

SabreClock — Feature List & Implementation Status

This document defines the complete feature set of **SabreClock** (Time & Attendance, Access Control & Payroll Integration). It is compiled from the public marketing website, legacy requirements, and a thorough audit of the active Node.js/SQLite codebase.

1. Dashboard & Overview

- **[x] Real-time Dashboard:** Live attendance statistics showing headcount present, absent, late, or on leave. (Implemented in `dashboard/js/pages/attendance.js`)
 - **[x] Headcount Summaries:** At-a-glance dashboard metrics segmented by company and department.
 - **[x] Role-Based Access (RBAC):** UI dynamically adapts view based on user roles: Super Admin (Owner), Partner (White-Label), Reseller, and Client. (Implemented via Keycloak/Sabre SSO in `backend/services/sabre-identity.js`)
 - **[x] Multi-Company Switcher:** Standardized client account switcher in sidebar footer for clients managing multiple branches or locations. (Using shared `sabre_active_org_id` context)
 - **[x] Dark Mode Theme:** Responsive, premium dark-themed web portal built with Outfit typography and HSL color design.
-

2. Employee Management

- **[x] Employee CRUD:** Full profile creation, editing, visual search, and deactivation toggles. (Implemented in `backend/routes/employees.js`)
- **[x] Webcam Profile Capture:** Capturing employee photos directly via

webcam or uploading file attachments.

- **[x] Sage Master File Import:** Uploading legacy CSV/text master files with custom field mapping template support. (Implemented in `backend/routes/legacy-import.js`)
 - **[x] Cost-Centre Assignment:** Organizing employees by department, position, and payroll cost centers.
 - **[x] Device ID & PIN Provisioning:** Setting Enroll ID, Card Number, and Device Privilege Level (Normal User vs Admin) per employee.
 - **[x] Bulk Device Assignment Wizard:** Drag-and-drop wizard to map multiple employees to specific biometric hardware. (Implemented in `dashboard/js/pages/assignment-wizards.js`)
-

3. Attendance & Clocking

- **[x] Real-Time Log:** Live feed of incoming transactions (IN, OUT, Access Granted). (Implemented in `backend/routes/attendance.js`)
 - **[x] Chronological Pairing Engine:** Smart daily pairing that pairs chronological IN/OUT events to compute worked minutes.
 - **[x] Manual Adjustments:** Creating manual clock corrections (with mandatory editor notes and change logs).
 - **[x] Custom Reference System:** Auto-generating unique reference codes (e.g. CLK-260315-00001) for every event with quick search (Ctrl+R keyboard shortcut). (Implemented in `backend/services/ReferenceService.js`)
 - **[x] Duplicate Clocking Shield:** Suppresses double-taps at physical readers within a customizable window (default: 3 minutes). (Implemented in `backend/services/DuplicateClockingService.js`)
-

4. Biometric Device Management

- **[x] ZKTeco ADMS Cloud Protocol:** Native cloud communication server supporting ZKTeco push protocol. Hardware connects directly to the server via WAN/Internet. (Implemented in `backend/routes/adms.js`)

- **[x] HikVision ISAPI Integration:** Support for HikVision face recognition terminals via XML/HTTP requests. (Implemented in `backend/services/hikvision-isapi.js`)
 - **[x] One-Click Device Ping:** Real-time connectivity check retrieving firmware, platform, and serial details. (Implemented in `backend/routes/devices.js`)
 - **[x] Time Synchronization:** Automatic server-time updates pushed to ZKTeco and HikVision terminals.
 - **[x] Full Biometric Sync:** Pushing employee data, cards, facial templates, and fingerprint templates directly to ADMS hardware.
-

5. A.C.E — Algorithm Conversion Engine TM

- **[x] Biometric Enrollment Pipeline:** Unified enrollment UI supporting multi-placement fingerprint capture and multi-pose face capture. (Implemented in `dashboard/js/pages/ace.js`)
 - **[x] Image Quality Scoring:** Contrast, variance, and entropy scoring to filter blurry or poorly-lit face images.
 - **[x] Template Interoperability:** Storing and converting ISO fingerprint templates (supporting ZKTeco v9 & v10 formats).
 - **[x] Command-Line Face Push:** Direct ZK ADMS command generation using protocol-level `CMD_UPDATE_FACE` to bypass legacy ADMS limitations.
 - **[x] Web Biometric Bridge:** Local web socket bridge to communicate with ZKTeco USB desktop fingerprint scanners during enrollment. (Implemented in `dashboard/js/pages/ace.js`)
-

6. SabreCellClock — Mobile Clocking PWA

- **[x] Progressive Web App (PWA):** Fully installable mobile portal. (Located in `SabreClock/cellclock`)
- **[x] Offline-First Architecture:** Local storage cache allows employees to clock in/out without internet; events automatically sync when connection is restored.

- **[x] PWA Clocking Modes:**
 - **[x] Search:** PIN-based clocking after employee lookup.
 - **[x] Face:** Camera-based facial recognition matched locally or verified server-side.
 - **[x] QR Code:** Scanning mobile badge QR codes.
 - **[x] NFC:** Reading RFID/NFC chips (on compatible Android devices).
 - **[x] Roll Call:** Supervisor group clocking mode.
 - **[x] Fingerprint:** Biometric hardware tap (where native OS API permits).
-

7. BioPC — Desktop Kiosk Station

- **[x] Electron Desktop App:** System tray application designed to run on Windows PCs at receptionist desk or security gates. (Located in `SabreClock/biopc-tray`)
 - **[x] Local SQLite Database:** Local offline database caching rosters, credentials, and logs. (Implemented in `backend/routes/biopc.js`)
 - **[x] Real-time Sync:** Automatic background syncer pushing offline clock logs to the cloud once network is restored.
-

8. Shifts & Rosters

- **[x] Shift Codes & Life Cycles:** Configuring shifts with unique identifiers (e.g. STD, NIGHT), colored tags, and midnight crossing flags. (Implemented in `backend/routes/shifts.js`)
- **[x] Shift Days Rules:** Per-day overrides (e.g. normal hours Mon-Thu, shorter target on Friday). (Implemented in `backend/database.js` table `shift_days`)
- **[x] Roster Planner:** Interactive weekly/monthly grid to schedule shifts, copy weeks, assign day-offs, and swap employees.
- **[x] Shift Sensing Engine:** Auto-detects the shift worked by matching raw clock times against shift profiles using proximity scores. (Implemented in `backend/services/ShiftSenseEngine.js`)

9. Leave Management

- **[x] Leave Type Config:** Custom leave codes (AL, SL, ML, STUDY) and visual color associations. (Implemented in `backend/routes/leave.js`)
- **[x] Approval Workflow:** Employee request dashboard with multi-level manager approval (Pending -> Approved/Rejected).
- **[x] Absence Mapping:** Highlights scheduled shifts with zero clockings as "Absent" and links them to approved leave codes.

10. Access Control

- **[x] Door & Gate Zones:** Mapping biometric readers to physical access zones. (Implemented in `backend/routes/access-control.js`)
- **[x] Anti-Passback Control:** Enforces strict IN/OUT alternation to block tailgating or credential sharing. (Implemented in `backend/services/anti-passback.js`)
- **[x] Access Schedule Groups:** Time-zone restrictions specifying when departments or employee tiers can unlock doors.

11. Payroll Export

- **[x] Period Data Extraction:** Extracting total worked minutes categorized by NT, OT1.5, OT2.0, Sunday, and Public Holidays.
 - **[x] Legacy Exporters:** Generating export files for third-party payroll processors (VIP, Sage Pastel, SimplePay, LabourNet). (Implemented in `backend/routes/payroll-export.js`)
-

12. Reports & Report Builder

- **[x] Pre-Built Reports:**
 - [x] Daily Hours (showing First-In / Last-Out).
 - [x] Late Arrivals & Early Departures.
 - [x] Full Time & Attendance Register (Daily grid showing Type, NT, OT1, OT2, Sun, PH, and Exceptions). (Implemented in `backend/server.js` route `/api/reports/full-time-attendance`)
 - **[x] Drag-and-Drop Report Builder:** Building custom tabular reports with data sources, filters, grouping, and Chart.js charts. (Implemented in `backend/routes/custom-reports.js`)
 - **[x] Export Formats:** Direct download as Excel (XLSX), PDF, or print-ready layouts.
-

13. Hierarchical Rule Inheritance

- **[x] Sparse Overrides:** Sparse database structure allowing overrides at:
 - [x] **Company Level:** Global rules default.
 - [x] **Department Level:** Overriding rules for a branch or team.
 - [x] **Employee Level:** Bespoke rules for single workers (e.g. flexi hours, do not pay OT).
 - **[x] Rule Resolution Service:** A recursive resolver that crawls Employee -> Department -> Company to fetch the active value. (Implemented in `backend/services/RuleService.js`)
-

14. Overtime & Compensation Rules

- **[x] Daily Overtime Thresholds:** Overtime kicks in once worked hours exceed the daily limit (configured per day-of-week). (Implemented in `backend/services/AttendanceCalculator.js`)
- **[x] Night Shift Allowance (NSA):** Computes night allowance hours for shifts overlapping a night window (e.g. 18:00 to 06:00).

- **[x] Undertime Allocation:** Automatically borrows hours from accrued OT1 to cover NT shortfalls. (Implemented in `backend/services/AttendanceCalculator.js`)
 - **[] Weekly/Monthly Overtime Sweep:** Converting cumulative NT minutes worked in a week/month to Overtime when they cross weekly (45-hour) or monthly limits. (*Database schemas exist, but backend calculation logic is unbuilt*)
-

15. Break Deduction Rules

- **[x] Per-Shift Deduction:** Fixed deduction (e.g. 30/60 minutes) automatically subtracted from worked hours.
 - **[] Variable Break Deduction:** Deducting actual break times calculated from break IN/OUT clockings. (*Unbuilt in attendance engine*)
 - **[] Greater-Of Break Deduction:** Deducting whichever is larger: the fixed shift break or the actual clocked break duration. (*Unbuilt in attendance engine*)
-

16. Security & Infrastructure

- **[x] Keycloak & Sabre Identity SSO:** Keycloak authentication with standard role mappings.
 - **[x] Localhost / VPN / Live Environments:** Responsive network routing to adapt API paths between local development, Tailscale VPN, and production Linux VPS.
 - **[x] Audit Logging:** Logging all rule changes, employee updates, and system events.
-