**HAWASSA UNIVERSITY**

**Institute of Technology**

**School of Informatics**

**Department of Information systems**

****

**Web Based Maintenance Tracking System For CBE Hawassa District**

**Industrial Project Documentation**

**Advisor: Mr. Ayele S.**

**Group Members ID Number**

**1. Abdulhafiz Muhammed………………………..….0091/14**

**2. Robel Mohammed………………………………...2612/14**

**3. Melat Tigistu………………………………………2093/14**

**4. Eskender Niguse…………………………………..1197/14**

**Chapter one**

**Introduction to the Maintenance Tracking System for the Commercial Bank of Ethiopia**

The banking industry is increasingly reliant on technology to enhance operational efficiency, maintain customer satisfaction, and ensure smooth day-to-day activities. Banks utilize web technologies to improve the efficiency of their operations and deliver enhanced services to customers. Web-based applications facilitate remote monitoring, reporting, and real-time communication among staff, enabling swift decision-making and problem resolution. Technologies such as web portals and management dashboards empower bank administrators to track and maintain equipment inventory, schedule repairs, and analyze data trends to improve service quality. Web technologies also provide a collaborative platform where multiple stakeholders—such as system administrators, technicians, and branch managers—can interact seamlessly. For CBE, implementing a web-based maintenance tracking system ensures that all relevant actors can access and update information anytime and anywhere, which is vital for maintaining continuous service.

One critical area in this digital transformation is the maintenance and tracking of essential bank equipment such as desktops, printers, and other technical assets. The proper functioning of these assets is crucial for the seamless delivery of banking services. The Maintenance Tracking System for the Commercial Bank of Ethiopia (CBE), Hawassa region district, is designed to streamline the process of managing and monitoring the repair and upkeep of branch equipment. This system serves as an essential tool for optimizing maintenance workflows, reducing equipment downtime, and ensuring that service delivery is uninterrupted.

**Background of the Project**

The Commercial Bank of Ethiopia (CBE) is the largest and most established financial institution in Ethiopia, with a history dating back to its establishment in 1942. As of recent data from **combanketh.et**, CBE serves millions of customers with a comprehensive range of banking services, including savings and current accounts, loans, digital banking solutions, and corporate financial services. The bank operates an extensive branch network, with over 1,700 branches throughout the country, making it a key player in supporting economic activities and financial inclusion in Ethiopia ("About Us," **combanketh.et**).

CBE has positioned itself at the forefront of technological adoption in the Ethiopian banking sector. It has been a pioneer in introducing modern banking services such as mobile and internet banking, ATMs, and POS services to meet the evolving needs of its customers. This focus on digitalization ensures that CBE remains competitive and capable of providing reliable services in an increasingly technology-driven environment.

The IT and Operations divisions of CBE are tasked with managing and maintaining the bank's technical infrastructure. This department is responsible for ensuring the optimal performance of IT equipment, such as desktops, printers, and networking devices, which are crucial for daily banking operations. The team conducts regular maintenance, handles technical troubleshooting, and coordinates with external vendors for specialized repairs. Implementing a web-based maintenance tracking system will support these responsibilities by offering an organized platform for managing repair requests, scheduling preventive maintenance, and tracking service completion. This initiative aligns with CBE's commitment to leveraging technology for enhanced operational efficiency and service delivery ("Services," **combanketh.et**).

**Statement of the Problem**

In large financial institutions such as the Commercial Bank of Ethiopia (CBE), maintaining the optimal performance of technical equipment across numerous branches is essential for efficient service delivery. However, the current reliance on manual or semi-automated methods for managing equipment maintenance poses significant challenges. These challenges include slow response times, ineffective communication between departments, difficulties in tracking maintenance history, and a lack of centralized data management.

The absence of an integrated maintenance tracking system leads to extended equipment downtime, disrupting branch operations and negatively impacting customer satisfaction. Technicians face difficulties in managing their workload efficiently due to the absence of a streamlined process for task assignment and status tracking. The current approach does not adequately monitor which technician is handling specific repair tasks, making it difficult to verify completed work and trace repair histories. This lack of transparency contributes to operational inefficiencies and miscommunication.

Additionally, technician shift management is poorly structured. This can lead to technicians being either overworked or underutilized, causing delays in repair times and an uneven distribution of tasks. Proper shift management is essential to ensure timely and effective handling of repair activities.

there also significant issues related to inventory management. The absence of an effective system to track the availability of accessories and spare parts means that managers and technicians often do not have a clear understanding of stock levels. This can lead to delays when sourcing necessary components, further prolonging repair times and affecting service continuity.  
**objective**   
**General Objective**

The general objective of this project is to develop a web-based maintenance tracking system for the Commercial Bank of Ethiopia (CBE) Hawassa District.

**Specific Object**

* Develop a centralized platform for tracking and managing maintenance activities within CBE.
* Develop methods to accurately record and verify technician assignments and completed repairs.
* Create an effective approach for optimizing technician schedules and managing workload distribution.
* Implement real-time reporting features for maintenance status updates.
* Develop an integrated system for tracking inventory of spare parts and accessories.

**5. Scope of the Project**

The project will cover the development and deployment of a web-based maintenance tracking system for CBE Hawassa District. This system will include:

* Management of maintenance requests.
* Technician task assignments and real-time status updates.
* Technician shift scheduling.
* Inventory tracking for spare parts and accessories.
* Storing and accessing detailed maintenance history.

**Limitations**

The proposed maintenance tracking system has some limitations that may affect its implementation and usage:

* **Dependence on Internet Connectivity:** The system requires stable internet access to function effectively, which may be a limitation in areas with poor connectivity.
* **Scalability:** The system may need further enhancement to scale effectively with future expansions of CBE’s operations or additional features.
* The system will not include direct integration with external vendors.
* It will not cover maintenance management for non-technical equipment.

**Significance of the Project**

* Enhances operational efficiency and service delivery by streamlining maintenance request handling.
* Reduces equipment downtime, enabling uninterrupted services for customers.
* Improves communication and coordination among branches and departments.
* Promotes accountability with transparent task assignments and progress tracking.
* Optimizes inventory management to ensure spare parts and accessories are readily available, minimizing delays.
* Facilitates better workload distribution and shift management for technicians, reducing fatigue and improving response times.
* Supports data-driven decision-making through comprehensive reporting and analysis.
* Contributes to higher operational standards, increased customer satisfaction, and efficient resource utilization within CBE.

**Methodology  
Data Collection Methodology**:

* **Interviews**: Conducted with IT staff and branch managers to understand current processes, identify gaps, and gather user expectations for the system.
* **Observation**: On-site observation of existing workflows to capture practical insights into how maintenance tasks are currently managed.
* **Document Review**: Analysis of existing maintenance records and operational manuals to understand the current documentation and procedures.
* **System Data Collection**: Insights from a former intern who worked on related data collection projects to provide an additional understanding of existing system data.

**System Analysis and Design Methodology**

The project will utilize a combination of **Object-Oriented System Analysis and Design (OOSAD)** and the **Waterfall Model** for system development. OOSAD will provide a structured approach to analyzing and designing the system using objects that represent real-world entities, ensuring modularity, reusability, and scalability of the system components. The **Waterfall Model** will guide the overall project phases, including requirement analysis, system design, implementation. his combined approach ensures a thorough understanding of user requirements and a clear development pathway.

The **Waterfall Model** was chosen for the development of this project for the following reasons:

* **Clarity of Requirements**: The project requirements are well-defined and understood, making a linear development approach suitable.
* **Structured Process**: The step-by-step nature of the Waterfall Model ensures that each phase is completed before moving on to the next, providing clear progress tracking.
* **One-Time Delivery**: The project aims for a one-time, complete delivery, fitting the Waterfall Model's sequential approach.

**System Requirements**

**Software Requirements**

* **XAMPP**: A server solution to test the system by running the PHP code and MySQL database locally.
* **MySQL**: Used for creating and manipulating databases to store and manage maintenance records.
* **Microsoft Word 2019**: Utilized for documenting project-related information and reports.
* **Visual Studio Code**: The primary code editor for writing the web application’s code.
* **Programming Languages**: PHP, JavaScript
* **Draw.io**: Employed for creating UML diagrams to illustrate system architecture and interactions.

**Hardware Requirements**:

* **Processor**: Dual-core 1.8 GHz or higher.
* **Memory**: Minimum 4 GB RAM.
* **Display**: Minimum resolution of 1366x768.

**Feasibility Study**

**Economic Feasibility** The development of a maintenance tracking system for the Commercial Bank of Ethiopia (CBE) is economically feasible. The potential benefits of implementing this system include reduced operational costs, improved resource management, and minimized equipment downtime. By automating and centralizing the maintenance tracking process, the bank will see long-term financial gains through:

* **Reduced Paperwork Costs**: Transitioning from paper-based and Excel-based tracking to an automated system will save on printing and storage expenses.
* **Increased Efficiency**: Streamlining repair and maintenance processes will enhance technician productivity, reducing time spent on administrative tasks.
* **Cost Savings on Equipment Repairs**: By efficiently managing repair schedules and technician assignments, the bank can prevent further damage to equipment, thereby reducing repair and replacement costs.
* **Optimized Inventory Management**: Real-time tracking of spare parts and accessories can prevent unnecessary purchases and ensure that resources are used effectively.

Overall, the initial development and implementation costs are justified by the anticipated long-term savings and operational improvements.

**Technical Feasibility** The project is technically feasible given the available technology and resources at CBE. The bank’s existing IT infrastructure can support the development and deployment of the proposed system. Key factors supporting technical feasibility include:

* **Available Development Tools**: The use of technologies such as PHP, MySQL, HTML, CSS, and JavaScript aligns with the skill set of the development team and ensures compatibility with existing systems.
* **IT Infrastructure**: CBE’s current network and server capabilities are sufficient for hosting a web-based tracking system with potential scalability for future needs.
* **Supportive Expertise**: The IT department at CBE possesses the knowledge required to maintain and support the new system post-implementation.
* **Security Measures**: Adequate cybersecurity protocols and encryption methods can be integrated to ensure data protection and secure transactions.

**Operational Feasibility** Implementing the new system is operationally feasible within CBE’s existing structure. The proposed maintenance tracking system will enhance daily operations and meet user requirements effectively. The factors supporting operational feasibility include:

* **User Acceptance**: Technicians, branch managers, and other stakeholders will benefit from an intuitive, user-friendly interface that simplifies their tasks.
* **Training and Transition**: Training sessions can be organized to familiarize staff with the system, ensuring a smooth transition from current practices.
* **Improved Accountability**: The system’s ability to assign and track technician responsibilities will address the current issues of accountability and workload distribution.
* **Operational Efficiency**: Automating processes will reduce manual work, allowing staff to focus on higher-value activities, thus increasing overall productivity.

**Budget Estimation**

The budget for developing the web-based maintenance tracking system includes costs related to software, hardware and other necessary components. The estimation is broken down into the following categories:

|  |  |  |
| --- | --- | --- |
| **Category** | **Item** | **Cost (Birr)** |
| **Software** | XAMPP (server solution) | Free |
|  | MySQL (database) | Free |
|  | Draw.io | Free |
|  | Vistal studio code | Free |
| **Hardware** | Laptop | 40000 |
|  | Mouse | 800 |
|  | Flash drive | 650 |
| **Miscellaneous** | Internet Data | 6000 |
| **Total** |  | 47450 |

### ****Scheduling****

Scheduling is a critical part of project management to ensure that all tasks are completed within the allocated time. For this project, scheduling has been organized into distinct phases to ensure a systematic workflow. Each phase has a defined duration to track progress and ensure timely delivery. Below is the schedule:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **November 25** | **January 25** | **February 5** | **May 14** |
| Requirement Analysis | **████████** |  |  |  |
| System Design |  | **████████** |  |  |
| Development Phase |  |  | **█████████████** |  |
| Submission of Implemented System |  |  |  | **████████** |

**Chapter two**

**Description of the Existing System**

**Introduction to the Existing System**

The existing maintenance management system at the Commercial Bank of Ethiopia, Hawassa region district, operates manually, lacking a structured and centralized platform to handle maintenance activities efficiently. Task assignment to technicians is performed informally, often based on immediate availability or ad hoc decisions. This random approach fails to consider critical factors such as technician expertise, workload, or proximity to the branch, leading to delays and ineffective task execution.

Moreover, the system has no mechanism to track the availability or usage of spare parts required for repairs. Technicians often face delays due to the unavailability of necessary components, which could have been prevented with proper inventory tracking. The absence of a spare parts management system also hampers inventory planning, resulting in frequent disruptions or overstocking.

Maintenance requests, task assignments, and updates are documented inconsistently across various formats, such as handwritten notes, emails, or phone calls. This fragmented process leads to challenges in tracking progress, retrieving historical data, and ensuring accountability. Communication between branch managers and technicians relies on informal channels, causing miscommunication and delays in task updates.

The system also lacks an effective monitoring or reporting mechanism to assess technician performance, track task completion, or identify recurring equipment issues. Tasks are often prioritized without considering urgency or the criticality of the equipment, resulting in inefficient use of resources and prolonged downtimes for essential operations. Additionally, there is no formal escalation mechanism for delayed or unresolved tasks, which further exacerbates operational inefficiencies.

Overall, the current system is time-consuming, unreliable, and incapable of meeting the bank's growing operational demands. These limitations necessitate the development of a more efficient and structured maintenance management solution.

**Description of the Proposed System**

The proposed system is a web-based Maintenance Tracking System designed to resolve the inefficiencies and challenges of the existing manual process. By introducing automation and centralized management, it ensures that maintenance tasks are handled efficiently and transparently. The proposed system provides a structured task assignment mechanism, allowing the system administrator to allocate tasks based on predefined criteria such as the technician’s expertise, workload, and proximity to the branch. This eliminates the random assignment issues of the existing system and ensures timely and effective task execution.

To address the lack of spare parts tracking, the system integrates an inventory management module that monitors the availability and usage of spare parts. This feature ensures that technicians are equipped with the necessary components before arriving at the branch, preventing delays caused by unavailability. The system also supports proactive inventory planning by generating alerts for low stock levels, thus reducing disruptions during repair activities.

All maintenance requests and updates are managed within a centralized database, eliminating the fragmentation of information across various formats. This centralization ensures consistency, improves accessibility, and supports better record-keeping for future reference. The system also includes real-time communication tools, providing branch managers, technicians, and system administrators with timely updates on task progress. Automated notifications ensure that all stakeholders remain informed throughout the process.

The proposed system prioritizes tasks based on urgency and equipment criticality, ensuring high-priority issues are resolved promptly. Additionally, it includes a robust reporting and monitoring module, enabling administrators to evaluate technician performance, analyze task completion rates, and identify recurring issues. The system also features an escalation mechanism that flags delayed or unresolved tasks, ensuring timely action by higher authorities to prevent prolonged downtimes.

By automating routine processes and structuring workflows, the proposed system significantly reduces the time and effort required for maintenance management. It supports better utilization of resources and contributes to operational efficiency. Furthermore, the system maintains a comprehensive record of all maintenance activities, providing valuable historical data for trend analysis, performance evaluation, and predictive maintenance planning.

In summary, the proposed system directly addresses the existing system's challenges by introducing efficiency, transparency, and accountability. It transforms the maintenance management process into a streamlined and reliable operation, ensuring continuous support for the bank's activities.

**Strengths of the Existing System**

The existing manual maintenance tracking system has the following strengths:

* **Flexibility**: The system allows for informal and quick communication to report maintenance issues.
* **Low Cost**: Since it relies on basic tools like emails, phone calls, and handwritten notes, it incurs minimal operational costs.
* **Simplicity**: The system is straightforward to use and does not require extensive technical knowledge or training.
* **Personal Interaction**: The manual approach fosters direct communication between branch managers and technicians, which can help in resolving certain issues quickly.

**Weaknesses of the Existing System**

Despite its strengths, the existing system suffers from several critical weaknesses:

* **Random Task Assignment**: Tasks are allocated without considering technician expertise, workload, or proximity, leading to inefficiencies and delays.
* **No Spare Parts Tracking**: There is no system to monitor spare parts availability or usage, causing delays and poor inventory management.
* **Fragmented Information**: Maintenance records are stored in unstructured formats such as emails and handwritten notes, making retrieval and tracking difficult.
* **Communication Gaps**: Informal communication channels often result in inconsistent and unclear updates on task progress.
* **Lack of Monitoring and Reporting**: There is no system to monitor technician performance, track task completion rates, or identify recurring issues.
* **No Escalation Mechanism**: Delayed or unresolved tasks cannot be formally escalated for quicker resolution, causing prolonged downtimes.
* **Dependency on Key Individuals**: The system relies heavily on specific personnel, and their absence can disrupt the entire process.
* **Inefficient Prioritization**: Tasks are not prioritized effectively, often resulting in delays for critical maintenance needs.
* **Time-Consuming Processes**: Manual logging and tracking of tasks are slow, delaying the overall maintenance cycle.

**Chapter Three: System Features**

**Functional Requirements**

The functional requirements outline the key operations that the Maintenance Tracking System must perform to fulfill the needs of all stakeholders.

**Maintenance request**

* send maintenance requests from branch to district
* Give feedback on maintained devices

**Admin Dashboard Overview**

* Display key metrics: devices under maintenance, spare parts availability, and completed tasks.
* Activity log to track actions like parts usage, device registrations, and report generation.

**User and Team Management**

* Add or remove employees, assign roles, and manage team shifts.
* View team performance.

**Spare Parts Management**

* Manage spare parts inventory with details like quantity and item type
* Track usage history, including technician, parts used, and associated devices.
* Generate low-stock alerts for timely spare parts reordering.

**Maintenance Tracking**

* View the status of maintenance tasks: Working, Pending, and Maintained.
* Access detailed device history, including past issues, repairs, and parts used.

**Report Generation**

* Generate customizable reports based on filters like date range, technician, device type, or branch.

**Notifications and Alerts**

* Receive alerts for critical updates, such as low spare parts inventory or task completion.
* Receive maintenance requests

**Technician Dashboard**

* View shift overview, task list.

**Device Management**

* Register new devices for maintenance with details like device type, issue, and branch location.
* Update device status (Working, Pending, Maintained) in real-time.

**Non-Functional Requirements**

The non-functional requirements outline the quality attributes, performance criteria, and operational constraints of the proposed Maintenance Tracking System.

**Performance**

* The system must operate efficiently, providing quick responses to user actions and handling multiple tasks simultaneously without delays.

**Availability**

* The system must ensure consistent access and operation, minimizing downtime to meet user needs at all times.

**Reliability**

* The system must function dependably with minimal interruptions, ensuring accurate data processing and task management.

**Usability**

* The system interface must be user-friendly and intuitive, allowing users with varying technical expertise to navigate and perform tasks easily.

**Security**

* The system must protect data through secure access controls and encryption, ensuring confidentiality and preventing unauthorized access.

**Compatibility**

* The system must work seamlessly across common platforms and devices, including different web browsers and hardware setups.

**Use Case Model**

A use case is an interaction between users and a system. It captures the goal of the users and the responsibility of system to its users. It is the functionality of the system or the service provided by the system.

**Actor Identification**

Actors are external entities that interact with the system. They represent the different types of users, systems, or other entities that engage with the system to perform specific tasks or achieve certain goals.

In our project, we have identified 3 actors.

1. Administrator
2. Technician
3. Branch manager

* **Administrator**

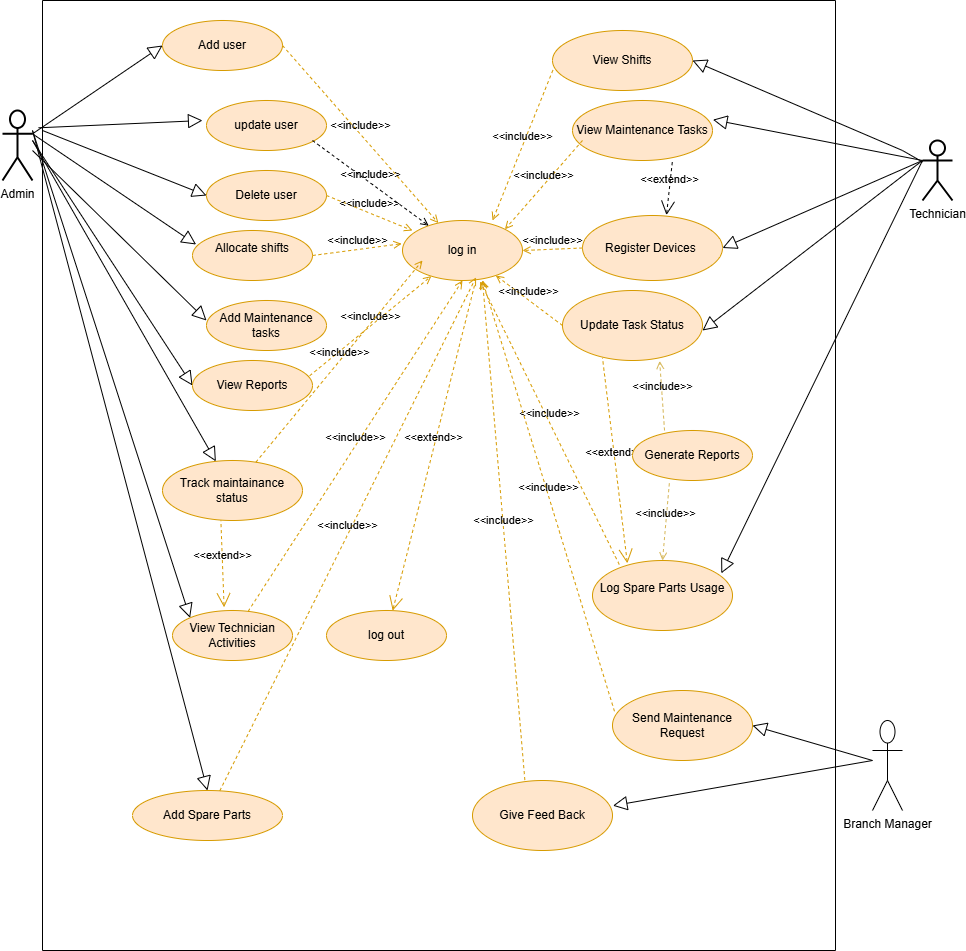
The System Administrator is a high-level user responsible for managing and maintaining the entire Maintenance Tracking System. This role ensures that the system operates smoothly, assignments are distributed effectively, and security protocols are in place.

* **Technician**

A Technician is an employee responsible for performing maintenance and repair tasks on the bank's equipment, such as computers, printers, and other devices. Technicians are skilled personnel who diagnose issues, replace faulty parts, and ensure that devices are working efficiently.

* **Branch Manager**

The Branch Manager, responsible for overseeing the operations of a specific branch, can leverage this system to send maintenance requests , offer valuable feedback on completed maintenance tasks.

****

**Use Case Description**

Table 1: log in

|  |  |
| --- | --- |
| **Use Case ID** | 001 |
| **Use Case Name** | Login |
| **Actors** | All Users |
| **Description** | Users log into the system to access their assigned tasks and functionalities. |
| **Trigger** | User enters login credentials. |
| **Preconditions** | 1. User must be registered in the system.  2. Login page must be accessible. |
| **Normal Flow** | 1. User enters username and password.  2. System authenticates credentials.  3. System grants access to the user. |
| **Postconditions** | 1. User is logged into the system. |
| **Alternative Flows** | 1. Invalid credentials: System displays an error message and prompts for re-entry. |
| **Exceptions** | None |
| **Priority** | High |
| **Frequency of Use** | Daily |
| **Business Rules** | Only registered users can log in. |
| **Special Requirements** | Passwords must be encrypted for security. |
| **Assumptions** | User has valid login credentials. |
| **Notes and Issues** | N/A |

Table 2: log out

|  |  |
| --- | --- |
| **Use Case ID** | 002 |
| **Use Case Name** | Logout |
| **Actors** | All Users |
| **Description** | Users log out from the system after completing their activities. |
| **Trigger** | User clicks the "Logout" button. |
| **Preconditions** | 1. User is logged into the system. |
| **Normal Flow** | 1. User selects "Logout" option.  2. System ends the session and redirects the user to the login page. |
| **Postconditions** | User is logged out of the system. |
| **Alternative Flows** | None |
| **Exceptions** | None |
| **Priority** | High |
| **Frequency of Use** | Daily |
| **Business Rules** | Users must log out when their session is complete. |
| **Special Requirements** | Ensure all session data is securely cleared during logout. |
| **Assumptions** | User is active in their session before logout. |
| **Notes and Issues** | N/A |

Table 3: view shifts

|  |  |
| --- | --- |
| **Use Case ID** | 003 |
| **Use Case Name** | View Shifts |
| **Actors** | Technician |
| **Description** | Technicians view their daily shifts. |
| **Trigger** | Technician logs in to check assignments. |
| **Preconditions** | 1. Technician is logged in.  2.shifts are already assigned. |
| **Normal Flow** | 1. System displays assigned tasks and shifts for the day. |
| **Postconditions** | Technician accesses their schedule. |
| **Alternative Flows** | None |
| **Exceptions** | None |
| **Priority** | Intermediate |
| **Frequency of Use** | Daily |
| **Business Rules** | Shifts must be accurately assigned in the system. |
| **Special Requirements** | System must provide real-time updates for any shift changes. |
| **Assumptions** | .Shifts are accurately assigned |
| **Notes and Issues** | N/A |

Table 4: Register Devices

|  |  |
| --- | --- |
| **Use Case ID** | 004 |
| **Use Case Name** | Register New Devices for Maintenance |
| **Actors** | Technician |
| **Description** | Technicians register newly identified devices that require maintenance into the system. |
| **Trigger** | A new device issue is identified. |
| **Preconditions** | 1. Technician is logged in. 2. Maintenance requests are recieved |
| **Normal Flow** | 1. Technician selects "Device Management" option.  2. Technician enters device details.  3. System logs the new device. |
| **Postconditions** | Device is registered in the system with all relevant details. |
| **Alternative Flows** | None |
| **Exceptions** | 1. Duplicate device entry: System alerts the technician. |
| **Priority** | Intermediate |
| **Frequency of Use** | Weekly |
| **Business Rules** | Devices must be uniquely identifiable in the system. |
| **Special Requirements** | System must validate device details before registration. |
| **Assumptions** | Technician has accurate information about the new device. |
| **Notes and Issues** | N/A |

Table 5:Update Task Status

|  |  |
| --- | --- |
| **Use Case ID** | 005 |
| **Use Case Name** | Update Task Status |
| **Actors** | Technician |
| **Description** | Technicians update the status of their assigned maintenance tasks (e.g., Pending, Working, Maintained). |
| **Trigger** | Technician starts or completes a maintenance task. |
| **Preconditions** | 1. Technician is logged in.  2.There are available tasks |
| **Normal Flow** | 1. Technician selects the "Update Task" option.  2. Technician updates the status.  3. System saves the updated status. |
| **Postconditions** | Task status is updated in the system. |
| **Alternative Flows** | None |
| **Exceptions** | 1. Task not found: System alerts the technician. |
| **Priority** | High |
| **Frequency of Use** | Daily |
| **Business Rules** | Task status must reflect the actual progress of the maintenance task. |
| **Special Requirements** | System must log the timestamp of the status update. |
| **Assumptions** | Technician has access to the system during the task. |
| **Notes and Issues** | N/A |

Table 6: log spare part usage

|  |  |
| --- | --- |
| **Use Case ID** | 006 |
| **Use Case Name** | Log Spare Parts Usage |
| **Actors** | Technician |
| **Description** | Technicians record the usage of spare parts during maintenance tasks. |
| **Trigger** | Spare parts are used for maintenance. |
| **Preconditions** | 1. Technician is logged in.  2. Spare parts are available in inventory.   3.technician has used the spare part |
| **Normal Flow** | 1. Technician selects " Spare Parts" option.  2. Technician enters details of the spare part and task.  3. System updates the inventory. |
| **Postconditions** | Spare part usage is recorded, and the inventory is updated. |
| **Alternative Flows** | 1. Spare part not available: System displays an alert. |
| **Exceptions** | None |
| **Priority** | Intermediate |
| **Frequency of Use** | Weekly |
| **Business Rules** | Spare parts inventory must be updated in real-time. |
| **Special Requirements** | System should generate low-stock alerts when thresholds are reached. |
| **Assumptions** | Technician accurately logs the details of used spare parts. |
| **Notes and Issues** | N/A |

Table 7**:**Report Maintenance Issues

|  |  |
| --- | --- |
| **Use Case ID** | 007 |
| **Use Case Name** | Report Generation |
| **Actors** | Technician |
| **Description** | Technicians report about his completed maintenance tasks |
| **Trigger** | Maintenance tasks are added |
| **Preconditions** | 1. Technician is logged in.  2. Maintenance task is completed. |
| **Normal Flow** | 1. Technician selects "Report Generation" option.  2. Technician describes the issue.  3. System logs the issue for review. |
| **Postconditions** | report is recorded and for review. |
| **Alternative Flows** | None |
| **Exceptions** | None |
| **Priority** | High |
| **Frequency of Use** | As needed |
| **Business Rules** | All reports must be addressed promptly. |
| **Special Requirements** | System must notify administrators of reports. |
| **Assumptions** | Technician has the necessary permissions to report issues. |
| **Notes and Issues** | N/A |

Table 8**:**Manage Users

|  |  |
| --- | --- |
| **Use Case ID** | 008 |
| **Use Case Name** | Manage Users |
| **Actors** | Administrator |
| **Description** | Administrators can add, remove, and update user accounts, including assigning roles and permissions. |
| **Trigger** | Administrator initiates user management tasks. |
| **Preconditions** | 1. Administrator is logged in.  2. User management interface is accessible. |
| **Normal Flow** | 1. Administrator selects "User Management"  2. Administrator performs desired actions (add, remove, or update users).  3. System saves changes. |
| **Postconditions** | User accounts and roles are updated successfully. |
| **Alternative Flows** | None |
| **Exceptions** | 1. User not found: System displays an error. |
| **Priority** | High |
| **Frequency of Use** | Weekly |
| **Business Rules** | Each user must have a unique ID and appropriate role permissions. |
| **Special Requirements** | System must validate user information before updates. |
| **Assumptions** | Administrator has the necessary permissions for user management. |
| **Notes and Issues** | N/A |

Table 9**:**Track Maintenance Status

|  |  |
| --- | --- |
| **Use Case ID** | 009 |
| **Use Case Name** | Track Maintenance Status |
| **Actors** | Administrator |
| **Description** | Administrators monitor the progress of ongoing maintenance tasks across branches. |
| **Trigger** | Administrator initiates a status check. |
| **Preconditions** | 1. Administrator is logged in.  2. Task statuses are updated by technicians. |
| **Normal Flow** | 1. Administrator selects "Maintenance Tracking."  2. System displays current task statuses. |
| **Postconditions** | Administrator views the latest maintenance task statuses. |
| **Alternative Flows** | None |
| **Exceptions** | None |
| **Priority** | High |
| **Frequency of Use** | Daily |
| **Business Rules** | Task status should reflect real-time updates from technicians. |
| **Special Requirements** | System should highlight overdue tasks. |
| **Assumptions** | Task data is updated regularly. |
| **Notes and Issues** | N/A |

Table 10**:**View Technician Activities

|  |  |
| --- | --- |
| **Use Case ID** | 010 |
| **Use Case Name** | View Technician Activities |
| **Actors** | Administrator |
| **Description** | Allows administrators to monitor the activities of technicians, such as task completion, parts usage, and performance stats. |
| **Trigger** | Administrator initiates the activity monitoring feature. |
| **Preconditions** | 1. Administrator is logged in.  2. Technician activity data is updated in the system. |
| **Normal Flow** | 1. Administrator selects "View Technician Activities."  2. System displays technician activity logs and performance metrics. |
| **Postconditions** | Administrator gains insights into technician productivity and task performance. |
| **Alternative Flows** | None |
| **Exceptions** | Activity logs unavailable: System notifies the administrator. |
| **Priority** | Intermediate |
| **Frequency of Use** | Weekly |
| **Business Rules** | Logs should provide up-to-date and accurate information. |
| **Special Requirements** | System should generate graphical insights for better understanding. |
| **Assumptions** | Technician activities are recorded in real time. |
| **Notes and Issues** | N/A |

Table 10**:Add spare parts**

|  |  |
| --- | --- |
| **Use Case ID** | 011 |
| **Use Case Name** | Add Spare Parts |
| **Actors** | Administrator |
| **Description** | Administrators can view, update, and manage spare parts inventory, including reordering low-stock items. |
| **Trigger** | Spare parts inventory requires updates or replenishment. |
| **Preconditions** | 1. Administrator is logged in.  2. Spare parts inventory is up to date. |
| **Normal Flow** | 1. Administrator selects " Spare Parts managements"  2. System displays current inventory levels.  3. Administrator chooses item type.   4.Administrator writes amount of item  5.administrator add the spare |
| **Postconditions** | Spare parts inventory is updated, and low-stock items are reordered. |
| **Alternative Flows** | None |
| **Exceptions** | Inventory update fails: System displays an error. |
| **Priority** | High |
| **Frequency of Use** | Weekly |
| **Business Rules** | Spare parts quantities must reflect actual stock. |
| **Special Requirements** | System should alert administrators for critical low stock. |
| **Assumptions** | Inventory data is accurate. |
| **Notes and Issues** | N/A |

**Table 11 Maintenance Request**

|  |  |
| --- | --- |
| **Use Case ID** | UC-012 |
| **Use Case Name** | Submit Maintenance Request |
| **Actors** | Branch Manager |
| **Description** | Describes how the Branch Manager submits a maintenance request for a device requiring attention. The request includes device details and the issue it is facing. |
| **Trigger** | The Branch Manager identifies a device requiring maintenance. |
| **Preconditions** | - The Branch Manager must be logged into the system.  - The device requiring maintenance must be available for registration. |
| **Normal Flow** | 1. The Branch Manager navigates to the "Maintenance Request" section.  2. Fills out the details   3. Submits the request to the Administrator. |
| **Postconditions** | - The Administrator is notified of the maintenance request.  - The request is logged in the system for tracking and future reference. |
| **Alternative Flows** | - If required fields are missing (e.g., serial number), the system prompts the Branch Manager to complete them. |
| **Exceptions** | - System downtime or connectivity issues prevent submission.  - Invalid device information causes rejection. |
| **Priority** | High |
| **Frequency of Use** | Frequent (as device issues arise regularly) |
| **Business Rules** | - Maintenance requests must be complete and accurate.  - Requests should be acknowledged within 24 hours. |
| **Special Requirements** | - User-friendly interface with clear instructions on required fields.  - Validation of device information. |
| **Assumptions** | - The Branch Manager is familiar with the system interface.  - Device information is updated and accurate. |
| **Notes and Issues** | none |

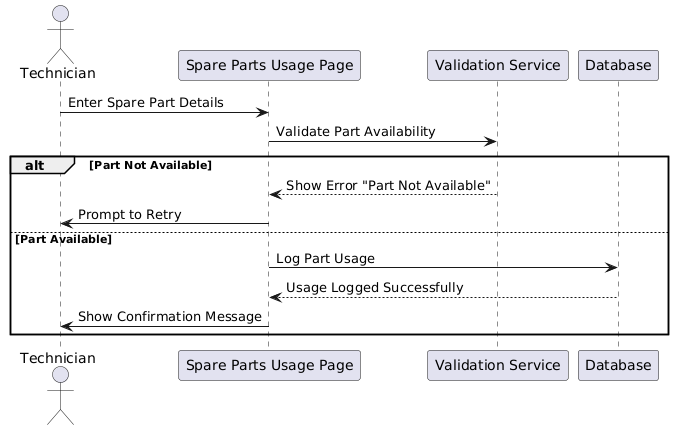
**Table 12: feedback**

|  |  |
| --- | --- |
| **Use Case ID** | UC-013 |
| **Use Case Name** | Provide Feedback |
| **Actors** | Branch Manager |
| **Description** | Describes how the Branch Manager provides feedback on completed maintenance tasks, including maintenance quality and technician performance. |
| **Trigger** | A maintenance task for a branch device is completed. |
| **Preconditions** | - The Branch Manager must be logged into the system.  - The maintenance task must be completed for a branch device. |
| **Normal Flow** | 1. The Branch Manager navigates to the "Feedback" section.  2. Selects the completed maintenance task from the list.  3. Provides feedback on quality and technician performance.  4. Submits the feedback. |
| **Postconditions** | - Feedback is logged in the system.  - Feedback is available for reports and technician performance evaluation. |
| **Alternative Flows** | - If the feedback form is incomplete, the system prompts the Branch Manager to complete the necessary fields. |
| **Exceptions** | - System downtime or network issues prevent feedback submission.  - The Branch Manager selects the wrong task, requiring correction. |
| **Priority** | Medium |
| **Frequency of Use** | Occasional (only after tasks are completed) |
| **Business Rules** | - Feedback should be submitted within 48 hours of task completion.  - Feedback must focus on both maintenance quality and technician performance. |
| **Special Requirements** | - Feedback form should include predefined options (e.g., satisfaction scale).  - Option for anonymous feedback. |
| **Assumptions** | - The Branch Manager has access to a list of completed tasks.  - The system will track feedback for evaluation. |
| **Notes and Issues** | - If forms are left incomplete, there may be issues with the user interface clarity. |

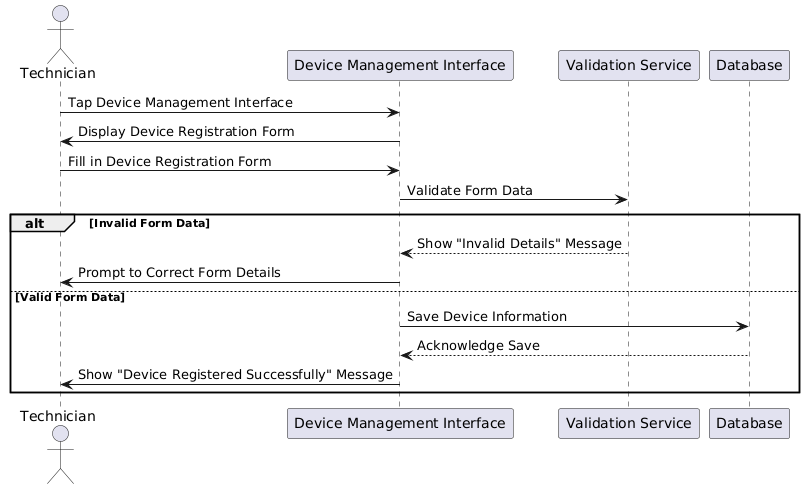
**Sequence Diagram**

A UML Sequence diagram showing the sequence of interactions among objects and used to represent or model the flow of messages, events and actions between the objects or components.

Squence diagram1 logs spare part usage

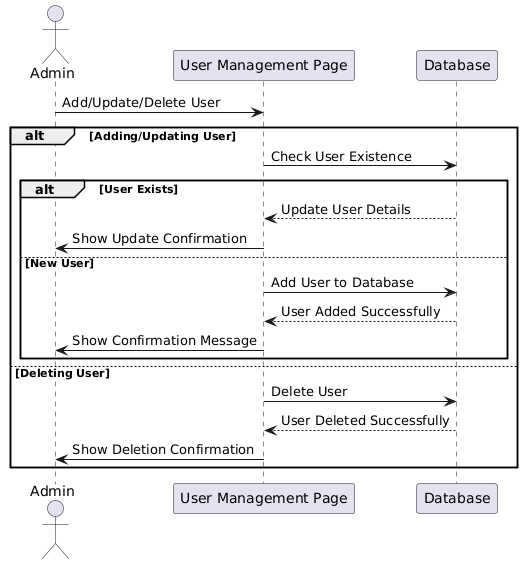


Sequence diagram 2 device registration

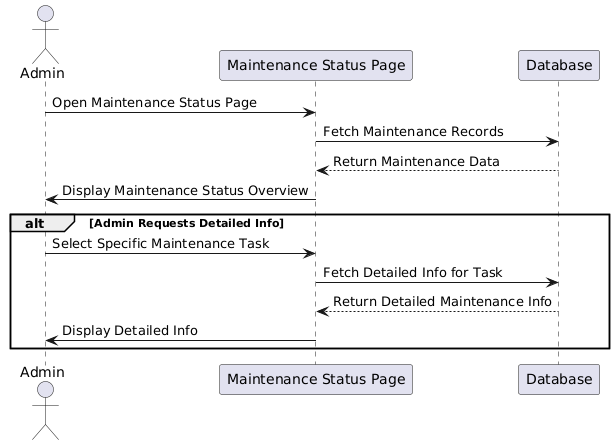


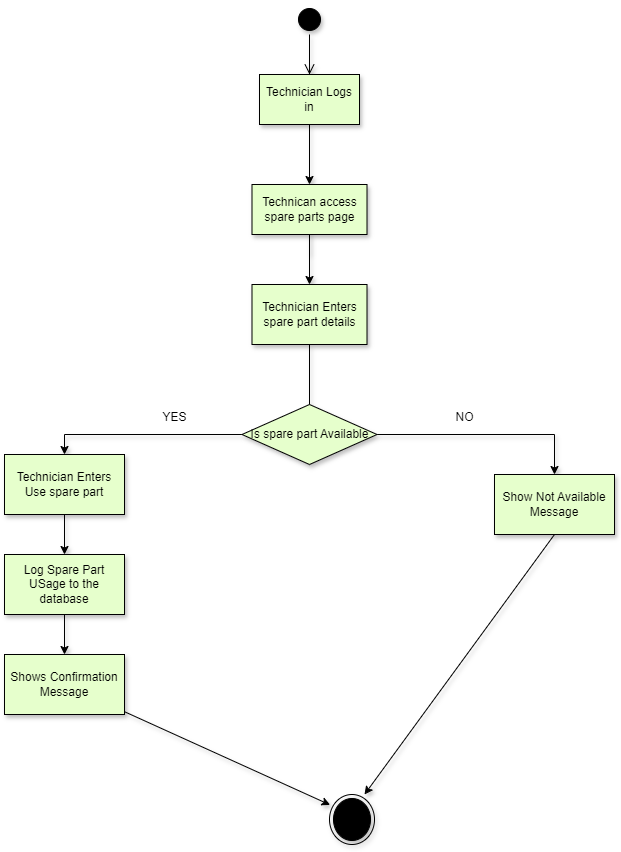
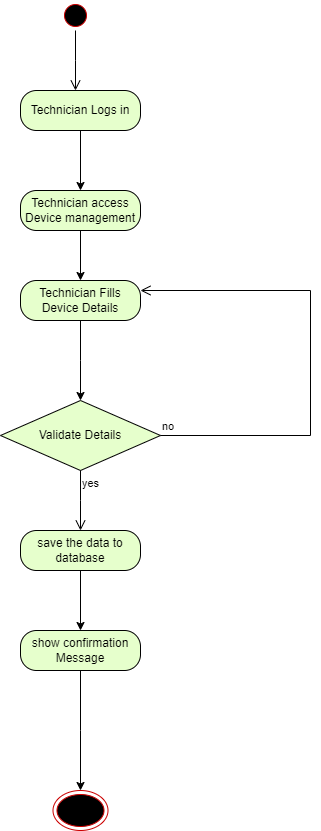
Squence diagram 3 managing user

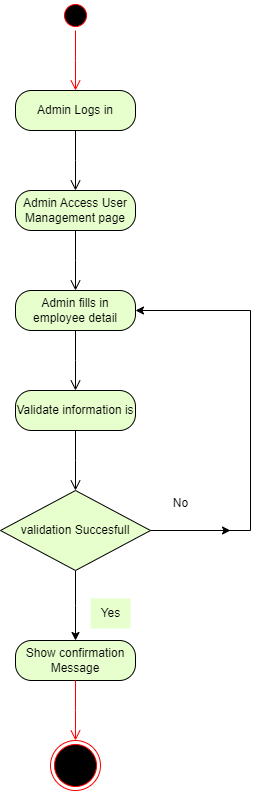
.

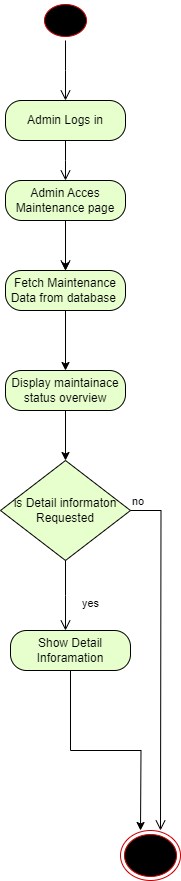


Squence diagram 4 maintainance tracking

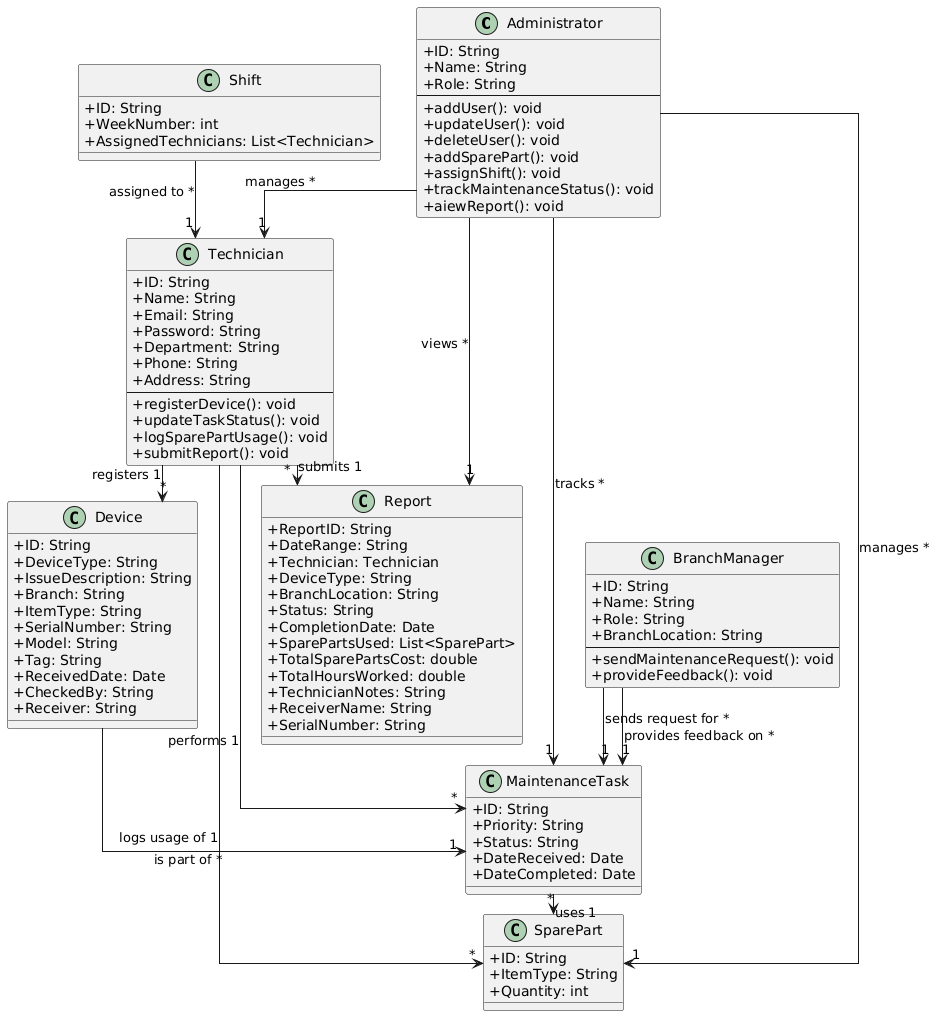








**Class Diagram**

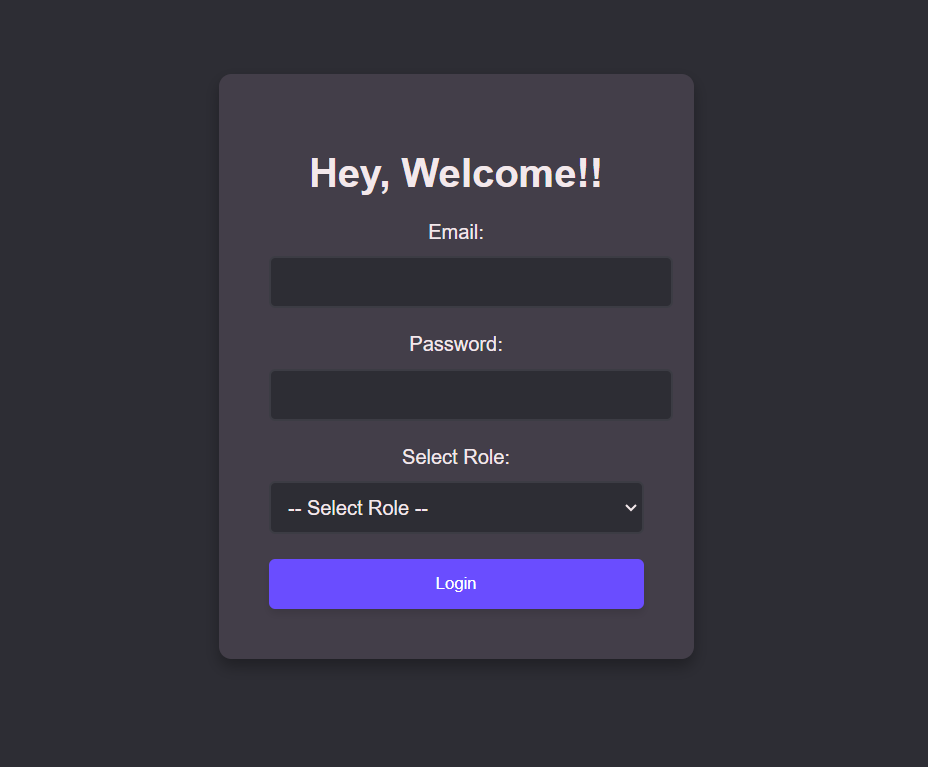


### User Interface Prototype

The **User Interface (UI) Prototype** for the Maintenance Tracking System (MTS) aims to provide a user-friendly, intuitive, and efficient platform for all stakeholders involved in the maintenance process. This prototype is designed to streamline the workflow, enhance the user experience, and support the seamless management of maintenance tasks, device registrations, shift assignments, and report generation.

The UI design will cater to the specific needs of different user roles, including the Administrator, Technician, and Branch Manager. Each role will have access to tailored features and functionalities to perform their respective tasks efficiently, ensuring smooth collaboration and real-time tracking of device repairs and maintenance.

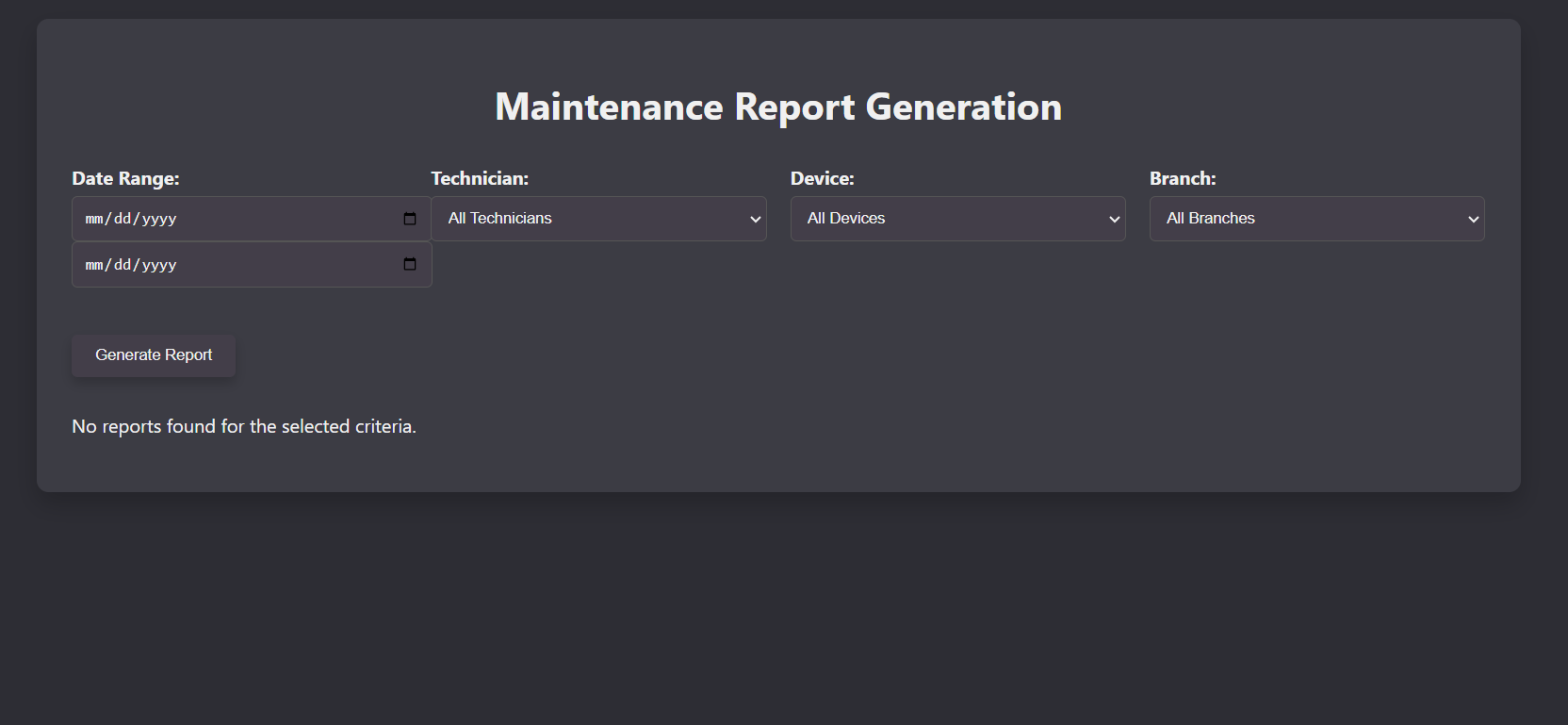
Login Page



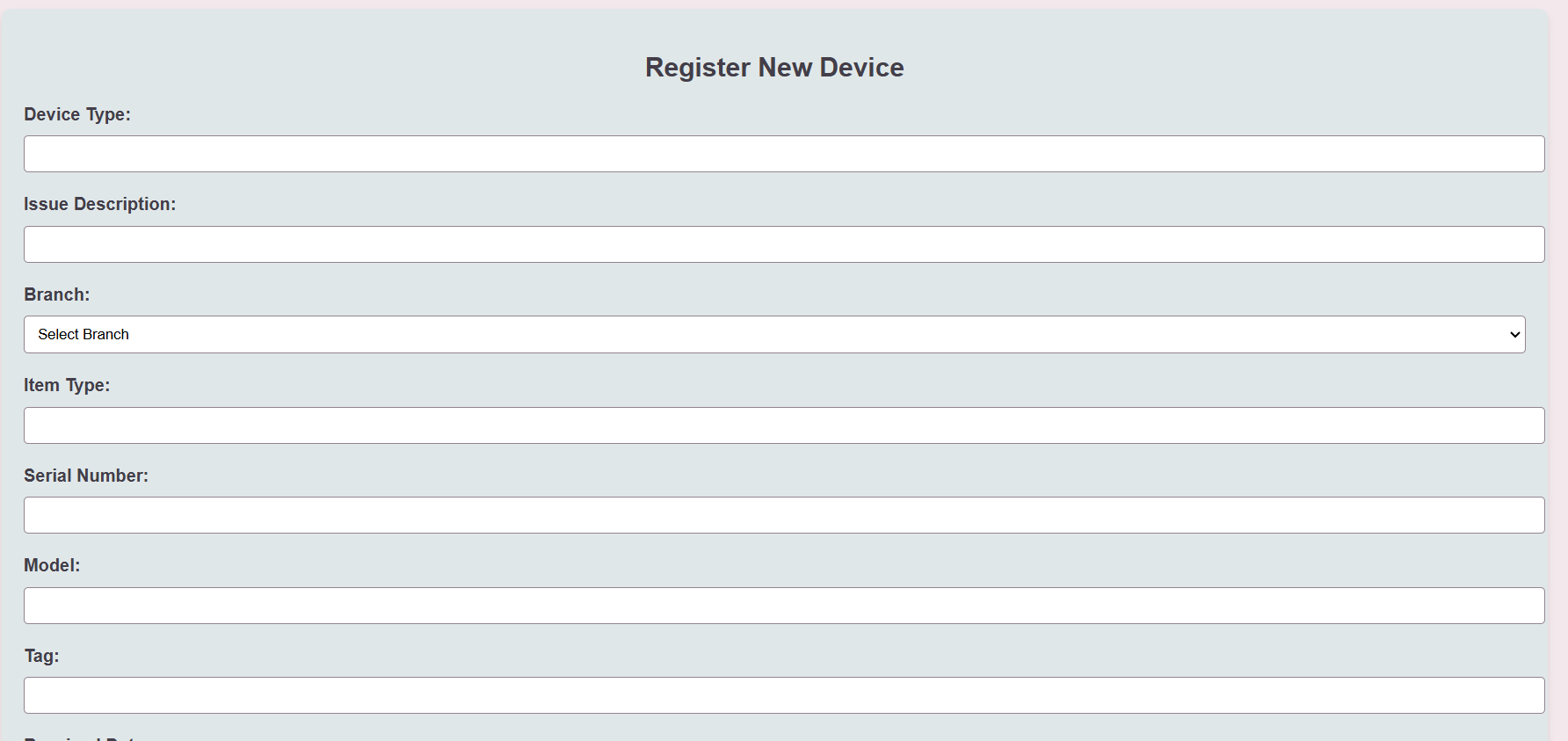
Technician Dashboard Overview



Technician report generation



Register new device Page



**Chapter Four: System Design**

**Purpose of the System Design Document (SDD)**

The System Design Document (SDD) provides a comprehensive overview of the proposed system architecture. It tracks the critical information necessary for defining the architecture of the Maintenance Tracking System (MTS). The SDD serves as a guide for the development process, ensuring the system aligns with technical, operational, and stakeholder requirements.

**Overview**

The architectural design of the MTS defines the overall system structure and the interactions between its components. The proposed system adopts a three-tier architecture to ensure scalability, modularity, and maintainability:

**Presentation Layer**: The user interface (UI) for Branch Managers, Technicians, and Administrators.

**Business Logic Layer**: Handles core system functions such as task assignment, status updates, and inventory management.

**Data Layer**: Responsible for storing and retrieving data related to maintenance requests, spare parts, and reports.

**Architectural Goals and Constraints**

* **Performance**: The system must handle multiple concurrent requests efficiently.
* **Security**: Ensure secure authentication, role-based access control, and data encryption.
* **Maintainability**: Modular design for ease of updates and troubleshooting.
* **Scalability**: Ability to expand to accommodate additional branches or new functionalities.
* **Portability**: Designed for web-based access to ensure compatibility across devices and platforms.

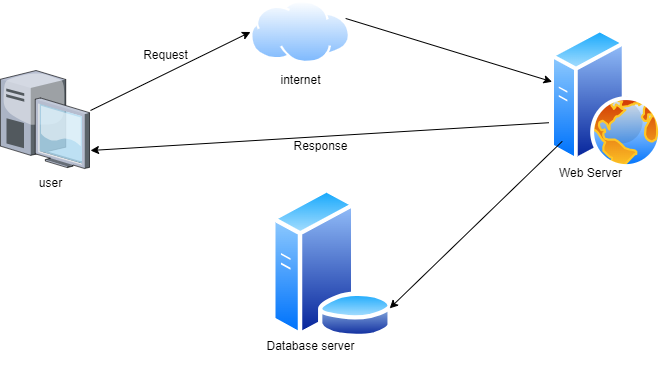
**Constraints:**

* The system relies on stable internet connectivity.
* No integration with third-party systems for spare parts procurement.
* Must adhere to CBE’s IT infrastructure policies.

### ****System Architecture Design****

### **The proposed system will have the following three tier architecture:**

1. **Presentation Layer**: This is the user interface (UI) of the system, accessible through web browsers. It allows actors like Administrators, Technicians, and Branch Managers to interact with the system.
2. **Application Layer**: This is where the business logic resides. It processes user requests, manages maintenance workflows, and communicates between the presentation and data layers.
3. **Data Layer**: This layer manages the database, storing and retrieving information about maintenance requests, spare parts, device history, user roles, and more.



**Subsystem Decomposition and Description**

**1. User Management Subsystem**

Functionality: Role-based access control and user authentication.

Actors: Administrators, Technicians, and Branch Managers.

**2. Maintenance Management Subsystem**

Functionality: Handles maintenance requests, task tracking, and status updates.

Actors: Branch Managers, Technicians, and Administrators.

**3. Inventory Management Subsystem**

Functionality: Tracks spare parts inventory, logs usage, and sends low-stock alerts.

Actors: Administrators and Technicians.

**4. Reporting Subsystem**

Functionality: Generates reports and view reports on technician performance, spare parts usage, and maintenance history.

Actors: Administrators and Technician

**5. Notification Subsystem**

Functionality: Sends real-time alerts for Maintenance Request, and inventory notifications.

Actors: branch Manager and Admin.

**Database Design**

The process of creating a structured framework for storing, managing, and accessing data efficiently in a database system. It involves defining the logical structure of the database, determining the relationships between different data entities, and ensuring the data is organized for optimal performance, consistency, and scalability.

**Steps in Database Design Process**

· **Identify Entities and Attributes**

* · Entities: Real-world objects (e.g., Technician, Device).
* Attributes: Properties of entities (e.g., TechnicianName, DeviceType).

· **Define Relationships**

* · Determine how entities are related (e.g., Technicians perform maintenance on Devices).

· **Create ER Diagram**

* · Visual representation of entities, attributes, and their relationships.

· **Normalize Data**

* · Apply normalization rules (1NF, 2NF, 3NF) to eliminate data redundancy and ensure consistency.

### ****Entities and Attributes****

**Users**

* 1. UserID (PK)
  2. Username
  3. Password
  4. Role (Administrator, Technician, Branch Manager)

**Branches**

* 1. BranchID (PK)
  2. BranchName
  3. Location

**Devices**

* 1. DeviceID (PK)
  2. DeviceType
  3. BranchID (FK)
  4. Status (Working, Pending, Maintained)

**Maintenance Requests**

* 1. RequestID (PK)
  2. BranchID (FK)
  3. DeviceID (FK)
  4. IssueDescription
  5. RequestDate

**Technicians**

* 1. TechnicianID (PK)
  2. UserID (FK)
  3. Expertise

**TaskAssignments**

* 1. AssignmentID (PK)
  2. RequestID (FK)
  3. TechnicianID (FK)
  4. Status (Pending, Working, Completed)
  5. AssignmentDate

**SpareParts**

* 1. PartID (PK)
  2. PartName
  3. Quantity
  4. Threshold

**SparePartUsage**

* 1. UsageID (PK)
  2. AssignmentID (FK)
  3. PartID (FK)
  4. QuantityUsed
  5. DateUsed

**Feedback**

* 1. FeedbackID (PK)
  2. RequestID (FK)
  3. Comments
  4. Rating
  5. DateGiven

**Notifications**

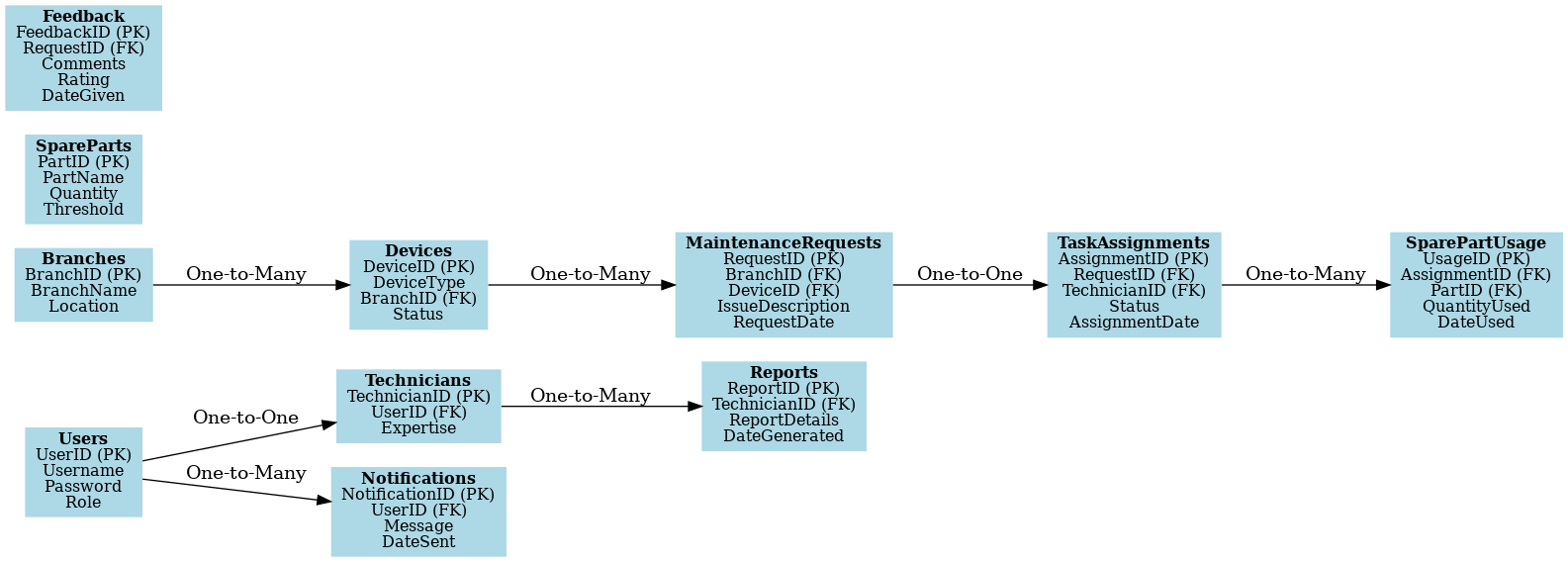
* 1. NotificationID (PK)
  2. UserID (FK)
  3. Message
  4. DateSent

**Reports**

* 1. ReportID (PK)
  2. TechnicianID (FK)
  3. ReportDetails
  4. DateGenerated

### ****Relationships****

* **Users** and **Technicians**: One-to-one
* **Branches** and **Devices**: One-to-many
* **Devices** and **MaintenanceRequests**: One-to-many
* **MaintenanceRequests** and **TaskAssignments**: One-to-one
* **TaskAssignments** and **SparePartUsage**: One-to-many
* **Users** and **Notifications**: One-to-many
* **Technicians** and **Reports**: One-to-many



**Normalization**

A key aspect of database design aimed at organizing data efficiently. It involves dividing a database into tables to minimize redundancy and dependency issues.

* **1NF (First Normal Form)**: Eliminates repeating groups; each column contains atomic values.
* **2NF (Second Normal Form)**: Removes partial dependencies; all non-key attributes depend on the whole primary key.
* **3NF (Third Normal Form)**: Eliminates transitive dependencies; non-key attributes depend only on the primary key.

A table with repeating groups and redundancies(Unnormalized Form)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RequestID | BranchName | DeviceType | TechnicianName | PartName | QuantityUsed | TaskStatus |
| 001 | Branch A | Printer | John Doe | Toner | 2 | Completed |
| 001 | Branch A | Printer | John Doe | Drum | 1 | Completed |
| 002 | Branch B | Desktop | Jane Smith | RAM | 1 | Pending |

### ****First Normal Form (1NF)****

Eliminate repeating groups by ensuring that each field contains only atomic values:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RequestID | BranchName | DeviceType | TechnicianName | PartName | QuantityUsed | TaskStatus |
| 001 | Branch A | Printer | John Doe | Toner | 2 | Completed |
| 001 | Branch A | Printer | John Doe | Drum | 1 | Completed |
| 002 | Branch B | Desktop | Jane Smith | RAM | 1 | Pending |

**Second Normal Form (2NF)**

Remove partial dependencies by ensuring all non-key attributes depend on the entire primary key:

Table 1: MaintenanceRequests

|  |  |
| --- | --- |
| **RequestID (PK**) | **BranchName** |
| 001 | Branch A |
| 002 | Branch B |

Table 2: Devices

|  |  |
| --- | --- |
| RequestID (FK) | DeviceType |
| 001 | Printer |
| 002 | Desktop |

Table 3: Technicians

|  |  |
| --- | --- |
| RequestID (FK) | TechnicianName |
| 001 | John Doe |
| 002 | Jane Smith |

Table 4: SparePartsUsage

|  |  |  |
| --- | --- | --- |
| RequestID (FK) | PartName | QuantityUsed |
| 001 | Toner | 2 |
| 001 | Drum | 1 |
| 002 | Ram | 1 |

**Third Normal Form (3NF)**

Eliminate transitive dependencies:

Table 1: MaintenanceRequests

|  |  |
| --- | --- |
| RequestID (PK) | BranchId (FK) |
| 001 | 01 |
| 002 | 02 |

Table 2: Branches

|  |  |
| --- | --- |
| RequestID (PK) | BranchName |
| 01 | Branch A |
| 02 | Branch B |

Table 3: Devices

|  |  |  |
| --- | --- | --- |
| DeviceID (PK) | RequestID (FK) | DeviceType |
| D001 | 001 | Printer |
| D002 | 002 | Desktop |

Table 4: Technicians

|  |  |
| --- | --- |
| TechnicianID (PK) | TechnicianName |
| T001 | John Doe |
| T002 | Jane Smith |

Table 5: SparePartsUsage

|  |  |  |  |
| --- | --- | --- | --- |
| UsageID (PK) | RequestID (FK) | PartType | QuantityUsed |
| U001 | 001 | P001 | 2 |
| U002 | 001 | P002 | 1 |
| U003 | 002 | P003 | 1 |

Table 6: SpareParts

|  |  |
| --- | --- |
| PartID (PK) | PartName |
| P001 | Toner |
| P002 | drum |
| P003 | RAM |