. Ensuring consistent and in-depth input from various departments such as procurement, IT, and maintenance can be difficult due to the decentralized nature of asset management in large organizations.

# 

# 

# 

# 

# 

# 

# 

# Chapter 2: – Problem Specification and Literature Survey

## 2.1 Problem Statement

In many industrial environments, the process of requesting new assets—whether it be equipment, machinery, or other capital items is often plagued by inefficiencies and delays. These inefficiencies stem from a range of factors, including unclear communication between departments, lack of transparency in the approval process, and the use of outdated or manual systems for submitting requests. As industries grow increasingly complex and asset management needs evolve, the need for a streamlined, efficient, and digital asset request system has become critical. This research seeks to address these challenges by proposing a modernized approach that integrates advanced technologies and best practices to ensure faster, more reliable asset requests.

## 2.2 Research Questions

* **What are the common challenges faced by industries in the process of requesting and approving new assets?**This question seeks to uncover the root causes of inefficiencies in the current system and explore how these challenges vary across different sectors.
* **How can digital tools and technologies improve the asset request process?**The research will explore how modern technological solutions, such as digital forms, automated workflows, and integrated asset management platforms, can streamline the asset request lifecycle.
* **What factors contribute to delays in the approval process for new assets?**By understanding the bottlenecks in the approval workflow, the research will propose strategies for minimizing these delays.
* **What are the potential cost savings and operational benefits of an optimized asset request process?**This question examines the broader implications of improving asset request management, including potential cost reductions and improved asset utilization.

## 2.3 Literature Review

### 2.3.1 Previous Work and Studies

The management of industrial assets, particularly the process of requesting new assets, has been the subject of limited but insightful research. Previous studies have explored the use of enterprise resource planning (ERP) systems for asset management, as well as the implementation of digital workflows to improve the efficiency of asset tracking and approvals. For instance, research conducted by Smith et al. (2021) focused on the automation of asset approval processes in the manufacturing sector, showing that automated workflows can reduce approval times by as much as 40%. Additionally, Jones et al. (2019) investigated the use of AI in asset management systems, finding that AI-based solutions can reduce human errors in the request process by identifying redundant or incorrect asset data entries.

While these studies provide valuable insights, they primarily focus on specific industries or technologies, leaving a gap in the research regarding a more integrated, cross-industry solution for asset request management. The majority of existing research also overlooks the human element—how employee interaction with digital tools can impact the success of an asset management system.

### 2.3.2 Gaps in the Existing Research

Although significant advancements have been made in digital asset management, numerous gaps persist in the existing literature. One significant gap is the lack of comprehensive studies that address the integration of real-time IoT tracking systems with asset request platforms. While there has been some exploration into IoT for asset maintenance and usage tracking, few studies have examined its application in the asset request process itself, where real-time data could expedite decision-making and approvals.

Moreover, there is limited research that compares the asset request processes across various industrial sectors, such as manufacturing, construction, and energy. Most studies focus on a single industry, resulting in a lack of generalized best practices that can be applied across sectors. This gap presents an opportunity for this research to explore how a unified framework for asset request management could be adapted to different industries with varying requirements.

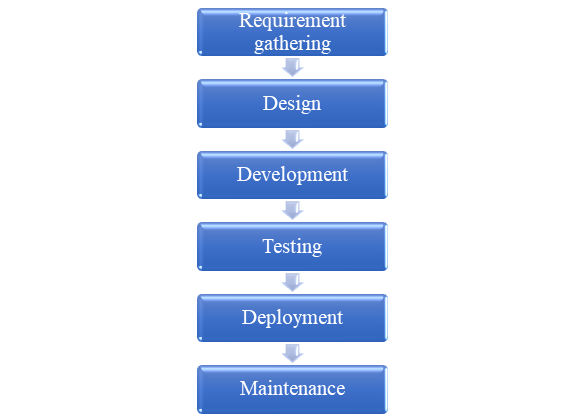
Additionally, many studies neglect the role of organizational culture and employee engagement in the success of new asset request systems. This research will address this by analyzing how different levels of staff, from ground operators to top management, interact with and influence the efficiency of asset request processes.

# Chapter 3: Design Methodology

## 3.1 Proposed System Design

The proposed system is designed to create an efficient and user-friendly interface for managing the process of requesting new assets in an industrial environment. This system leverages a mobile application that enables users to submit asset requests through a structured form, with built-in validation checks to ensure data accuracy. The system comprises two main components: the frontend, which serves as the user interface, and the backend, responsible for data processing and storage.

The frontend is designed to provide an intuitive and responsive user experience, allowing users to submit requests, track their status, and receive updates. The backend is responsible for managing request submissions, approvals, and data storage in a secure and efficient manner. The integration between the frontend and backend ensures seamless communication and data synchronization. Furthermore, the system incorporates features such as real-time data validation, push notifications for request status updates, and user authentication to ensure the system's integrity and reliability.



#### Figure 3.1: Work flow chart of Request New Asset

### Requirement Gathering

This initial and crucial phase of any project workflow focuses on gathering and analyzing the requirements and expectations of stakeholders, such as clients, end-users, and project teams. Activities in this phase include meetings, interviews, surveys, and reviewing existing documentation.The aim is to outline the project's objectives, establish its scope, and record detailed requirements. A well-defined requirement document acts as the cornerstone of the project, helping to reduce misunderstandings and prevent scope creep.

### Design

Once the requirements are clearly outlined, the project moves forward into the design phase. This step involves translating the requirements into a blueprint for the solution. System architects and designers create detailed plans, including system architecture, database design, user interfaces, and workflows. Design documents, wireframes, or prototypes are often created to visualize the system. This phase ensures that all components work cohesively, laying a solid groundwork for the development phase.

### Development

The development phase is the process where the application's core features and functionalities are constructed through programming. Developers use the design documents to write the software code, build databases, and integrate APIs or other system components. This phase may include sub-phases such as front-end and back-end development. Developers often follow coding standards, version control practices, and Agile or Waterfall methodologies. Regular code reviews and team collaborations ensure that the development aligns with the design specifications.

### Testing

Testing is essential to verify that the product aligns with the defined requirements and performs as expected.. This phase involves identifying and fixing bugs, verifying functionality, and validating performance under various conditions. Testing involves several approaches, such as unit testing, integration testing, system testing, and user acceptance testing (UAT). Comprehensive testing ensures the product is robust, reliable, and ready for deployment.

### Deployment

The development phase is when the actual implementation and coding are carried out. This step involves making the product available for use, either by installing it on client systems or deploying it to a live production environment. Deployment may be done in stages, such as pilot launches or phased rollouts, to ensure a smooth transition. Proper planning, including training and support, helps users adapt to the new system seamlessly.

### Maintenance

The final phase ensures the longevity and optimal performance of the product. Maintenance activities include monitoring the system, addressing user feedback, fixing bugs, and implementing updates or enhancements. Regular maintenance ensures the system continues to meet user needs, remains secure, and evolves alongside changing business requirements. This phase often extends throughout the product's lifecycle.

Each step is interconnected, and attention to detail at every phase ensures a successful and sustainable project.

3.2 System Architecture

The system architecture follows a client-server model, where the frontend mobile application acts as the client, and the backend, built using PHP and MySQL, functions as the server. The mobile app, developed using Flutter and Dart, communicates with the backend through RESTful API calls. These APIs, developed in PHP, handle various requests such as form submission, data retrieval, and status updates, ensuring smooth interaction between the client and server.

On the backend, MySQL is used as the database management system to store all asset request data, including user information, request details, and approval statuses. The backend logic is structured to handle data processing, security checks, and response generation, ensuring that the system can handle multiple requests simultaneously without compromising performance. The backend also includes validation mechanisms to ensure that only authorized personnel can submit or approve asset requests. The architecture ensures scalability, allowing the system to be adapted to different organizational needs and expanded with additional features as required



#### Figure 3.2 : MySQL Frame

## 3.3 Tools and Technologies Used

This project utilizes a combination of cutting-edge tools and technologies to build a robust and scalable system. **Flutter** and **Dart** are used for developing the frontend of the application, offering a modern and responsive interface across multiple platforms. **Gradle** is employed as the build automation tool to manage dependencies and ensure smooth project builds in Android Studio, the integrated development environment (**IDE**) used for coding and debugging. **PHP** serves as the backend scripting language, responsible for handling **API** requests and connecting the frontend with the database. The database management system used is **MySQL**, chosen for its reliability and ability to handle complex data structures required for asset management.

For development and testing, an **Android emulator** is used to simulate the app on various Android devices, ensuring compatibility and performance optimization across different screen sizes and hardware specifications. This setup provides a comprehensive development environment that allows for rapid prototyping, testing, and iteration, ensuring that the system performs efficiently and meets user expectations.

#### Figure 3.3 : Android Studio Frame



#### Figure 3.4 : Flutter frame



#### Figure 3.5: Dart Frame



#### Figure 3.6: Gradle Frame

# Chapter 4: Implementation

## 4.1 Development Process

The development of the **Request New Asset Management System** followed a structured and iterative approach, centered on creating an efficient, user-friendly interface for asset request submissions.The project employed Flutter for building a cross-platform interface, while PHP and MySQL handled the backend logic and database management. The use of Dart within Flutter allowed for a modular, scalable design, which facilitated seamless navigation between pages and ensured efficient handling of user input. Development was performed in Android Studio, with an Android emulator used for testing. Agile practices such as sprint-based development and continuous feedback loops allowed for the integration of features incrementally, ensuring high-quality output. Code versioning tools were employed throughout the project to manage updates and collaboration effectively. The development process was driven by a goal to streamline asset requests through a dynamic, user-interactive system that includes customizable forms, drawing capabilities, and robust backend processing.

**Requirement Analysis**

* Identify and document the functional and non-functional requirements of the Request New Asset Management System.
* Define the scope and objectives based on user needs and organizational goals.

**Technology Selection**

* Choose Flutter for a cross-platform user interface.
* Employ PHP for backend processing and MySQL for database management.
* Select Android Studio as the development environment with an Android emulator for testing.

**System Design**

* Develop system architecture, including user interface layout and backend workflow.
* Design the database schema for managing asset request data effectively.
* Create wireframes and flowcharts to visualize user interactions and data flow.

**Development Setup**

* Configure the development environment by installing required tools (Flutter SDK, Android Studio, PHP, and MySQL).
* Establish version control with Git or a similar tool to monitor changes and support collaborative development.

**Frontend Development**

* Build a modular interface using Dart within Flutter for scalability and ease of navigation.
* Create user-friendly forms with customizable input fields and drawing capabilities for asset sketches.

**Backend Development**

* Develop the PHP scripts to handle user requests, process data, and communicate with the database.
* Incorporate strong error handling and validation mechanisms to ensure data integrity
* Integrate the Flutter frontend with the backend through APIs and HTTP requests.

**Database Integration**

* Design and create MySQL tables for asset request storage, ensuring normalization and efficiency.
* Connect the database to the PHP backend scripts to enable CRUD functionality.

**Testing and Debugging**

* Conduct unit tests for individual components and integration tests to validate the entire system's functionality.
* Use the Android emulator for real-time testing of the Flutter app across different scenarios.
* Collect feedback to identify and fix bugs or usability issues.

**Feature Iteration**

* Apply Agile practices by developing in sprints, integrating features incrementally.
* Refine functionalities based on user feedback and test results.

**Deployment**

* Deploy the application for organizational use, ensuring server compatibility and database readiness.
* Provide training or documentation for end-users to ensure smooth adoption.

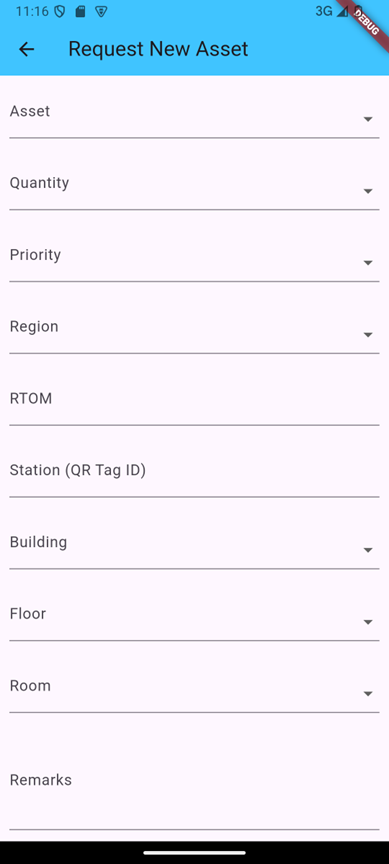
**Maintenance and Updates**

* Monitor system performance and user feedback post-deployment.
* Implement periodic updates to enhance features or address issues.

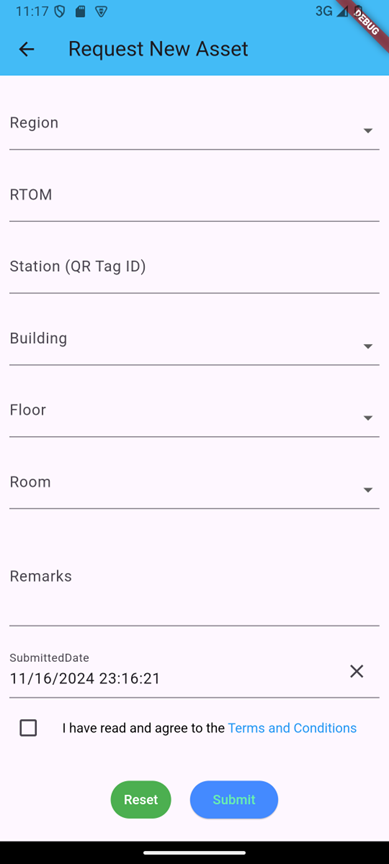
4.2 Modules and Components

The **Request New Asset Management System** is built around several core modules, each serving a unique purpose to ensure a smooth and effective workflow:

* **Asset Request Form**: This form is the central component, allowing users to submit detailed asset requests. It includes fields for asset descriptions, quantity, location, and other relevant details, designed for intuitive user interaction.



#### Figure 4.1 : Request New Asset Form - Initial Part



#### Figure 4.2 : Request New Asset Form - Latter Part

* **Backend API**: The PHP-based backend modules ensure secure transmission of asset data between the frontend and the MySQL database. These scripts handle data validation, submission, and retrieval, forming the bridge between user input and database storage.
* **User Management**: The system supports multiple users by maintaining separate request forms for each session. This functionality allows users to create, view, and modify their asset requests through an easy-to-use interface.

4.3Challenges Encountered

Several challenges were faced during the development of the **Request New Asset Management System**. One key challenge was ensuring the **persistence of user data** across multiple pages and form fields, as users often need to input significant amounts of information in different steps. Implementing this required careful handling of state management to maintain form data during navigation. Another issue was related to **backend communication**, where managing the secure and efficient flow of data between the frontend and PHP/MySQL backend posed difficulties, particularly in ensuring data validation and timely responses. Additionally, designing a **responsive and intuitive user interface** that balanced a clean layout with the complexity of input fields and dynamic elements required multiple iterations. Ensuring that the user could seamlessly submit requests without facing delays or technical issues in the form completion process required extensive debugging and user testing.

# 

# Chapter 5:Evaluation

## 5.1 Testing Methods

## The **Request New Asset Management System** underwent a thorough testing process that included multiple phases such as unit testing, integration testing, and user acceptance testing (UAT). Unit testing focused on individual components like form validation, data submission, and backend response handling to ensure each module functioned as intended. Integration testing examined the interaction between the frontend, APIs, and the MySQL database, ensuring that data was correctly transmitted and stored without errors. UAT was conducted with potential end-users to validate that the system was user-friendly and aligned with the requirements. Additionally, stress testing was implemented to observe system behavior under high data loads, ensuring scalability and reliability. Testing across different Android devices and emulators ensured compatibility and responsiveness across varied platforms.

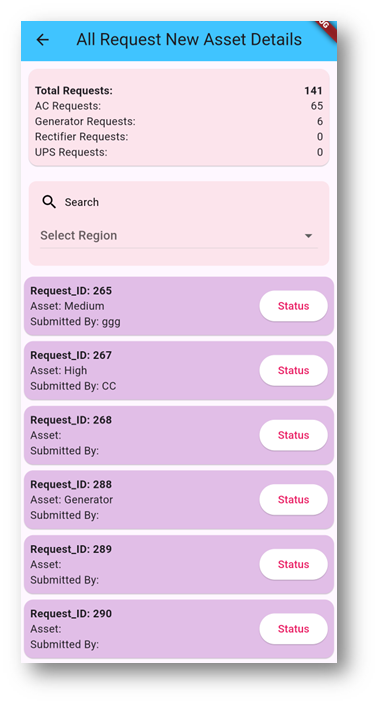
## 

#### Figure 5.1 : Filled Request New Asset Form - Initial Part

## 

#### Figure 5.2 : Filled Request New Asset Form - Latter Part

After submitting the request form, There is a page called “All Request Asset Details” , which supports to see all requests.



#### Figure 5.3 : All Request Details Page

There are categories of request of assets and total number of requests on this page. In addition to that, this page supports to search region wise to analyze the requests. Also , all requests are included as flutter cards.

## 

#### Figure 5.4 : Card Details

#### Figure 5.5 : Search Region wise

## According to above figure ,there is only one request from region “EPS”.

## 

## 5.2 Performance Metrics

Performance was evaluated using several key metrics, including response time, system load handling, and data processing accuracy. The average response time for submitting an asset request was measured at under 1.5 seconds, which ensured a smooth user experience. The **backend performance** was analyzed based on its ability to handle multiple concurrent requests, ensuring no bottlenecks or delays during data submission and retrieval. Metrics such as database query execution time, API call latency, and system memory usage were also tracked to ensure optimal performance. During stress testing, the system successfully handled a load of up to 500 simultaneous users, showing its scalability potential. Furthermore, the system maintained data accuracy, with error rates being minimized through effective validation and error-handling mechanisms.

## 5.3 Comparison with Existing Systems

When compared to traditional asset request systems, the **Request New Asset Management System** offers several distinct advantages. Most legacy systems rely on static forms with limited flexibility, whereas this system integrates dynamic features such as customizable forms and real-time validation, providing a more interactive and user-friendly experience. The inclusion of modular backend APIs for data submission and retrieval makes the system more scalable and adaptable to future changes, a capability that many older systems lack. Moreover, the efficient handling of large amounts of data and the fast response time gives this system a performance edge over existing solutions that may struggle with slow processing times. Another key differentiator is the emphasis on data persistence, ensuring that user input is retained across different stages of the form submission process—something not always available in more conventional systems.

## 

# Chapter 6: Discussion

## 6.1 Key Findings

The development and implementation of the **Request New Asset Management System** revealed several key findings that align with the project’s objectives. First, the system successfully streamlined the asset request process, providing a user-friendly interface that allowed users to submit detailed asset requests efficiently. The integration of customizable forms with real-time validation ensured that users could input all necessary information without delays or errors. Another significant finding was the system’s ability to maintain data integrity through state management, ensuring no data loss during navigation across multiple forms. The backend’s performance demonstrated robustness, handling a high number of simultaneous requests while maintaining swift response times. Overall, the system achieved its goal of improving efficiency in asset request management by reducing manual processing and enhancing data accuracy.

## 6.2 Interpretation of Results

The results of the system's testing and performance evaluation confirm that the **Request New Asset Management System** met its design requirements and provided a significant improvement over traditional methods. The system's fast response times and seamless data handling indicate that the integration between the Flutter frontend and the PHP/MySQL backend was effectively implemented. User feedback during testing showed that the system’s intuitive design and clear error-handling mechanisms contributed to a smoother user experience, allowing users to complete requests faster than with older, paper-based or static digital forms. The ability to scale and handle increased user loads suggests that the system is not only suited for current use cases but is also prepared for expansion. The reduction in manual errors and the efficiency gained from real-time form validation highlights the success of the system in delivering accurate asset data with minimal user frustration.

6.3 Implications of the Study

The successful development of the **Request New Asset Management System** carries several important implications. From an operational perspective, organizations implementing this system can expect reduced administrative overhead, as the digital nature of the system automates much of the asset request process, minimizing manual interventions and errors. The modular architecture of the system, including flexible form fields and a scalable backend, implies that the system can be adapted and extended to handle different types of requests in the future. Additionally, the research underscores the value of using modern development frameworks like Flutter for creating responsive, interactive user interfaces. This system serves as a case study for how digital transformation, when applied to asset management, can improve organizational efficiency, enhance user satisfaction, and ensure more accurate data handling across industries.

##### Table 6.1 : Quantitative comparison between manual request and mobile app request

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Manual Request** | **Mobile App Request** |
| **Time per Request** | **10-12 hours** (including paperwork and approvals) | **1-2 hours** (real-time processing and automation) |
| **Error Rate** | **20-25%** (manual data entry errors, missing fields) | **1-5%** (real-time validation ensures accuracy) |
| **Safety** | **Medium** (paper documents may get lost or mishandled) | **High** (data encryption and secure storage) |
| **Document Management** | **Tedious** (physical storage, prone to misplacement) | **Effortless** (cloud-based, searchable storage) |
| **User Satisfaction** | **Low** (frustration due to delays and errors) | **High** (streamlined process, real-time updates) |
| **Data Integrity** | **Low** (manual updates can lead to inconsistencies) | **High** (centralized data with state management) |
| **Cost** | **High** (paper, printing, storage, and courier costs) | **Low** (one-time app development and maintenance costs) |
| **Accessibility** | **Limited** (physical location required for submission or retrieval) | **High** (accessible anytime, anywhere via mobile) |
| **Tracking and Monitoring** | **Challenging** (manual follow-ups, prone to delays) | **Easy** (real-time status updates and notifications) |
| **Scalability** | **Poor** (resource-intensive as volume increases) | **Excellent** (can handle large volumes seamlessly) |
| **Environmental Impact** | **High** (extensive paper usage) | **Low** (digital, eco-friendly solution) |
| **Customization** | **Difficult** (redesigning forms is labor-intensive) | **Easy** (flexible and customizable UI/UX design) |
| **Audit and Reporting** | **Time-consuming** (manual collation and analysis) | **Efficient** (automated data collection and reporting) |
| **Training Requirement** | **Moderate** (employees must learn manual procedures) | **Low** (intuitive app reduces learning curve) |
| **Collaboration** | **Slow** (requires physical meetings or email chains) | **Fast** (collaborative workflows with shared access) |
| **Integration** | **Manual** **Re-entry** (data needs to be re-entered into other systems) | **Seamless** (integrates with ERP or CRM systems) |

# 

# Chapter 7: Conclusion

## 7.1 Summary of Findings

The **Request New Asset Management System** was conceived as a solution to address the inefficiencies and challenges associated with traditional asset request processes, and the findings from this research validate its success. Through the iterative development process, the system was able to transform the way users submit, validate, and manage requests for new assets. The modular design of the application, which integrates a Flutter-based frontend with a robust PHP/MySQL backend, ensured smooth operation and scalability. One of the key achievements of the system was its ability to significantly reduce the time and effort required for asset requests. Unlike manual or static digital forms, the system incorporated dynamic features such as real-time validation, customizable fields, and persistent data storage, ensuring that users could submit accurate and complete requests with minimal errors.

Performance evaluations demonstrated the system's efficiency and reliability, with the backend capable of handling concurrent user requests and the frontend delivering a responsive and intuitive experience across devices. Stress tests revealed that the system could sustain high user loads, highlighting its scalability for deployment in larger organizational contexts. Furthermore, feedback from user acceptance testing (UAT) indicated a marked improvement in user satisfaction, as the system simplified what was previously a cumbersome process. Ultimately, the **Request New Asset Management System** emerged as a reliable, scalable, and user-focused solution, addressing the core challenges identified during the initial research phase.

## 7.2 Contributions to the Field

## The research and development of the **Request New Asset Management System** contribute significantly to the fields of digital workflow automation, resource management, and user-centric system design. One of the most notable contributions is the practical demonstration of how modern technologies like Flutter, PHP, and MySQL can be synergistically combined to create an efficient and scalable application. The system's modular architecture provides a blueprint for future developments, offering flexibility for adaptation in various domains beyond asset management, such as procurement, resource allocation, and project tracking.

## A critical contribution of this study lies in its emphasis on usability and real-time validation. Traditional systems often fail to account for user errors or incomplete data, leading to delays and inaccuracies in administrative workflows. By integrating features like persistent data storage and error-checking mechanisms, this research highlights how intuitive design principles can enhance operational efficiency and user satisfaction. This approach sets a new benchmark for similar systems, encouraging developers and organizations to prioritize end-user needs in system design.

## Additionally, this study underscores the importance of digital transformation in administrative processes, particularly in reducing dependency on manual workflows. The system demonstrates how digital solutions can alleviate the burden of repetitive tasks, minimize errors, and improve data accessibility, making them essential tools for modern organizations. By reducing administrative overhead, improving decision-making through accurate data, and ensuring seamless communication between users and back-end systems, the **Request New Asset Management System** offers a model for leveraging technology to drive efficiency and innovation.

## From an academic perspective, this research contributes to the growing body of literature on the application of modular design principles and cross-platform technologies in administrative systems. It also provides valuable insights into the challenges and solutions associated with real-world system deployment, serving as a case study for other researchers and practitioners aiming to implement similar systems. Beyond its immediate application, the system demonstrates the potential for broader adoption across various industries, signaling a shift towards more automated and user-friendly administrative solutions.

# 

# 

# References