PERMISSION TRACKING SYSTEM IN PUBLIC ORGANIZATIONS

ABSTRACT

Public organizations face significant challenges in managing permission requests, such as leave, travel, and other official duties. Traditional manual processes are often inefficient, prone to errors, and lack transparency, leading to delays and dissatisfaction among employees and managers. To address these issues, a Permission Tracking System for Public Organizations is proposed.

This web-based system aims to streamline the entire process of managing permissions by proper workflows, enhancing transparency, and improving communication between employees and management. The system will feature a user-friendly interface, role-based access control, real time notifications, and comprehensive reporting and analytics capabilities. It will also include a centralized database for storing permission records, ensuring easy access.

By integrating organizational policies directly into the system, the project ensures consistency in decision-making while maintaining compliance with regulatory requirements. The implementation of this system is expected to reduce administrative overhead, improve decision-making, and provide a more seamless experience for all stakeholders involved.

It represents a significant step towards digitalizing and modernizing administrative processes, fostering efficiency, accountability, and employee satisfaction.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION AND PROBLEM STATEMENT

1.1 Introduction

Public organizations today face significant challenges in managing user access to meet organizational staff. Traditional methods of permission management, often relying on manual processes and spreadsheets, are prone to errors, inconsistencies, and security vulnerabilities. These manual systems lack the necessary features for effective tracking, auditing, and enforcement of access controls. This leads to increased security risks, operational inefficiencies, and difficulties in complying with relevant data protection regulations. Furthermore, the lack of a centralized and automated system hinders accountability and makes it challenging to respond effectively to security incidents. To address these challenges, there is a critical need for an efficient Permission Tracking System (PTS) that can centralize permission management, enhance security, improve operational efficiency, and ensure compliance with regulatory requirements within public organizations.

1.2 Background

Historically, permission management in public organizations has relied on manual processes, such as paper-based records and spreadsheets. These methods are prone to errors, inconsistencies, and security vulnerabilities. The rise of digitalization and the increasing reliance on information technology have further complicated the issue. The need for secure and efficient access control mechanisms has become paramount to protect sensitive data, ensure compliance with regulations, and maintain operational integrity.

1.3 Problem statement

Currently, there is no uniform system for tracking and managing permissions in most public organizations, however many organizations still use traditional paper-based methods or manual spreadsheets to handle these processes. These outdated approaches often result in inefficiencies, delays, and errors, making it difficult to track, manage and approve permissions effectively.

Without system, Organization stakeholders experience prolonged waiting times for approvals,

Therefore, in order to address these challenges, there is a need for implementing a Permission Tracking System that will improve the approval process and ensures efficient management of permissions. This system will provide a structured workflow and secure record-keeping improving decision-making and operational efficiency in public organizations.

1.4 Project objectives

1.4.1 General objectives

General objectives for a project is to design and implement a permission tracking system that will allow approval and monitoring of permission requests within public organizations.

1.4.2 Specific objectives.

- To gather requirements needed for project implementation.
- To design the project architecture.
- To build permission specific functions
- To ensure permission process integrity testing focusing on verifying the ability to handle permission related tasks.

1.5 Significance and scope of project

1.5.1 Significance of the project

- Improve decision making, this is due to faster approvals of request sent to ask a certain permission.
- Saving time, time to travel to a particular organization to just ask permission to do something or meeting to a particular staff will be saved through the use of this system.

- Saving resources, Through the use of this system resources like money for printing forms and paper for writing other documents such as attached latters will be saved and used for other activities.
- Reduce paperwork, this minimizes the use of physical documents, promoting paperless environment and reduce storage costs.
- Facilitate collaboration, collaboration between users may increase due to the fact that they will communicate directly via the system and a requested staff will be allowed to comment on the requested permission e.g., changing date either start or end date.

1.5.2 Scope of the project

The project will focus on designing and developing a system named ‘permission tracking system in public organizations’ that is specifically tailored for use in public organizations. It will be web-based and accessible via an online interface, allowing users to submit requests and approve permissions.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This literature review examines the existing research and literature relevant to the development of a Permission Tracking System within public organization. In most public organizations, managing permissions-grants of access or approval for specific tasks—is a vital aspect of daily operations. These permissions often follow complex workflows involving multiple stakeholders. Traditional methods of managing these workflows, such as paper-based approvals or email chains, are inefficient and prone to human error, resulting in delayed decisions, loss of critical records, and reduced accountability (Smith et al., 2020). A Permission Tracking System aims to address these challenges by providing a centralized platform that streamlines workflows, enhances transparency, and ensures compliance with organizational policies (Garcia et al., 2022). This review highlights existing literature on permission tracking, focusing on its relevance, implementation challenges, and potential solutions for public organizations.

2.2 Topic review

Permission tracking involves the systematic management of requests and approvals to ensure accountability and accurate record-keeping. Information Technology (IT)-based solutions for permission tracking, workflow management, and reporting tools to enhance system efficiency and reliability. Research has consistently highlighted the benefits of these systems. Smith et al. (2020) demonstrate that permission tracking system significantly reduce administrative overhead, improve response times, and ensure all permissions are documented and traceable, enhancing organizational efficiency.

According to Johnson and Lee (2021), public sector institutions require permission management solutions that can handle multi-tiered approval processes and meet regulatory compliance standards, which many standard systems fail to address

2.3 Domain review

Public organizations are governed by stringent policies, regulations, and administrative procedures that make permission tracking particularly challenging. Several factors are critical to understanding the specific needs of permission tracking in this context:

- Hierarchical Approval Processes: Public organizations typically involve multiple approval levels. This multi-level approval process requires sequential approvals from various stakeholders, each evaluating requests based on their responsibilities and expertise. For example, a leave request may need approval from a direct supervisor, followed by departmental head approval, and finally, clearance from the Human Resources department (Smith et al., 2020). A robust PTS must ensure that each approval step is clearly defined, tracked, and escalated if delays occur, preventing bottlenecks (Garcia et al., 2022).
- Regulatory Compliance: Compliance with legal and organizational policies is crucial in public organizations. Permission tracking systems must guarantee secure record-keeping, audit trails, and other features that meet legal requirements (Johnson & Lee, 2021). These systems should allow for transparent audits and ensure that permissions are granted in accordance with regulations and policies (Smith et al., 2020).
- Diverse Stakeholders: Permissions in public organizations may involve various stakeholders, such as employees, department heads, auditors, and external regulatory bodies. A PTS must support seamless collaboration and communication across these groups to ensure effective decision-making and accountability (Garcia et al., 2022)

Advancements in IT have provided solutions to some of these challenges. For instance, Garcia et al. (2022) discuss how permission tracking systems can improve operational efficiency. These technologies can analyze historical data, and identify potential workflow, ultimately enhancing decision-making.

2.4 Problem conclusion

The literature review reveals significant gaps in existing permission tracking systems, particularly within the public sector. Although many systems provide essential functionalities, they often fail to meet the specific needs of public organizations, such as handling complex hierarchical approval structures, ensuring strict regulatory compliance, and managing diverse stakeholder involvement. Furthermore, traditional permission tracking systems often lack advanced tracking and analytics capabilities, limiting their ability to optimize decision-making (Smith et al., 2020).

The proposed project seeks to develop a Permission Tracking System specifically designed for public organizations. This system will incorporate features such as multi-level approval workflows, compliance-focused design, real-time tracking dashboards, and enhanced security measures. By addressing the shortcomings of existing systems, this proposed solution aims to improve workflow efficiency, enhance transparency, and ensure greater accountability in permission management processes within public organizations (Garcia et al., 2022).

CHAPTER THREE

REQUIREMENTS ELICITATION AND SYSTEM ANALYSIS

3.1 Introduction to Requirement Elicitation

Requirement elicitation is a vital phase in the software development lifecycle, wherein the needs and expectations of stakeholders are gathered and thoroughly understood. The process ensures that the final system aligns with user requirements, organizational objectives, and operational constraints. This phase is particularly crucial for the development of the "Permission Tracking System" (PTS) for public organizations, as it defines the necessary functionalities, constraints, and workflows that the system must support to manage and track permissions effectively.

The "Permission Tracking System" is designed to automate and streamline the process of requesting, approving, and managing permissions within public organizations. These permissions, whether related to leaves, system access, or other specific actions, often follow complex approval workflows in hierarchical structures. The system's purpose is to enhance efficiency, ensure compliance, and improve accountability in managing these permissions.

Key functionalities for the system include:

- User Management: Registration, authentication, and role-based access control for users, categorized as employees, managers, and administrators.
- Permission Request Submission: Users can submit permission requests, including details like purpose, duration, and category.
- Approval Workflow: Requests are routed automatically to the relevant approvers based on predefined approval hierarchies.
- Permission Tracking: Real-time tracking of requests, including approvals, denials, and historical logs.
- Reporting and Analytics: Reports generated on usage trends and audit trails for compliance and decision-making purposes.

3.2 Requirement Gathering methods

To understand the diverse needs of stakeholders and ensure the system's design meets their expectations, various requirement elicitation techniques were employed:

- Observation: The current manual processes were observed to highlight inefficiencies and identify potential areas for automation. This process provided valuable insights into the challenges faced by users and administrators.
- Document Analysis: An analysis of existing policies, workflows, and approval hierarchies was carried out to ensure the proposed system complies with organizational standards, legal requirements, and operational needs.
- Interviews: One-on-one interviews were conducted with key stakeholders, including endusers, to identify their needs, expectations, and pain points in the existing manual process of permission handling.

3.3 Functional Requirements

Functional requirements define the core operations the system must perform. For the "Permission Tracking System," these requirements ensure the system meets the needs of stakeholders and facilitates permission management effectively. They include:

- Permission Request Submission and Management: Users should be able to submit, edit, or cancel permission requests with relevant details (e.g., type, duration, purpose).
- Approval Workflow: Requests should be routed through the appropriate approval chain based on predefined hierarchies.
- Status Tracking: The system must provide live updates on the status of requests (approved, rejected, pending).
- Notifications: Notifications for key events such as request submission, approval, rejection, and pending actions should be sent to relevant users.
- Administrator Dashboard: An intuitive dashboard for administrators to manage and monitor all pending and completed requests.

- Reporting & Analytics: Generation of reports that track trends, audit trails, and compliance for organizational oversight.

3.4 Non-Functional Requirements

Non-functional requirements are critical to the system's performance and user experience. These include the operational constraints and system attributes necessary for the effective functioning of the Permission Tracking System:

- Performance: The system should process and display request statuses in under two seconds to ensure smooth user interactions.
- Scalability: The system must handle up to 1,000 concurrent users without a degradation in performance, ensuring support for a growing user base in public organizations.
- Security: The system must ensure data confidentiality, integrity, and secure access through encryption and multi-factor authentication.
- Usability: The system should have an intuitive user interface, accessible to users with minimal training or technical expertise.
- Reliability: The system must maintain an uptime of at least 99.9%, ensuring continuous availability and minimizing downtime.
- Compliance: The system must adhere to relevant organizational and legal standards regarding data privacy, auditability, and regulatory compliance.

3.5 System Analysis

3.5.1 Analysis of current system

The current system for managing permissions in public organizations predominantly relies on manual processes such as paper-based forms, emails, and in-person approvals. While this approach may have sufficed in the past, it is prone to errors and inefficiencies. Common issues associated with the current system include lost requests, delayed approvals, and a lack of transparency, making it difficult to track and manage historical data. The absence of automation in these processes contributes to delays and administrative overhead, which hinders the overall operational efficiency.

3.5.2 Analysis of proposed system

The proposed "Permission Tracking System" aims to overcome the limitations of the current manual system by automating and digitizing the entire workflow. This transition is expected to result in several key benefits, including:

- Faster and More Efficient Handling: Automated workflows will ensure quicker processing and reduce the time spent on approvals and follow-ups.
- Enhanced Transparency and Accountability: Real-time tracking of requests, along with a detailed audit trail, will ensure all actions are logged, enabling accountability at each stage of the workflow.
- Reduced Administrative Burden: HR and management teams will experience a reduction in the time spent on manual tracking, filing, and responding to requests.
- Improved User Experience: Streamlined workflows and automated notifications will enhance the user experience, ensuring that requests are handled quickly and efficiently.

3.5.3 System requirements

The system requires the following specifications to operate efficiently:

- Hardware Requirements: The system will require dedicated servers, adequate storage capacity, and backup systems to ensure seamless data processing and retrieval.
- Software Requirements: The system will be web-based, built using robust frameworks and technologies. It will rely on reliable database management systems (DBMS) and APIs for integration with other internal systems.
- Network Requirements: The system will require stable and reliable internet connectivity to ensure uninterrupted access, communication, and data processing across multiple locations.

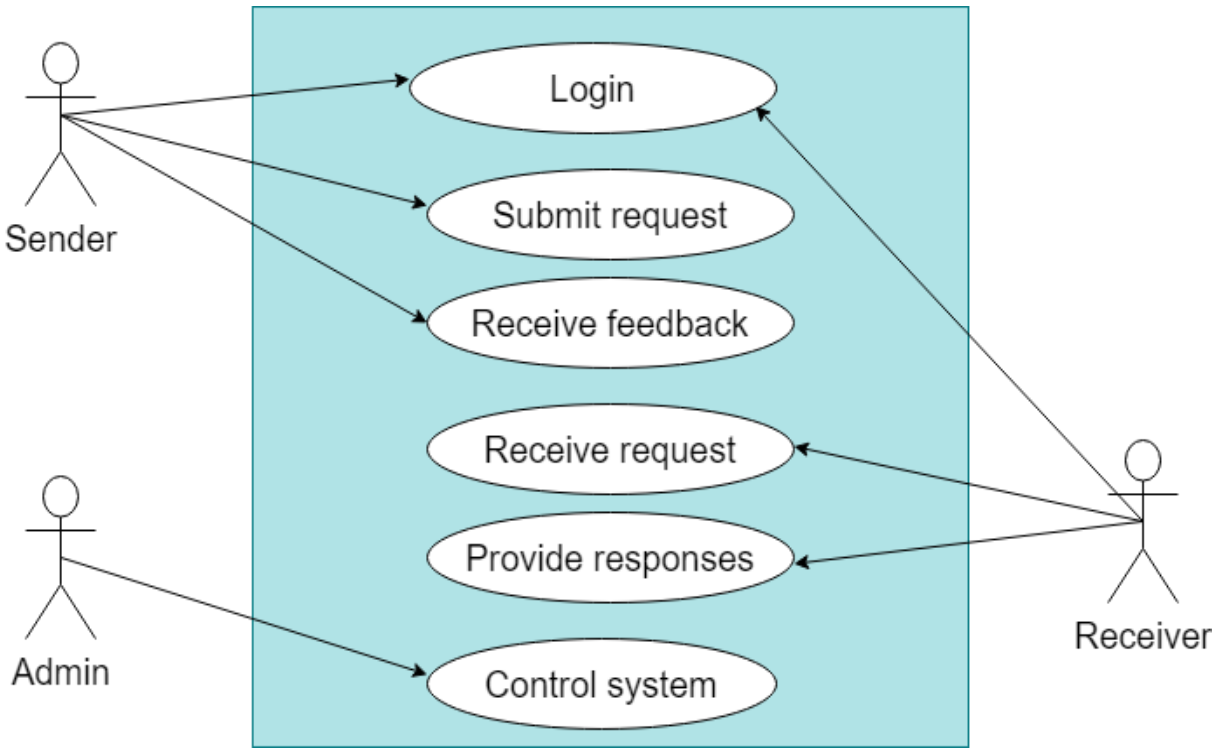

Figure 1: Use case diagram for a project

Where by;

- Sender represents all Applicants
- Receiver represents all approvers (OfficerInCharge)

CHAPTER FOUR

SYSTEM DESIGN

4.1 Introduction to System Design

System design is defined as the process of creating the architecture for different components, interfaces, and modules of the system, and providing the data necessary for implementing such elements so that they meet end-user requirements. It is a multidisciplinary process involving trade-off analysis, balancing conflicting requirements, and making design decisions that will impact the overall system.

This phase acts as an intermediate bridge between system analysis and system implementation in the system development life cycle. The purpose of system design is to produce sufficiently detailed data and information about the system elements to enable implementation consistent with the system architecture's models and views.

In this permission tracking system, the design covers system methodology, database design, and user interface design.

The design is categorized into:

- Architectural Design: Defining the overall structure, components, relationships, and behavior of the permission tracking system. It serves as a conceptual framework for implementation.
- Logical Design: Providing a conceptual representation of system functions and behavior, independent of technologies, showing the logical structure, processes, and rules of the system.
- Physical Design: Transforming the logical design into a concrete implementation using specific technology, platforms, and infrastructure to ensure the system meets performance and deployment needs.

4.2 System Design Methodology and Technologies

A systematic methodology is vital to define data structures, their dependencies, their orderings, and their attributes, with internal consistency verified for correctness before moving to implementation. Optimization is then applied to transform these structures into executable forms.

4.2.1 Agile model

The Agile methodology was chosen for developing the Permission Tracking System because it allows iterative, incremental development. Agile supports collaboration with stakeholders, continuous improvements, and flexibility to changing requirements.

Agile Phases:

- Planning
- Iterative development
- Continuous feedback
- Release

Figure 2: Agile development life cycle

4.2.2 Advantages of methodology

- Flexibility and Adaptability: Agile supports dynamic requirements and ongoing improvements.

- Customer Satisfaction: Stakeholders are engaged frequently, resulting in a product that meets their expectations.
- Early and Continuous Delivery: Delivering working modules regularly enables validation and user acceptance.
- Improved Quality: Testing and reviews are performed early and continuously.
- Risk Mitigation: Risks are identified and addressed early due to short iterations.
- 4.2.3 Technologies and tools used

  The following technologies will be used for the Permission Tracking System:

- SQLite (via BD Browser): A lightweight, opensource relational database used for storing permission requests, user records, approval and related data. It is easy to use, requires no server setup, and well suited for local or small-scale systems.
- Microsoft Windows: The primary OS platform for development and testing.
- HTML: For structuring the web pages and forms of the permission system.
- CSS: For styling and presenting the system's interfaces.
- Python (Django Framework): Python is high-level, readable, and secure, making Django suitable for the web-based permission system.

4.3 Database Design

The Permission Tracking System uses a SQLite database (managed via DB Browser) to maintain records of applicants, permission requests, officer assignments, and approval histories.

Key Features:

- Managing permission request submissions
- Tracking approval status
- Recording officer feedback
- Reducing redundancy by linking requests to user IDs

Entity-Relationship Diagram

This shows the relationships among entities.

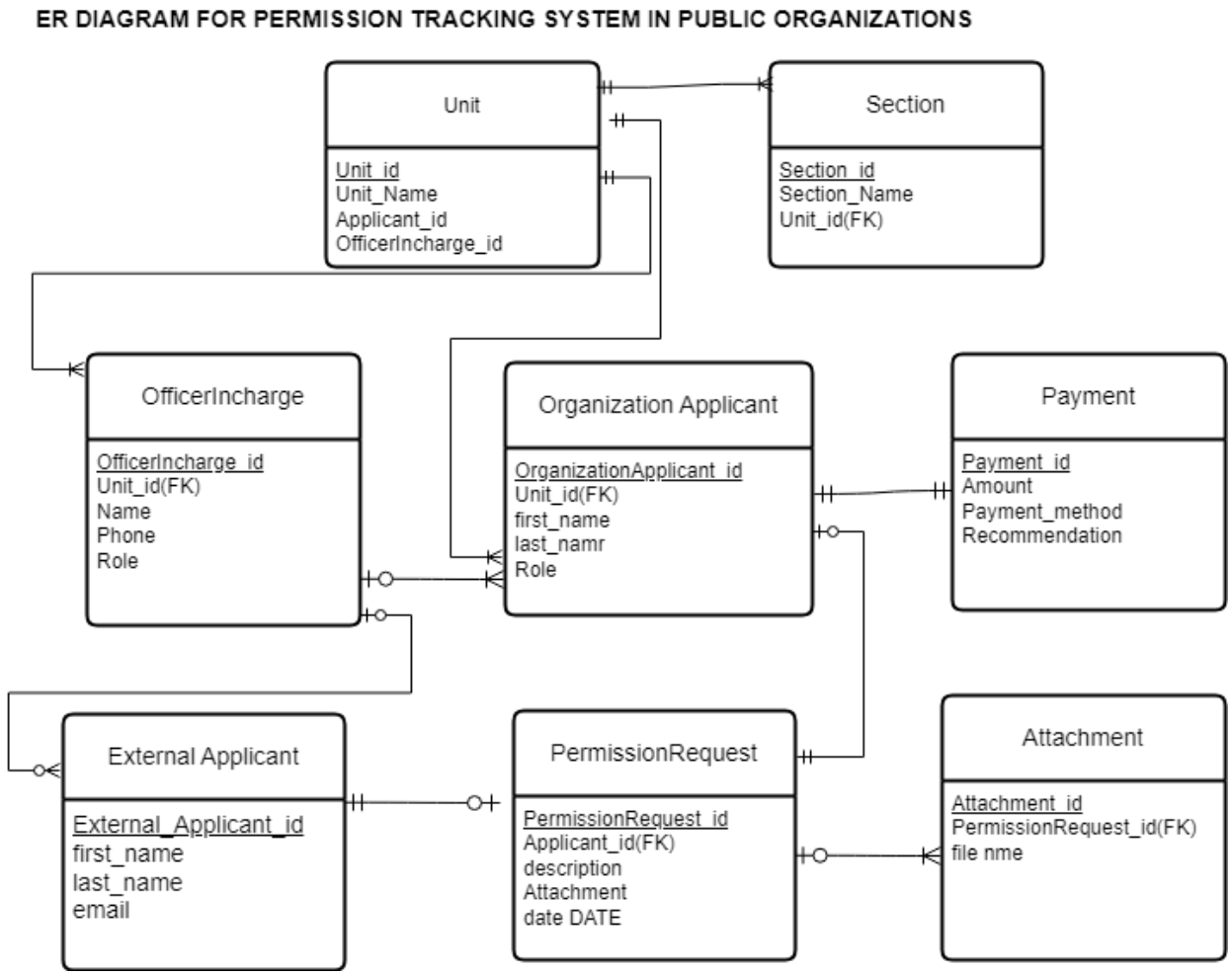

Figure 3: ER diagram for a Project

4.4 User Interface

The user interface design focuses on usability, clarity, and easy navigation.

Layout and Navigation:

- Consistent menu and sidebars for navigation
- Modular views for different user roles
- Easy search and filter tools

Responsive Design:

- Works on desktops, tablets, and smartphones

Forms and Validation:

- User registration
- Login
- Permission request submission
- Validation to catch missing or invalid entries

4.4.1 Activity diagram

- An activity diagram is a type of UML diagram that models the dynamic aspects of the system by showing the flow of control or data from activity to activity. It is especially useful for visualizing business processes, workflows, or the logic of complex use cases in the Permission Tracking System.

  In this system, the activity diagram shows how applicants, officers, and admins interact with the system through a sequence of activities.

  Applicant Activities

- Register an account
- Log in
- Submit permission request
- Track status
- Receive approval/rejection notification

  Officer Activities

- Log in
- View assigned permission requests
- Approve/reject or forward requests

- Provide feedback
- Log out

Admin Activities

- Configure user roles
- Manage user accounts
- Oversee submitted permission requests

The activity diagram will clearly illustrate transitions between these activities, decisions (for example, *approved vs. rejected*), and parallel actions (e.g., multiple officers processing requests simultaneously).

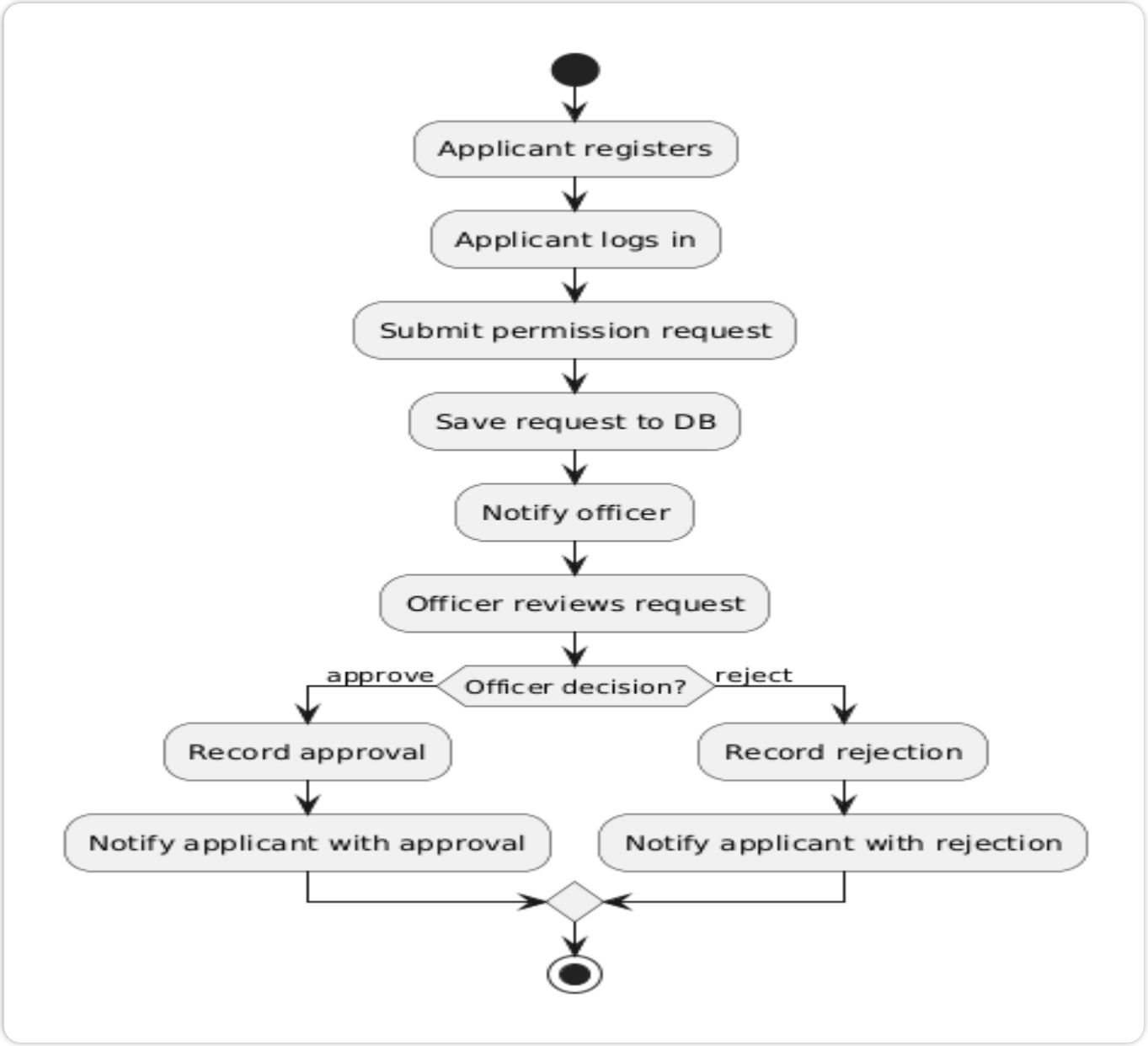

Figure 4: Activity diagram for a project

4.4.2 Sequence diagram

A sequence diagram is another UML behavioral diagram that focuses on how objects in the system interact with each other over time. It captures the time ordering of messages passed between system components.

In the Permission Tracking System, a sequence diagram describes the interactions among the applicant, the system, the officer, and the admin during a permission request process.

Key interactions include:

- Applicant registers and logs in
- Applicant submits a permission request
- The system stores the request and notifies the appropriate officer
- Officer retrieves and reviews the request
- Officer approves, rejects, or forwards the request
- The system records officer feedback
- Admin reviews overall requests and may intervene in final approvals

The sequence diagram helps to:

- Clarify how different actors exchange data
- Show the order of operations
- Identify any missing interactions or timing issues in the process

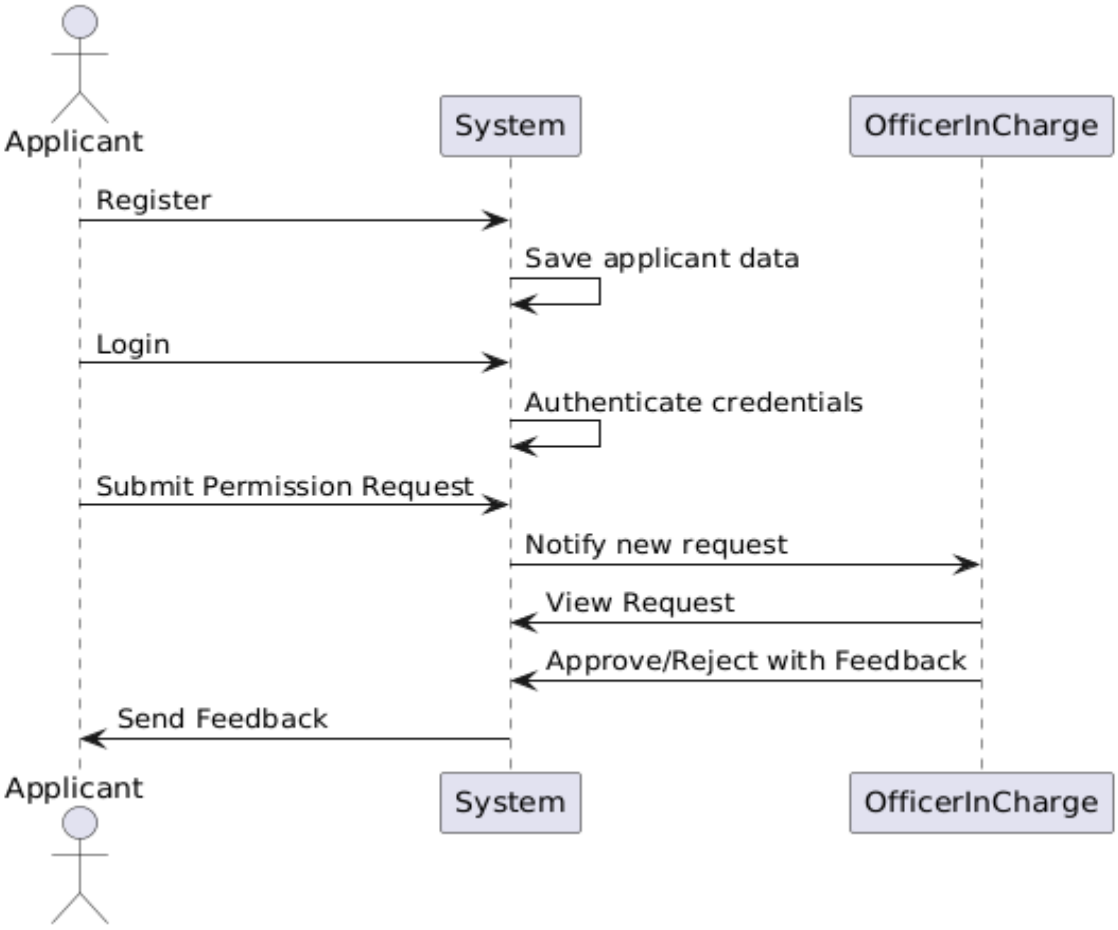

Figure 5: Sequence diagram for a project

CHAPTER FIVE

SYSTEM IMPLEMENTATION

5.1 Introduction to System Implementation

System implementation transitions the Permission Tracking System from design to actual deployment. It includes installing software, configuring environments, and ensuring users can operate the system with minimal disruption.

The goal is to automate permission request workflows, simplify approvals, and enhance transparency within a public organization.

Functional Requirements

1. User Registration and Authentication

- The system shall allow applicants to register with personal details.
- The system shall authenticate registered users during login using secure credentials.

2. Permission Request Submission

- Applicants shall be able to submit new permission requests with required details such as destination, dates, and purpose.
- Applicants shall be able to attach supporting documents (attachments) to each request.

3. Request Routing and Feedback

- The system shall route submitted permission requests to the designated OfficerInCharge for review based on department or role.
- OfficersInCharge shall receive notifications of new requests.
- Applicants shall receive confirmation of submission.

4. Request Approval Workflow

- OfficersInCharge shall be able to view, approve, or reject permission requests.
- OfficersInCharge shall be able to add feedback upon approval or rejection.

- The system shall notify applicants of the final decision.

5. Feedback Management

- OfficersInCharge shall provide feedback which is stored and viewable by applicants.
- Feedback shall include request details, officer comments, and status updates.

6. Security

- The system shall maintain an audit trail for all permission requests, approvals, feedback, and blacklisting actions.
- Only authorized users shall access, modify, or approve requests.

5.2 Development Tools

The Permission Tracking System was developed using the following tools and technologies:

- HTML (HyperText Markup Language)

  Used to define the structure and layout of web pages, ensuring content is semantically organized and accessible.

- CSS (Cascading Style Sheets)

  Utilized for styling the user interface, providing a visually appealing and responsive design across devices.

- Python (Django Framework)

  Django, a high-level Python web framework, was chosen for its rapid development features, built-in security, and scalable architecture. Django handles the server-side logic, routing, authentication, and data models of the system.

- DB Browser for SQLite

  Used as a graphical interface for managing and inspecting the SQLite database during development. It enables developers to view tables, run SQL queries, and manage data more efficiently.

- Adobe photoshop

  It used for desining of some images before inserted to the system.

5.3 Database Implementation

In the permission tracking system in public organization project, I have chosen to implement an SQLite database as the backend storage . The database configuration is defined in the DATABASES setting within the Django project's settings file.

```
82  DATABASES = {
83      'default': {
84          'ENGINE': 'django.db.backends.sqlite3',
85          'NAME': BASE_DIR / 'db.sqlite3',
86      }
87  }
```

Figure 6: Project's database implementation

- ENGINE: This parameter specifies the database engine or backend that Django uses to interact with the database. In this configuration, 'django.db.backends.sqlite3' is specified, indicating that the application is using the SQLite database engine. SQLite is a lightweight, file-based database that does not require a separate server process, making it well-suited for development and small-scale production use.

- NAME: This parameter defines the name (or path) of the database file. In this case, it is set to BASE_DIR / 'db.sqlite3', meaning that Django will create and use a database file named db.sqlite3 located in the project's base directory.

  Unlike other database systems such as MySQL or PostgreSQL, SQLite does not require additional parameters like USER, PASSWORD, HOST, or PORT since it operates directly on a local file. This makes it simpler to set up and ideal for quick deployment or prototyping.

5.4 User Interface Implementation

Login and Registration:

- Secure authentication for applicants, officers, and admins

- Registration with verification

Figure 7: User login and Registration

Dashboards:

- Admin Dashboard: manage users and permissions

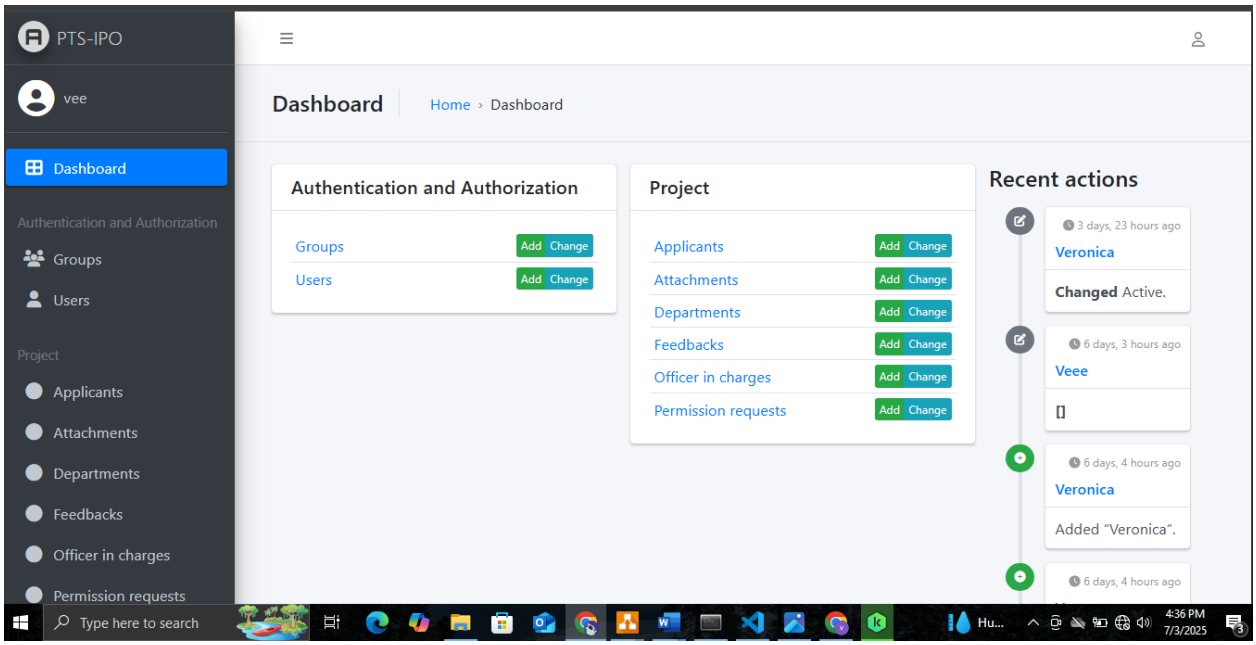

Figure 8: Admin dashboard

- Officer Dashboard: review and act on requests

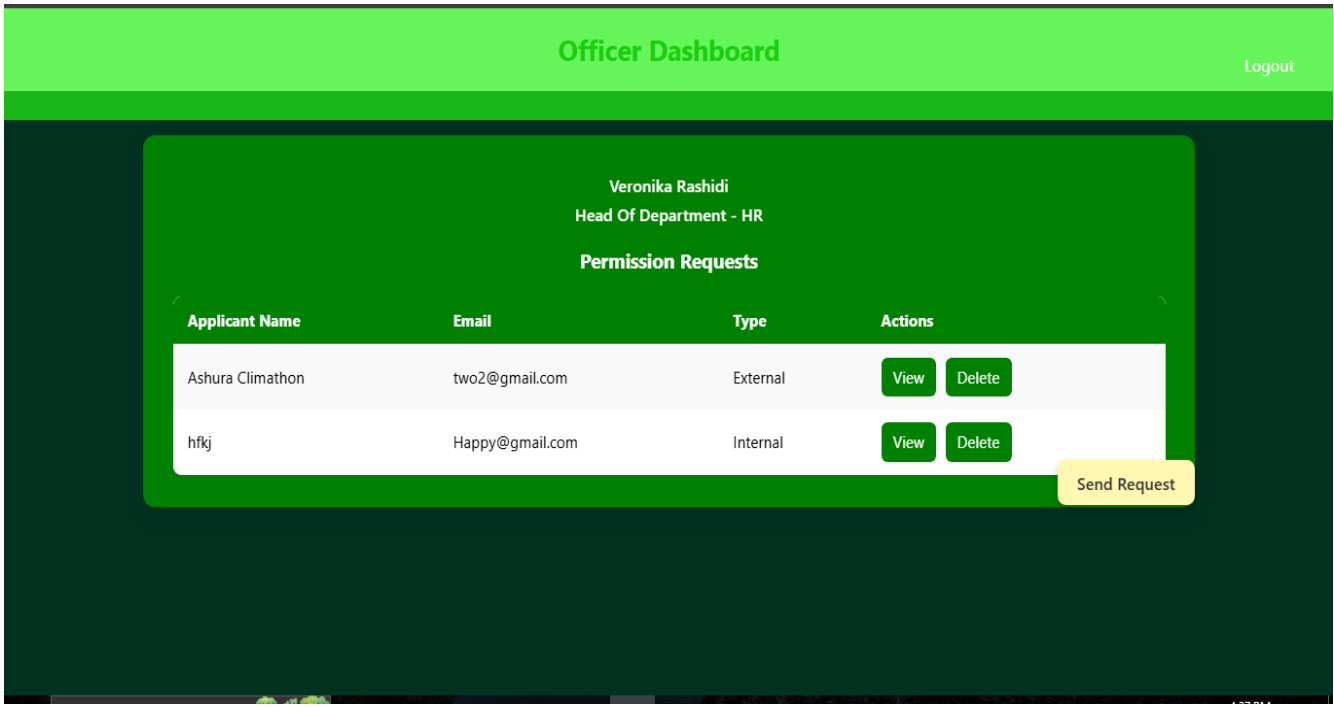

Figure 9: OfficerIncharge dashboard

- Applicant Dashboard: submit and track requests

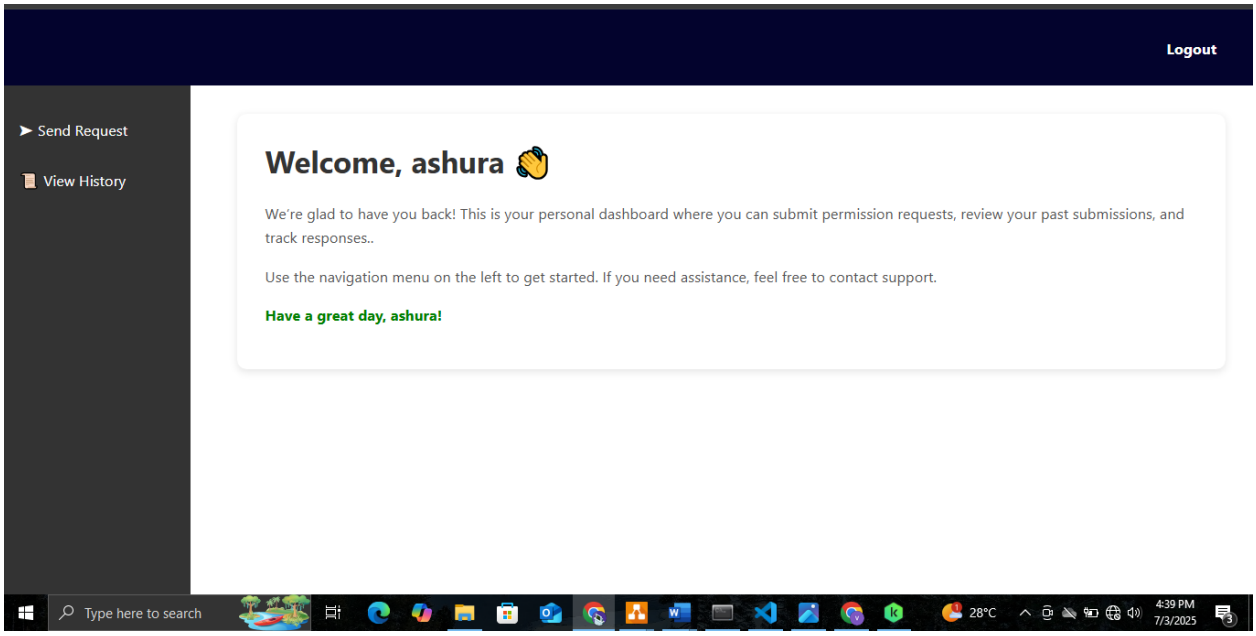

Figure 10: Applicant dashboard

5.5 System Testing and Evaluation

5.5.1 System testing

The system testing was done throughout the development process in all phases to guarantee error free and fulfil system functionalities all as planned. System testing can be of two types unit testing and system testing.

❖ Unit testing

Unit testing was performed on each module or block of code during development of the system to ensure that functionalities are meet. Here we can test a small unit like users try to login into the system by providing wrong information

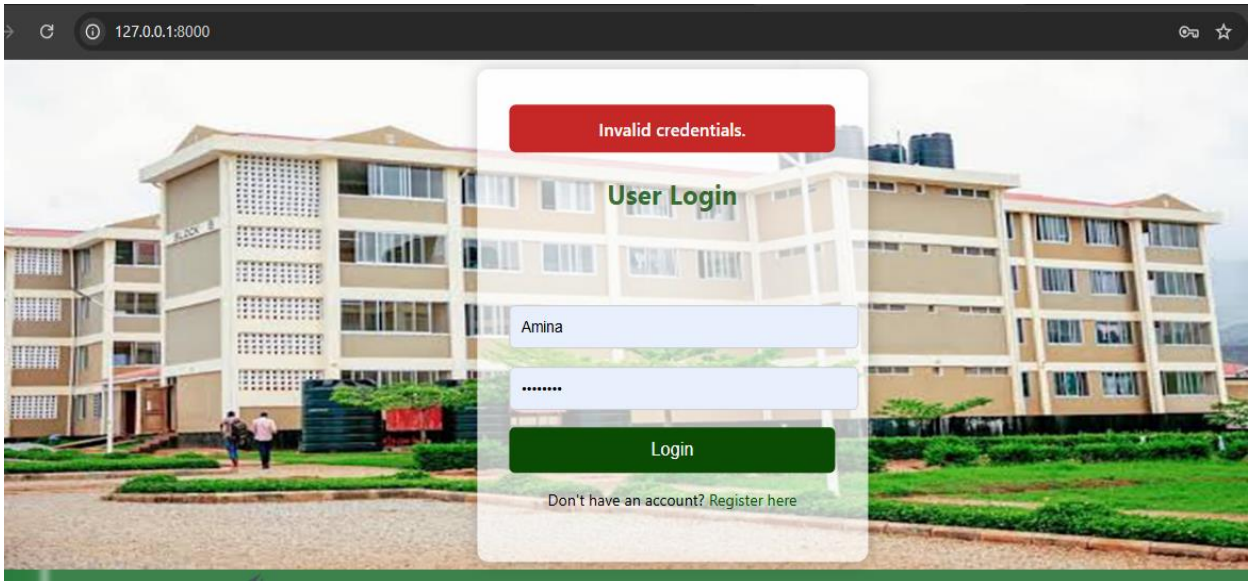

Figure 11: Unit testing

System testing is openly related to the system design phase. System tests inspection the entire system ability and the transmission of the system under improvement with externally systems.

Functional Testing:

- Verify request submission
- Verify approval and feedback
- Validate user logins

5.5.2 System evaluation

- Usability: Tested by staff and applicants
- Performance: Checked for response time and concurrency
- Functionality: Evaluated by supervisors and user representatives

CHAPTER SIX

CONCLUSION AND RECOMMENDATIONS

6.1 Summary of Achieved Objectives

The Permission Tracking System in Public Organizations has successfully achieved several key objectives;

- Firstly, it receive the permission request, allowing both internal and external applicants to submit their requests, thereby eliminating the need for physical document submission and reducing delays.
- Secondly, the system enabled officers to efficiently track, review, and take action on submitted requests, with features for approval, rejection, and forwarding, including the ability to provide direct feedback to applicants. This improved transparency and accountability across departments.
- Additionally, the system facilitated better communication between applicants and officers by providing status updates and feedback mechanisms.
- Finally, by reducing manual paperwork, the system minimized the administrative burden on public organization staff, contributing to a more efficient and digitized public service environment.

6.2 Conclusion

Despite the challenges faced in implementing complex routing logic and securing active engagement from management during development, the Permission Tracking System has made significant strides toward achieving its objectives. It has modernized the way permission requests are handled in public institutions by replacing manual processes with a streamlined, digital platform. While some advanced features—such as full integration with national SMS or email APIs—were not implemented due to external constraints, the system effectively addressed major user needs. These include simplifying request submission, improving response time from officers, and ensuring applicants can track the status of their requests. As a result, the system has enhanced operational efficiency, accountability, and service delivery in public organizations.

6.3 Recommendations

To further improve the system, it is recommended to simplify the routing and approval logic to enhance usability and reduce system errors. This involves evaluating and optimizing the decision paths taken by requests across various departments. Engaging top-level management and institutional leadership is also essential to ensure organizational buy-in, resource support, and wider adoption of the system. Since API integration with formal national communication platforms proved difficult, alternative solutions such as local SMS gateways or WhatsApp Business API should be explored to support real-time notifications. Moreover, priority should be given to enhancing the feedback and request history tracking features, which are crucial for building trust and accountability between applicants and decision-makers. Lastly, the system can be expanded to support other organizational workflows such as leave applications, event approvals, or logistics coordination, to maximize its impact and relevance within the publicsector.

REFERENCES

Fernandez, D. J., & Fernandez, J. D. (2008). Agile project management.

Garcia, L., Perez, R., & Watson, J. (2022). *The Role of Machine Learning in Streamlining Permission Tracking Systems*. Journal of Public Administration Technology, 12(4), 157-175.

Griffee, D. T. (2005). Interview data collection.

Johnson, T., & Lee, M. (2021). *Public Sector Compliance and Permission Management: Addressing Regulatory Requirements in Digital Systems*. Public Management Review, 18(3), 220-235.

Smith, A., Brown, J., & Davis, S. (2020). *Optimizing Permission Tracking in Organizations: A Case Study Approach*. Journal of Information Systems, 45(1), 34-50

Sommerville, I. (2011). *Software Engineering* (9th ed.). Pearson Education.