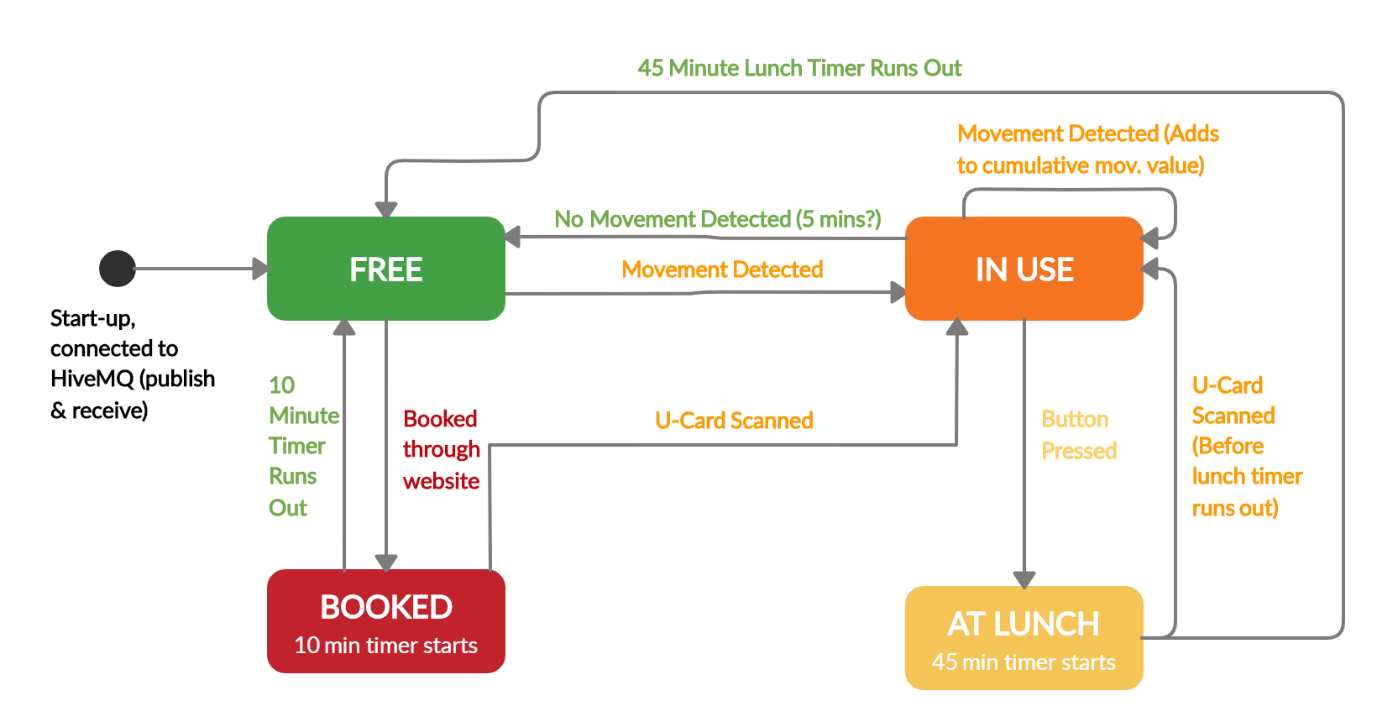
**M5 Stack Documentation**

**M5 Stack Requirements:**

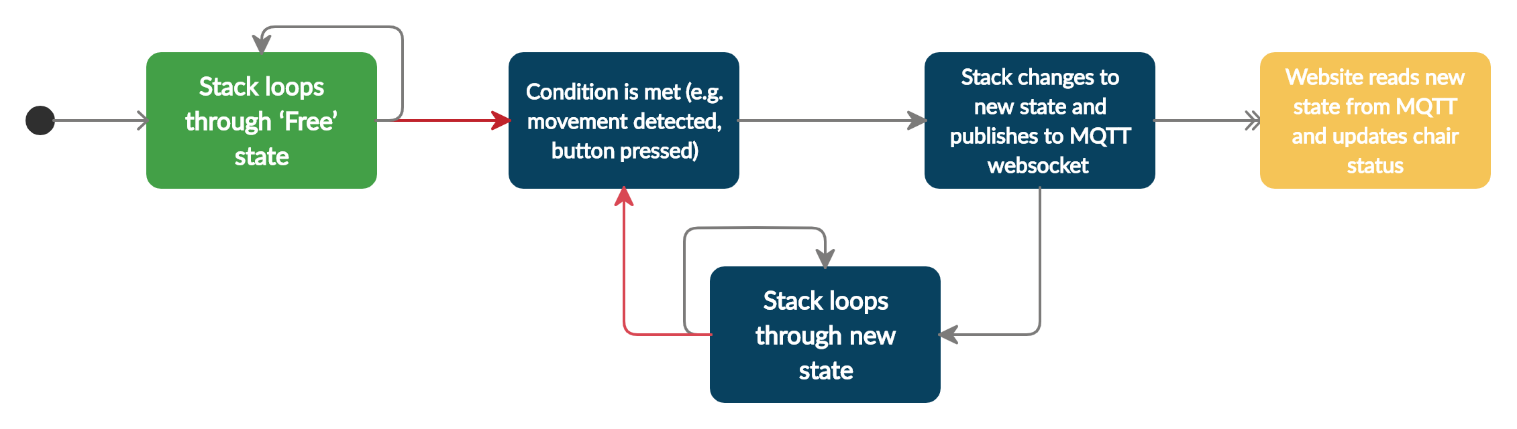
**M5 Stack/Stick Code:**

****Both devices are programmed to follow the below state machine:

**Design Process:**

Initially had the stack set to free, then a lot of ‘if’ statements testing for the conditions above all in one loop. Once conditions were met, the appropriate message was published to MQTT (‘booked’, ‘use’, etc) which triggered the callback function when the stack read its own messages. The callback function then printed the appropriate screen to the stack and changed the state.

* This caused a significant delay between conditions changing (i.e. detecting movement) and the screen changing because it had to publish then read back its own message.
* Code was very messy with no clear functions, and trying to set a timer was very difficult.
* The ‘In Use’ mode only changed based on current movement.

Revised code design compartmentalised the states in to functions which run while loops until conditions are met to change them. Code now follows below process:

This refactoring removes the delay from publishing and reading, while still publishing the chair state for the website to update and keep track of.

‘Use’ mode now tracks cumulative movement and has an initial starting value which decreases with time and increases when it detects movement (proportional to the amount of movement detected). This prevents the state from reverting back to ‘Free’ as soon as no movement is detected, and also stops the chair from staying in ‘Use’ when it is knocked by somebody walking past.

‘Lunch’ mode on the stick differs from the stack because of the RFID capabilities. Stack reverts to ‘Use’ from ‘Lunch’ when a card is scanned on the stack, but when it detects continued movement on the stick. ‘Booked’ is also disabled by pressing the button on the stick, but scanning a card on the stack.