Supplementary Requirements Documentation

Interactive House - Subgroup 2: Units

Revision History

Date	Version	Description	Author
2026-02-04	1.0	Initial revision (Add S1 – S8)	Daniel Jönsson

Supplementary Requirements List

Supplementary Requirement Name	Priority
S1. Usability, Accessibility & Independence	Essential
S2. Reliability - Synchronized State	Essential
S3. Performance - Interaction Latency	Essential
S4. User Interface - Dynamic Scalability	Essential
S5. Hardware Requirements - Multi-modal Input	Essential
S6. Design language	Essential
S7. Implementation Platform	Desirable
S8. Local Persistence – Offline Mode Mock-up	Optional

Supplementary Requirements Descriptions

S1. Usability, Accessibility & Independence

The interface must prioritize self-control and accessibility for users with high grades of disabilities. Controls must be learnable within a minute, enabling patients to access home functionalities like turning on lights, opening windows, doors, or a coffee machine independently.

S2. Reliability - Synchronized State

The unit must maintain a stable connection with the House Server. It must reflect the present devices for observation without locking the program, ensuring the user always sees an accurate list of controllable devices provided by the server.

S3. Performance - Interaction Latency

The interface must be highly responsive to provide a ubiquitous feel. Commands sent from the unit like toggling a light and the rendering of new UI components uploaded from the server should occur within 3 seconds to ensure real-time control.

S4. User Interface - Dynamic Scalability

The Unit must be capable of rendering unique UI software components sent from the server for any plugged-in device. To maintain simplicity for disabled users, the maximum menu depth for any device control should not exceed 3 levels.

S5. Hardware Requirements - Multi-modal Input

Mobile Units: Must support touchscreens like iPhone and Android, speech recognition, and gestures (at screen or in the air).
Web Units: Must be compatible with standard laptop displays and mouse utilities.
Other: Support for components detecting brain activities for severely disabled users.
(maybe it's too difficult?)

S6. Design Language

The unit's architecture and interaction logic must be described through specific UML diagrams, including Use Case, Class, Interaction Sequence, and State diagrams.

S7. Implementation Platform

We have an alternative of using Android Studio + Jetpack Compose specifically for mobile devices, but this only works on Android. The Units will therefore be implemented using React Native and Next.js. This choice is a non-functional constraint to ensure a uniform model of communication. By using a JavaScript-based stack, the subgroup can fulfil the requirement to dynamically upload and render UI components received from the server, while maintaining cross-platform support for both mobile touchscreens and web-based mouse interfaces.

S8. Local Persistence – Offline Mode Mock-up

The Unit may implement a demo mode or cached state. If the House Server is temporarily unreachable, the Unit should show the last known state of the house rather than a blank screen.