PROJECT_ALPHA — System Design

12. Detailed Sprint Plan (8 Weeks)

Below is the expanded sprint plan derived from the MVP roadmap. Each sprint includes specific goals, tasks, deliverables, and acceptance criteria.

Week 0: Project Setup & Planning

Objectives: Establish the technical foundation and environment.

Tasks: - Define full tech stack (backend, frontend, database, DNS engine, proxy solution). - Configure version control (GitHub/GitLab) and CI/CD pipelines. - Prepare infrastructure: choose deployment environment (local, cloud, or hybrid). - Create initial repository structure and project documentation. - Set up Docker base images (Pi-hole, FastAPI, Postgres, Redis, React frontend).

Deliverables: - Working repo with initial README, contribution guide. - Docker environment builds successfully. - CI pipeline runs automated tests and lints code.

Acceptance Criteria: - All core services can be built locally with one command. - Environments reproducible across team systems.

Sprint 1 (Week 1–2): Core DNS Filtering & Device Registry

Objectives: Implement network-level domain blocking and device management.

Tasks: - Integrate Pi-hole or dnsmasq for DNS filtering. - Implement backend API for device registration (`/api/v1/devices`). - Implement blocklist import and query endpoints. - Build backend `dns-lookup` endpoint used by the DNS filter. - Establish Redis caching layer for quick rule lookups. - Implement simple web UI (React + Tailwind) to view devices and blocklists.

Deliverables: - DNS filter blocks listed domains and logs DNS queries. - Devices can be registered, listed, and associated with blocklists. - Web dashboard displays connected devices.

Acceptance Criteria: - Device-specific DNS blocking functional. - API unit tests pass (policy checks, device registration, blocklist evaluation).

Sprint 2 (Week 3–4): Logging, Sessionization & Reporting Basics

Objectives: Capture DNS/proxy logs and convert them into user-friendly analytics.

Tasks: - Create `dns_logs` ingestion pipeline (from Pi-hole to backend). - Develop log parser + batch importer to ClickHouse or TimescaleDB. - Implement TimeSessionizer (session creation based on DNS activity). - Create dashboard widgets: Top domains, Blocked attempts, Session durations. - Add time range filters (Today, Last 7 days, Custom).

Deliverables: - Automated ingestion from DNS filter. - Computed domain sessions visible in dashboard. - Exportable CSV reports for devices.

Acceptance Criteria: - Sessionization algorithm tested on sample data (accuracy >90%). - Dashboard shows aggregated metrics correctly.

Sprint 3 (Week 5–6): Policy Management & Admin UI Enhancements

Objectives: Build tools for creating and managing filtering policies.

Tasks: - Develop Policy Editor UI (assign categories, import/export policies). - Add backend endpoints for policy CRUD operations. - Implement group assignment: device → group → policy mapping. - Add user authentication + RBAC (admin, parent, teacher). - Extend dashboard: Policy overview and quick actions (enable/disable, edit). - Generate automated email reports (daily/weekly summaries).

Deliverables: - Functional policy management via UI and API. - Secure login with JWT or OAuth2. - Automated policy enforcement verified.

Acceptance Criteria: - Policy changes propagate to DNS filter in real time. - User roles enforced correctly on API endpoints.

Sprint 4 (Week 7–8): Agent Prototype & VPN Detection

Objectives: Build optional endpoint agent and integrate VPN detection.

Tasks: - Develop lightweight Python/Go agent to capture active domains, send events. - Implement agent enrollment flow (unique device key, secure token exchange). - Extend backend with `/api/v1/agent/events` endpoint. - Add firewall rules + DNS/port filters for VPN detection. - Build dashboard alert panel for detected VPN or DoH activity. - Add audit logs (policy changes, log access, VPN detections).

Deliverables: - Agent prototype for Windows/Linux with event reporting. - Dashboard displays VPN detection alerts and suspicious activity. - Compliance logs recorded in backend.

Acceptance Criteria: - Agent sends data successfully to backend. - VPN detection rules trigger alerts on simulated VPN traffic. - Admin can view, acknowledge, and clear alerts in UI.

Sprint 5 (Optional Stretch): Security, Privacy, and Scalability Enhancements

Objectives: Harden system and prepare for pilot rollout.

Tasks: - Add data encryption at rest and in transit. - Implement GDPR/COPPA-compliant data retention policies. - Conduct load testing on DNS filter and log ingestion. - Optimize caching and query performance in ClickHouse. - Package solution as Docker Compose or Helm chart.

Deliverables: - Secure, production-ready build. - Privacy & compliance documentation. - Pilot-ready deployment scripts.

Acceptance Criteria: - System stable for 50+ concurrent devices. - Compliance checklist completed. - Successful end-to-end pilot test with real devices.

Overall Timeline Summary

Sprint	Duration	Focus
0	Setup	Environment & CI/CD
1	Weeks 1–2	DNS Filtering + Device Registry
2	Weeks 3–4	Logging + Sessionization + Reporting
3	Weeks 5–6	Policy Management + RBAC + UI Enhancements
4	Weeks 7–8	Agent + VPN Detection + Alerts
5	Stretch	Security + Compliance + Scalability

Next available step: I can generate the **detailed sprint backlog** (user stories + subtasks + acceptance tests) if you'd like to track it in Jira or Notion.