Domain Model

Model	**Fields**
User	id (str), email (str), password (str), salt (str), role (str)
Preferances	user* (User), language (str), darkMode (bool)
Endpoint	id (str), name (str), X-position (float), Y-position (float), Z-position (float), status_code (int), status_msg (str), apiKey (str), online (bool)
SignalReading	id (str), endpoint* (Endpoint), timeStamp (int), rssi (float), freqMHz (float), channel (int)
DevicePosition	Timestamp (timestamp), X-position (float), Y-position (float), Z-position (float)
ErrorReport	id(str), endpoint(endpoint), timestamp(timestamp), type(enum: “dongle_removed” , “shutdown”,), message(str)
FloorplanConfig	id(str), timestamp(timestamp), X-position (float), Y-position (float), Z-position (float), confidence(float)
Session	id(str), user*(User), issuedAt(timestamp), expiresAt(timestamp)
healthMonitor	id (str), endpoint* (Endpoint), timestamp (int), cpuPercentage (float), memPercentage (float), diskPercentage (float), donglePresent (bool), online (bool)

Component Analysis

Endpoint

Component	Component Name	Responsibility	Call Server API	Data Needed
Endpoint	config	Load the configuration data of the endpoint from a local file	FALSE	endpoint_ID, server_URL, API_key
Endpoint	scanner	Captures Wi-Fi scan data and packages it	FALSE	MAC, RSSI, timestamp
Endpoint	health_monitor	Detects the devices current health	FALSE	Dongle_presence, disk_usage, CPU_stats, memory_stats, online_status
Endpoint	local_log	Contains operational events	FALSE	Auth_events, errors
Endpoint	auth_system	Verifies endpoint successfully connects to the server	TRUE	API_key
Endpoint	scanner_report	Sends the scanner package to the server	TRUE	scanner_package
Endpoint	error_report	Generates error numbers, logs the event, and sends the error status to the server	TRUE	Error_type, timestamp, device_ID
Endpoint	device_status	Sends the monitored health data to the server	TRUE	health, device_ID

Server

Component	Component Name	Responsibility	Data Needed
Server	authentication_service	authenticate users, validate endpoint credentials, handle password reset, enfore HTTPS	user, session, endpointAuth
Server	data_ingestion	receive WiFi scans, receieve endpoint errors, validate each request, queue logic to avoid data loss	WiFiScan, ErrorReport, Endpoint
Server	status_monitor	accept health updates from endpoints, mark endpoints online/offline	healthStatus, endPoint
Server	location_processor	transform recent scans and known endpoint coordinates into estimated device positions	WifiScan, endpointLocation, FloorplanConfig

Client

Component	Component Name	Responsibility	Call Server API	Data Needed
Client	authenticateUser	Displays login status with user credentials. Validates users' credentials and prevents unauthorized users from using the	True	User == True

		system.		
Client	endpointStatus	Displays endpoint status based on data	True	Endpoint.status
Client	floorMapChange	Changes the floor map based on the file upload	True	Client.floormap
Client	accessibilityOptions	Applies alternative text to images and videos	False	accessibilityButton == True
Client	displayMap	Displays a heatmap of device locations	True	Endpoint.scanner
Client	editEndpoints	Edits the coordinates of the endpoints on the client	True	Endpoint.location

Server API

CREATE	**createUser**	**createUser(email, password**
CREATE	**createSession(login)**	**createSession(email, password)**
DELETE	**deleteSession(logout)**	**deleteSession(token)**
UPDATE	**resetPassword**	**resetPassword(email)**
CREATE	**authenticateEndpoint**	**authenticateEndpoint(endpointID,**

		apiKey)
CREATE	**ingestScans**	**ingestScans(endpointID, scans[{mac, rssi, timestamp}])**
CREATE	**reportError**	**reportError(endpointID, type, message, timestamp)**
CREATE	**sendHeartbeat**	**sendHeartbeat(endpointID, disk, cpu, mem, timestamp)**
READ	**getEndpointStatus**	**getEndpointStatus(endpointID)**
READ	**listEndpointStatus**	**listEndpointStatus()**
READ	**getHeatmap (current)**	**getHeatmap(floorplanID)**
READ	**getHeatmapWindow**	**getHeatmapWindow(floorplanID, fromTime, toTime)**
UPDATE	**updateFloorPlan**	**updateFloorPlan(file, coordinateMap)**
READ	**getFloorPlan**	**getFloorPlan()**
UPDATE	**updateEndpointPosition**	**updateEndpointPosition(endpointID, x, y, z)**
DELETE	**removeEndpoint**	**removeEndpoint(endpointID)**