Design Evolution

The feedback from the usability tests and evaluation conducted during the formative feedback helped guide modifications and evolutions to the design. The following design modifications were implemented for our Alpha System:

1. Removal of Remote Calibration Options from Startup Sequence

Change: The remote calibration option that appeared during the initial startup sequence was removed entirely from the system.

Source: Peer feedback recommendations during the formative assessment.

Rationale:

- **Usability:** The calibration process added unnecessary steps to the first-time setup experience. During our formative feedback review, our reviewer outlined how we could forgo this step. Users want to begin using the mirror immediately, and requiring calibration before the system could be used introduced cognitive load and failure points.
- **Feasibility:** Standard infrared (IR) remotes operate on universal protocols that do not need device-specific calibration. By relying on those standardized signals, we eliminated a technically unnecessary step.

Impact: The startup flow is not streamlined, reducing the time-to-first use by approximately 15 seconds. This change reduces the risk of abandonment during setup.

2. Transition from Computer Prototype to Physical Mirror

Change: Transformed the computer-based prototype into a fully functional physical smart mirror using a Raspberry Pi, LCD monitor, acrylic panel, and a two-way mirror film. We also added an IR sensor to the Raspberry Pi for remote control compatibility.

Source: Designer insight based on need for authentic user testing and technical implementation requirements.

Rationale:

- **Usability:** A physical prototype allows users to interact with the system in its intended real-world context- as a bathroom or bedroom mirror. This authentic environment reveals usability issues that cannot be discovered via screen-based simulations, like viewing angles, ambient lightning interference, and the ergonomics of using a remote while standing at different distances.
- **Feasibility:** Moving to physical hardware was essential to validate technical assumptions about display visibility via the two-way mirror, IR signal reception range, and reliability.

Impact: This enabled real usability testing and showed important insights about optimal display brightness, mirror positioning, and interaction distance that would have been impossible to discover with computer simulation.

3. Implementation of Google Calendar Synchronization

Change: Added Google Calendar integration allowing users to sync their calendar by scanning a QR code or entering a pairing code using their smartphone. This deature was implemented using Google Cloud Platform APIs and OAuth 2.0 Authentication.

Source: User feedback from formative testing and designer insight about desired functionality.

Rationale:

- **Usability:** The QR code scanning option provides a frictionless authentication method that takes less than 10 seconds, while the manual code option serves s a fallback for accessibility. Allowing the user to use their smartphone for this makes it a lot easier than requiring them to type in their information using a remote.
- **Feasibility:** Google Calendar is the most widely used digital calendar platform, making it the logical integration choice. We chose this over Apple Calendar, as we would have needed to pay a hefty fee to use Apple APIs. The OAuth authentication flow ensures security without requiring users to enter passwords using the remote. Also, it positions the system for future integrations with other Google services.

Impact: Calendar synchronization is one of the most valued features brought up during testing. This makes the feature accessible.

4. Removal of WiFi Setup from Settings and Startup Menus

Change: Eliminated all WiFi configuration interfaces from both the initial startup flow and the settings menu.

Source: Suggestion from the evaluation that Raspberry Pi's have their own WiFi network already.

Rationale:

- **Usability:** Configuring WiFi via a TV remote is notoriously frustrating as it needs slow character-by-character entry using directional buttons. This process is time-consuming and error-prone, especially for complex passwords. By removing this requirement, we eliminated a significant pain point in the setup experience.
- **Feasibility:** The Raspberry Pi's built-in WiFi configuration is already taken care of by the Raspberry Pi. Hence, this made in-app WiFi setup redundant.

Impact: Reduces setup time and user frustration.