

Server User Stories

User Story	Acceptance Criteria
As a system operator, I want all communications between endpoints, server, and clients to use HTTPS or another encrypted protocol so that data and credentials are secure.	Given an endpoint or client connects When data is transmitted Then it uses HTTPS/TLS (verified via inspection).
As a system operator, I want to receive Wi-Fi scan data (MAC, RSSI, timestamp) from multiple endpoints in real time so that it is available for queries and analysis later.	Given an endpoint sends scan data When the server is running Then the data is written to the database without loss.
As a system operator, I want the server to track and expose the best-known status of all endpoints (online, last scan) so that I can monitor fleet health.	Given endpoints send status messages When queried Then the server returns best-known status (online/offline, last timestamp).
As a client application developer, I want to query for the estimated locations of people in a room so that I can visualize activity at the current time.	Given a query for people's locations When the query is executed Then results include the coordinates of each person in the room.
As a user, I want the server to automatically preserve my data even if the system crashes so that I don't lose progress and can resume where I left off.	Given a database restarts When a session is interrupted, the server restores the last known valid state once the connection is re-established. Then it automatically retries to log when a request to the database fails.
As a user, I want the system to handle unexpected errors so that I receive feedback instead of crashes.	Given the system encounters an error When a request fails Then the user receives a clear error message instead of a crash.
As a user, I want to enable two-factor authentication so that my account is	Given a user has enabled 2FA When they log in with correct credentials

protected even if my password is compromised.	Then the system requires a second verification code before granting access.
As a user, I want my session to automatically expire after inactivity so that unauthorized people cannot use my account if I forget to log out.	Given a user is logged in When they remain inactive for a set period (ex. 15 minutes) Then the system automatically ends the session.
As a user, I want to log in with my email and password so that I can securely access my account.	Given valid email and password credentials When the user submits them at login Then the system authenticates and grants access to their account.
As a system operator, I want endpoints to authenticate with the server using a unique key or certificate so that only authorized endpoints can send data.	Given an endpoint sends scan/status data When the server validates its credentials (API key/certificate) Then the data is accepted; otherwise it is rejected with an authentication error.
Stretch Goal As a user, I want to log in with my Google account so that I can quickly authenticate without creating another password.	Given the user selects "Login with Google" When OAuth flow completes Then the system grants access