ELIKSER PEOPLEOPS V3 · MODERNIZATION DOCUMENTATION

Environment URLs, Repo Strategy & V3 Database Design

Elikser Employee Portal V3 · Cloud Naming, Monorepo Governance & 3NF Azure SQL Blueprint v1.1

Environment Topology, Repository Strategy & V3 Database Architecture

Document Version: 1.1 (Aligned with `CustomerPortal` Standards)
Date: November 20, 2026
Executable SQL DDL: `docs/sql/V3_Database_Schema.sql`
Companion PRD: `docs/PRD_Employee_Portal_Modernization.md`

Part 1 — Environment URLs, Cloud Resources & `CustomerPortal` Alignment

1.1 Finalized Environment & Resource Matrix

Cross-referencing `E:\dev\CustomerPortal`, **one small naming catch** stands out: `CustomerPortal` uses **singular** `service.` (`service.customer.dev.elikser.com`, `service.customer.qa.elikser.com`, `service.customer.elikser.com`) rather than plural `services.`. Using singular `service.employee.<env>.elikser.com` keeps every Elikser platform 100% uniform.

Environment	Git Branch	Azure Resource Group	Portal Web URL (Azure SWA)	Backend API URL (Container App)	Azure SQL Database	Azure Blob Storage (Files)
Mockups *(Temp)*	Static	`employee-portal-dev`	`https://mockups.employee.elikser.com`	*(Static HTML +* `localStorage` *)*	N/A	N/A
Local Dev	`feature/*`	N/A	`http://localhost:4200`	`http://localhost:5186` (`/swagger`, `/health`)	LocalDB `sqldb-employee-local`	Azurite / `portal-files`
DEV	`Develop`	`employee-portal-dev`	`https://employee.dev.elikser.com` (`swa-employee-dev`)	`https://service.employee.dev.elikser.com` (`service-employee-dev`)	`sqldb-employee-dev`	`stemployeedev` / `portal-files`
QA	`QA`	`employee-portal-qa`	`https://employee.qa.elikser.com` (`swa-employee-qa`)	`https://service.employee.qa.elikser.com` (`service-employee-qa`)	`sqldb-employee-qa`	`stemployeeqa` / `portal-files`
PROD	`Prod`	`employee-portal-prod`	`https://employee.elikser.com` (`swa-employee-prod`)	`https://service.employee.elikser.com` (`service-employee-prod`)	`sqldb-employee-prod`	`stemployeeprod` / `portal-files`

Confirmed Infrastructure Decisions:

- **TLS Certificates:** Azure Managed Certificates per custom hostname (no manual wildcard SAN management needed).
- **Entra ID Tenant:** Single Corporate Tenant (`719456a1-069e-4f5a-a577-85431acfb7aa` — `*@elikser.com`), simpler than `CustomerPortal` since all Employee Portal users are internal staff.
- **Container Registries:** `epacrdev`, `epacrqa`, `epacrprod` (matching `cpacrdev` in `CustomerPortal`).
- **Database Engine: Azure SQL Database** (`sqldb-employee-dev`, `sqldb-employee-qa`, `sqldb-employee-prod` — can share the existing logical server `sql-perahun-centralus.database.windows.net` for cost efficiency).
- **Binary File Storage: Azure Blob Storage** (`stemployeedev`, `stemployeeqa`, `stemployeeprod`, private container `portal-files`, accessed via backend streaming or short-lived user-delegation SAS).

Part 2 — Repository Cleanup, Archival & Monorepo Naming Strategy

2.1 Why Start With a Fresh Git History (Do Not Carry Legacy History Forward)

`E:\dev\EmployeePortal` currently contains **4 separate legacy codebases** (`EmployeePortal-Service`, `EmployeePortal-FrontEnd`, `EmployeePortal-2-Service`, `EmployeePortal-2-Web`), legacy **TFVC** metadata (`.tfignore`, `.vspscc`), a KeePass vault file (`EmployeePortal.kdbx`), and historical plaintext DB/SMTP/Google ServiceAccount secrets. Initializing a fresh Git monorepo ensures **zero historical secret leakage** and a fast, lightweight clone—matching `CustomerPortal` (`https://dev.azure.com/elikser/CustomerPortal/_git/CustomerPortal`).

2.2 Recommended Repository Naming & Archival Plan (Azure DevOps)

Repo Role	Recommended Repo Name (in Azure DevOps)	Action & Permission State
Active V3 Monorepo	`EmployeePortal`	**Create fresh Git repo** (`Develop` default branch + `QA` + `Prod`). Active development happens strictly here.
V1 Archive	`EmployeePortal-Legacy-V1` *(archive)*	Push final snapshot of `EmployeePortal-Service` + `EmployeePortal-FrontEnd`, tag `v1-final-archive`, then toggle **"Disable repository"** (or set `Contribute` = **Deny**) in Azure DevOps Project Settings.
V2 Archive	`EmployeePortal-Legacy-V2` *(archive)*	Push final snapshot of `EmployeePortal-2-Service` + `EmployeePortal-2-Web` (excluding `.kdbx`, `bin/`, `obj/`, `node_modules/`), tag `v2-final-archive`, then **Disable / Lock Read-Only**.
Legacy TFVC *(if active on server)*	`$/EmployeePortal/_Archive/`	Move TFVC folders under `_Archive` and set TFVC `Check-in` permission to **Deny** so no new check-ins occur in TFVC.

> **CAUTION: Rotate Exposed Legacy Secrets During Archival**
> Just like `CustomerPortal` (`bugfix/4132`), rotate the legacy Gmail SMTP password, Google Calendar `serviceAccount.json` private key, legacy Entra client secret (`HC6zl...`), and SQL passwords that existed in V1/V2 config files, and store `EmployeePortal.kdbx` in a secure password manager rather than inside any Git repo.

2.3 Target Clean Monorepo Structure (`EmployeePortal`)

Mirrors `E:\dev\CustomerPortal` 1-for-1 (using a single `web/` app since all roles are internal Elikser staff rather than split B2B `client/` + internal `admin/`):

```text
EmployeePortal/
├── web/                                  # Modern Angular LTS SPA (Standalone Components, Signals, CDK,
Tailwind, MSAL)
├── api/                                  # .NET 10 Web API + EF Core Migrations (Azure SQL + Azure Blob)
├── api.tests/                            # xUnit Unit, RBAC & Security Regression Tests
├── docs/                                 # Authoritative Product, QA & Architecture Docs
│   ├── PRD_Employee_Portal_Modernization.md
│   ├── QA_How_To_And_Testing_Guide.md
│   ├── LOCAL_SETUP_GUIDE.md
│   ├── Environment_URLs_And_Database_Design_V3.md
│   ├── sql/V3_Database_Schema.sql
│   └── mockups/                          # Interactive V3 HTML Prototype (deployed to
mockups.employee.elikser.com)
├── pipelines/                            # Path-filtered Azure DevOps YAML pipelines
│   ├── azure-pipelines-web-dev.yml
│   ├── azure-pipelines-web-qa.yml
│   ├── azure-pipelines-web-prod.yml
│   ├── azure-pipelines-service-dev.yml
│   ├── azure-pipelines-service-qa.yml
│   └── azure-pipelines-service-prod.yml
├── scripts/                              # Entra App Registration & Azure Resource Group setup scripts
(.ps1)
├── .gitignore                            # Blocks .env*, appsettings.Development.json, *.kdbx, bin/, obj/,
node_modules/
└── .dockerignore
```

2.4 Branching & Work-Item Convention (Matching `CustomerPortal`)

- **Long-Lived Protected Branches:** `Develop` (default, deploys to `dev`) → PR to `QA` (deploys to `qa`) → PR to `Prod` (deploys to `prod`).
- **Short-Lived Work Branches:** `feature/<devops-id>-<kebab-summary>` and `bugfix/<devops-id>-<kebab-summary>`.
- **Local Disk Cleanup (`E:\dev\`):** Move the current 4 legacy folders into `E:\dev\EmployeePortal-Archive\` for local reference, and make `E:\dev\EmployeePortal\` the clean Git clone of `EmployeePortal`.

Part 3 — V3 Azure SQL Database Architecture (`sqldb-employee-<env>`)

3.1 Legacy `epdbdev07` Issues Eliminated

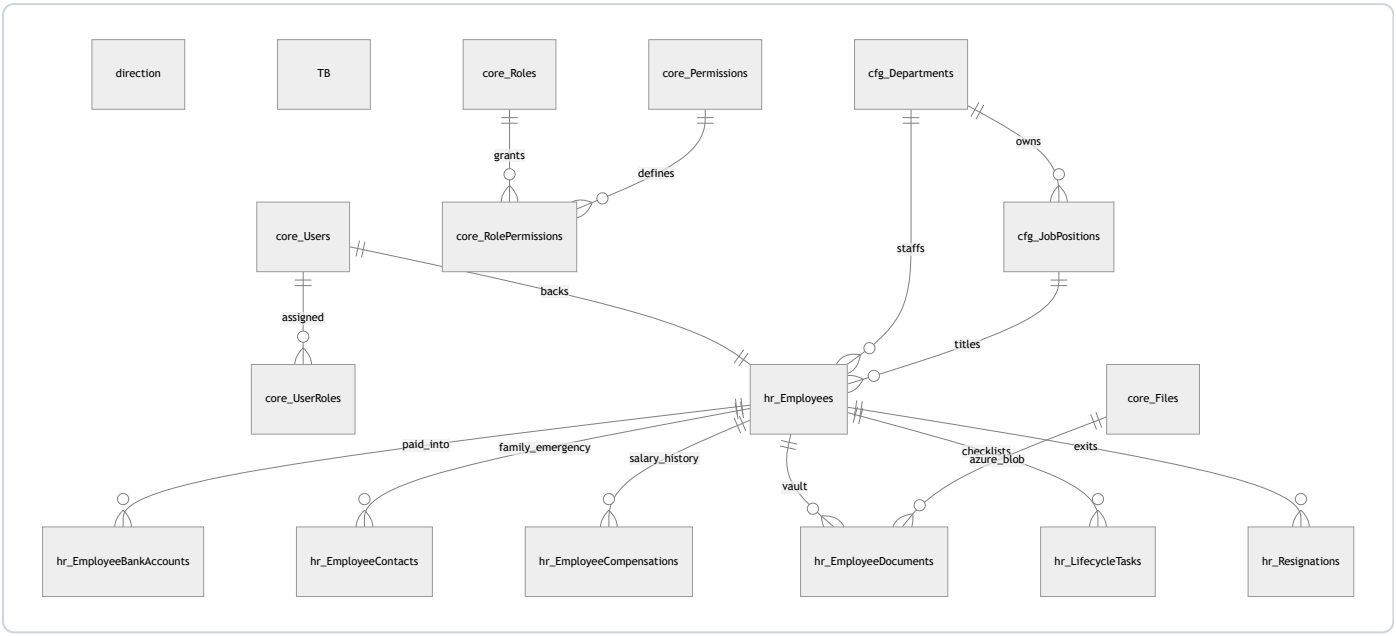
1. **Zero Blobs in SQL (`core.Files` -> Azure Blob `stemployee<env>`):** Eliminates multi-MB `VARBINARY(MAX)` and Base64 `NVARCHAR(MAX)` columns from relational rows; list queries stay fast and backups stay tiny.
2. **Table Consolidation (45+ messy tables → 9 clean 3NF schemas, 48 normalized tables incl. join tables):**
   - `EmployeeTaskList` + `EmployeeProbationTaskList` + `EmployeeOffBoardingTaskList` → single `hr.LifecycleTasks`.
   - 7 `*File` tables → single `core.Files`.
   - 4 lookup tables (`ActivityGroup`, `Tool`, `Item`, `Subject`) → single `cfg.CatalogItems`.
   - `PolicyandProcedure` + `BenefitsAndEntitlement` + `Guidelines` → `hub.KnowledgeDocuments` + `hub.KnowledgeVersions` + `hub.DocumentAcknowledgments`.
   - 45-column `Employee` + `Child` → `hr.Employees` + `hr.EmployeeContacts` + `hr.EmployeeBankAccounts` + `hr.EmployeeCompensations`.
3. **UTC `DATETIME2(3)` + `DATE` Everywhere:** Eliminates timezone/midnight bugs (`CheckIn.Date == Now`).

3.2 Legacy-to-V3 Schema Consolidation Map

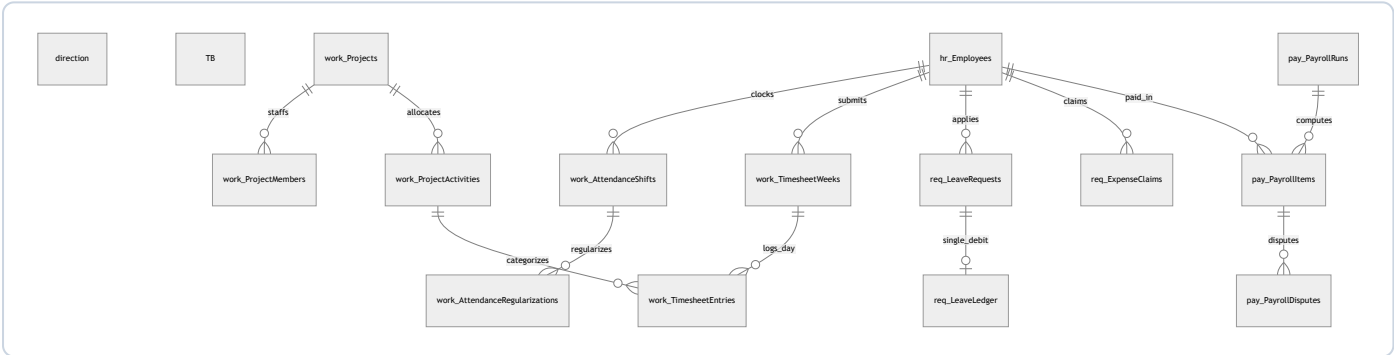
Domain Schema	V3 Tables (Clean 3NF)	Legacy Tables Replaced / Consolidated
core *(8 tables)*	Users, Roles, Permissions, RolePermissions, UserRoles, Files, Notifications, NotificationReads	AppUser, App_user, AspNetUsers, Role, RoleType, Feature, RoleToFeature, AppUserToRole, Notification, + 7 *File tables
cfg *(5 tables)*	Departments, JobPositions, CatalogItems, PublicHolidays, DocumentTemplates	JobPosition, Subject, ActivityGroup, Tool, Item, DocumentType, RequestedDocumentType *(+ Holidays & Template registry)*
hr *(7 tables)*	Employees, EmployeeBankAccounts, EmployeeContacts, EmployeeCompensations, EmployeeDocuments, LifecycleTasks, Resignations	45-col Employee, Child, EmployeeDocuments, EmployeeTaskList, EmployeeProbationTaskList, EmployeeOffBoardingTaskList, EmployeeResignation
ats *(5 tables)*	Candidates, Interviews, InterviewPanelists, InterviewEvaluations, QuestionBank	Candidate, InterviewRecord, Interview, Interviewer, Evaluation, InterviewFeedBack, Category, EvaluationInfo, InterviewQuestions
work *(8 tables)*	AttendanceShifts, AttendanceRegularizations, Projects, ProjectMembers, ProjectMemberTools, ProjectActivities, TimesheetWeeks, TimesheetEntries	EmployeAttendence, Attendance, Project, ProjectType, Team, TeamMembers, ProjectTeam, EmployeeProject, ProjectActivityGroup, EmployeeActivityGroup, Timeframe, EmployeeTimesheet, Timesheet, TimesheetDetail
req *(5 tables)*	LeaveRequests, LeaveLedger, DocumentRequests, ExpenseClaims, EquipmentRequests	LeaveApplication, LeaveRequest, LeaveHistory, EmployeeRequestedDocument, Documents, ItemRequest, ClaimExpenseRequest
pay *(3 tables)*	PayrollRuns, PayrollItems, PayrollDisputes	V1 PayRoll + PaySlipFile *(restored & normalized with batch header + dispute workflow)*
hub *(6 tables)*	KnowledgeDocuments, KnowledgeVersions, DocumentAcknowledgments, Ideas, IdeaUpvotes, BlogPosts	PolicyandProcedure, BenefitsAndEntitlement, Guidelines, Idea, IdeaManagement, Blog, BlogRequest
audit *(1 table)*	AuditEvents	*(Append-only immutable audit log for Auth, RBAC, Compensation & Attendance edits)*

3.3 Domain Entity-Relationship Diagrams

Identity, Master Config & HR (`core` + `cfg` + `hr`)



Operations, Requests & Payroll (`work` + `req` + `pay`)



3.4 Key Engine-Level Invariants in `V3_Database_Schema.sql`

- `pay.PayrollItems` **Persisted Computed Columns:** `GrossIncome` and `NetSalary` are computed by Azure SQL (`PERSISTED`) — cumulative bonus double-addition and stale net pay are impossible.
- **Non-Negative Paid Leave Check:** `CK_PayItem_NonNegLeave CHECK (PaidLeaveHours ≥ 0)`.
- **Idempotent Leave Ledger:** Filtered unique index `UX_LeaveLedger_SingleDebitPerRequest` prevents double-deducting leave balances.
- **Salary Isolation:** `hr.EmployeeCompensations` (`UX_Comp_ActiveSalary WHERE EffectiveTo IS NULL`) isolates compensation from self-service profile updates.
- **Mandatory Audit Reason:** `CK_Audit_MandatoryReason` enforces a `≥ 6 char` reason on every `ATTENDANCE_EDIT` and `COMPENSATION` event.