**Smart University Device & Materials Maintenance System**

---

**Executive Summary**

The Smart University Device & Materials Maintenance System is designed to manage devices and materials efficiently across all universities and colleges in Rwanda. It ensures tracking, maintenance, reporting, and accountability at a national scale.

**Key Features:**

- Role-based dashboards: Admin, Technician, Staff, Cleaner/Student
- Multi-factor authentication with OTP
- Preloaded universities, colleges, devices, and materials
- “Other” option for missing entries (admin approval required)
- Task assignment, tracking, and alerts
- Analytics, reporting, AI-ready predictive maintenance

---

**Problem Statement**

Challenges faced by universities:

- Duplicate or missing devices (laptops, routers, projectors, lab equipment)
- Delayed maintenance for doors, glass, furniture, lab instruments
- Low accountability for maintenance tasks
- Poor reporting and analytics

**Impact:** Wasted resources, higher costs, delays in teaching/research, reduced lifespan of devices and materials.

---

**Solution Overview**

Web-based platform using Node.js (backend), Bootstrap + JS (frontend), and MySQL (XAMPP).

**Core Functions:**

1. Secure login with multi-factor authentication
2. University/College selection
3. Dropdown-driven device/material registration
4. Admin assignment of maintenance tasks
5. Alerts and notifications
6. Analytics and reporting
7. AI and IoT readiness

---

**Users and Roles**

| Role | Description | Tasks |
|---|---|---|
| Admin | Oversees universities, approves accounts, assigns tasks, tracks maintenance, verifies completion | Approve new users & “Other” requests, assign tasks, monitor dashboards, generate reports |
| Technician | Performs maintenance tasks | Start/update task, complete, add notes/photos, respond to alerts |
| Staff / Lab Supervisor | Reports/records devices and materials | Add devices/materials, report issues, submit “Other” requests, track status |
| Cleaner / Student | Reports damaged materials | Select room/material type, submit notes/photos, request “Other” |

---

**Login and Authentication**

- Role selection: Admin, Technician, Staff, Cleaner
- Username/email + password
- OTP verification via SMS/email
- University/College selection (searchable dropdown)
- Access role-based dashboard

**Security:** bcrypt password hashing, JWT sessions, audit logs

---

**Universities & Colleges**

**Public Universities (UR):** CAVM, CASS, CBE, CE, CMHS, CST

**Public Integrated Polytechnics (RP):** GIP, ILPD, RP Ngoma, RP Kigali, RP Huye, RP Karongi, RP Tumba, RP Rusizi, RP Kitabi, Musanze Polytechnic, RTC, CIP

**Public Nursing Schools:** Byumba, Kibungo, Nyagatare, Rwamagana

**Private Universities:** RICA, EACC, KIC, UoK, RHIH, UNILAK, AUCA, ALU, AIWK, East African University, CMUR, ICK, CUR, HAIP, UNIK, MIPC, INES, ISPG, KP, PECDTC, ULK, KIM, MKU Kigali, PIASS, UTB, UTAB, University of Global Health Equity, ICK

Users select university/college via dropdown. “Other” requires admin approval.

---

**Device Registry**

**Device Categories:** Laptop, Desktop, Router, Switch, Projector, Printer, Server, UPS, Network Cable, Whiteboard, Microscope, Centrifuge, Lab Instrument, Table, Chair, Furniture, Microphone, Camera

**Device Attributes:** Device ID (unique), Type, Room/Location, Condition, Assigned Technician, Last Maintenance, Warranty

**Workflow:** Staff adds device → Admin verifies → Assign task → Technician updates → Admin confirms

---

**Material Registry**

**Material Categories:** Door, Glass, Table, Chair, Furniture, Whiteboard, Lab Instruments, Window, Fan, Shelf, Cabinet, Light Fixture

**Material Attributes:** Type, Room/Location, Condition, Notes/Photos

**Workflow:** Report material damage → Admin assigns technician → Technician completes → Admin verifies

---

**Maintenance Workflow**

[Report Issue] → [Admin Assigns Task] → [Technician Updates Status] → [Admin Verifies Completion] → [History Logged]

Tracks task status: Pending → In Progress → Completed

---

**Alerts and Notifications**

- Task not started → alert technician & admin
- Delayed tasks → reminders
- Completed tasks → admin notified

**User Responsibilities:** Admin monitors dashboards; Technician updates status

---

**Progress Dashboard**

- Admin: Tracks tasks by status, technician, room, or device/material
- Technician: Tracks assigned tasks

- Staff: Tracks issues they reported

---

**Reporting and Analytics**

- Maintenance history per device/material
- Technician performance reports
- Recurring issue trends, room/device heatmaps
- KPI dashboards for admin

---

**Database Design**

**Entities:** Users, Universities, Colleges, Devices, Materials, Maintenance Requests, Notifications, Technicians

**Relationships:** Users ↔ Maintenance Requests ↔ Devices/Materials ↔ Technicians ↔ Universities/Colleges

**Constraints:** Unique device ID per university; unique material+room per maintenance record

---

**Technology Stack**

| Layer | Technology |
|---|---|
| Backend | Node.js + Express.js |
| Frontend | HTML + Bootstrap + JS |
| Database | MySQL (XAMPP) |
| Notifications | Email (SMTP), SMS (Twilio), node-cron |
| Security | JWT, bcrypt, MFA |

| Layer | Technology |
| --- | --- |
| Optional AI | Python microservices |
| Optional IoT | Sensors, QR/RFID |

---

**Security**

- Role-based access
- MFA (password + OTP)
- Admin approval for “Other” entries
- Audit logs

---

**AI & Innovation**

- Predictive maintenance using historical logs
- IoT-enabled real-time monitoring
- National benchmarking dashboards
- Alerts optimized using AI

---

**System Lifecycle**

- Planning → Design → Development → Testing → Deployment → Maintenance → Innovation
- Agile methodology
- Continuous monitoring

---

**User Journey**

1. Login → OTP → University/College selection
2. Dashboard loaded by role
3. Staff adds devices/materials → Admin assigns maintenance
4. Technician updates → completes maintenance
5. Admin verifies → analytics updated
6. Alerts sent to users

---

**Implementation Roadmap**

| Stage | Duration | Activities |
|---|---|---|
| Planning | 2 weeks | Define users, universities, devices, materials |
| Design | 3 weeks | UI/UX, database, APIs, workflows |
| Development | 6 weeks | Backend, frontend, authentication, notifications |
| Testing | 2 weeks | Functional, integration, user acceptance |
| Deployment | 1 week | Pilot → national rollout |
| Maintenance | Continuous | Updates, AI & IoT planning |

---

**Conclusion and Next Steps**

- Ensures accuracy, accountability, and efficiency
- Pilot deployment in select universities
- Nationwide rollout
- AI & IoT integration
- Mobile app for technicians and staff

---

This document includes **all universities, colleges, devices, materials, user workflows, dashboards, alerts, authentication, AI/IoT readiness, database design, and lifecycle planning** for national-level implementation.