

Requirements Documentation

Interactive House – Subgroup 2: Units

Revision History

Date	Version	Description	Author
07/02/2026	1.0	Initial draft for Units subgroup (week 1)	Mustafa Al-Bayati
17/02/2026	1.1	Add Requirements for Mobile App (R14-R18)	Daniel Jönsson

Requirements List

Requirement Name	Priority
R1. User authentication (login/logout)	Essential
R2. Connect to House Server / database	Essential
R3. Display device list and status	Essential
R4. Load device UI components	Essential
R5. Send control commands to devices via server	Essential
R6. Receive real-time updates/events from server	Essential
R7. Handle connection loss and reconnection	Essential
R8. Basic access control on client (authorized features only)	Essential

R9. Activity/history view (user actions + device events)	Desirable
R10. Notifications/alerts (e.g., alarm, device status change)	Desirable
R11. Multi-platform support (web+Mobile)	Essential
R12. Usability & accessibility (clear UI, large buttons, contrast)	Desirable
R13. Setting (language/theme)	Optional
R14. Mobile Biometric Authentication	Desirable
R15. Mobile Native Push Notifications	Desirable
R16. Mobile Haptic Feedback	Desirable
R17. Mobile Gesture-based Navigation	Essential
R18. Mobile Design Coordinated with Web	Essential

Requirements Descriptions

1. General

R1. User authentication (login/logout) - Essential

The unit shall provide a login screen where a user can authenticate against the House Server using credentials (e.g., username/password). The unit shall support logout and clear any locally stored session token on logout.

R2. Connect to House Server - Essential

The unit shall be able to connect to a configured House Server endpoint (IP/host + port). The unit shall indicate connection state (Connected / Connecting / Disconnected).

R3. Display device list and status - Essential

The unit shall show a device list including at minimum: device name/id, device type (if available), and basic status (e.g., on/off/temperature value/online).

R4. Load device UI components provided by server - Essential

The unit shall be able to receive device-specific UI definitions/components from the server and render them in the unit UI to control/observe the device (e.g., “Light ON/OFF” UI).

R5. Send control commands to devices via server - Essential

The unit shall allow the user to perform control actions (e.g., turn light on/off) and send the corresponding command to the server using the agreed protocol. The unit shall show command success/failure feedback.

R6. Receive real-time updates/events from server - Essential

The unit shall be able to receive device updates (state changes, sensor values, online/offline) from the server without requiring manual refresh (e.g., via WebSocket, SSE, long polling, or periodic polling if needed).

R7. Handle connection loss and reconnection - Essential

If the connection drops, the unit shall notify the user and attempt reconnection automatically (or provide a clear “Reconnect” action). The unit shall not crash on disconnect.

R8. Basic access control on client (authorized features only) - Essential

The unit shall only display devices and actions that the logged-in user is authorized to access, based on information returned by the server.

R9. Activity/history view - Desirable

The unit should provide a history screen showing recent actions and events (e.g., “User X turned Light ON”, “Temperature updated to 22°C”), retrieved from server and/or cached locally.

R10. Notifications/alerts - Desirable

The unit should notify the user of important events (e.g., alarm triggered, device offline) via UI alerts and, on mobile, push/local notifications if feasible.

R11. Multi-platform support (Web + Mobile) - Essential

The Units solution should support at least one of:

Web UI (browser-based)

Mobile app UI (Android or iOS or both)

R12. Usability & accessibility - Desirable

The unit UI should be simple, with clear device names, large buttons for critical actions, readable font sizes, and high contrast options where possible.

R13. Settings - Optional

The unit may provide a setting options like theme mode light/dark/color, and two languages like en/se.

2. Mobile (React Native + Expo)

R14. Mobile Biometric Authentication - Desirable

The mobile app should allow users to enable Biometric Login (TouchID, FaceID, or Android Biometrics) after the initial manual login. This provides a faster, secure way to access the House Server without re-entering credentials.

R15. Mobile Native Push Notifications - Desirable

The app should integrate with Expo Notifications to receive and display system alerts even when the app is in the background, or the phone is locked. This is critical for high-priority events like a security breach or connection loss.

R16. Mobile Haptic Feedback - Desirable

The app should provide tactile (vibration) feedback via the Expo Haptics API when a user toggles a switch or when a command fails. This improves accessibility and provides physical confirmation for blind interactions.

R17. Mobile Gesture-based Navigation - Essential

To optimize the small-screen experience, the app shall support native mobile gestures. This can include pull to refresh on the device list to sync with the server or to navigate through the different tabs smoothly.

R18. Mobile Design Coordinated with Web - Essential

The mobile UI shall maintain visual consistency with the Web version by using the same color palette and terminology. While the layout must be optimized for vertical touchscreens, the paired functionality shall ensure that a user transitioning from the Web dashboard to the Mobile app experiences no learning curve.