Functional design – ESE & IPO

By Manon Sijben, Marlon Reijlink, Emiel Visser and Joris Bol

HAN_

DOCUMENT ORGANISATIONAL DETAILS:

Document Type:	
Skills Consultant:	Paul van Wegen
Project Client:	Johan Korten
Study:	Embedded Systems Engineering & Industrial Design Engineering
Institute:	Hoge school Arnhem-Nijmegen
Date:	1-10-2023
Version:	V1.0
Authors:	2108100 (Joris Bol), 2111740 (Emiel Visser), 2106573 (Manon Sijben), 2107003 (Marlon Reijlink)

Version management

Version	Date	Change	Name
V1.0	5-9-2023	Initial setup	J. Bol
V1.0	9-9-2023	Added information	E. Visser
V1.0	11-9-2023	Added information	Everybody
V1.0	12-9-2023	Added information	J. Bol & E. Visser
V1.0	26-9-2023	Added UML.	J. Bol
V1.0	1-10-2023	Spellcheck.	E. Visser
V1.0	12-12-2023	Added changes.	E. Visser

Inhoudsopgave

1 – Functional design

The primary function of our product is to introduce a variation in skin colours. The following list of functional requirements has been defined in consultation with our client. This list consists of three columns: The requirement number, prioritization based on the MoSCoW method and formulation of the requirement according to the SMART method.

With the MoSCoW method we can determine the importance of each specification. The M stands for *must have*. We must implement this specification in our final product. The S stands for *should have.* denoting features that are highly desirable and expected in the final product. The C stands for *could have*. suggesting optional features that may be considered for inclusion. The W stands for *won't have*. indicating that these features won't be incorporated into the final product due to their perceived difficulty or lack of importance.

The SMART method dictates that each specification must be formulated according to the following guide rules:

- Specific: Goals should be clear and specific.
- Measurable: Goals should have measurable criteria for tracking progress.
- Achievable: Goals should be realistic and attainable.
- Relevant: Goals should be relevant to your overall objectives.
- Time-bound: Goals should have a defined timeframe or deadline.

Using the SMART method helps ensure that goals are well-defined, achievable, and aligned with your larger objectives, increasing the likelihood of success.

Functional specifications		
#	MoSCoW	**Description**
F1	**M**	**The skin should be able to change colour realistically.**
F1.1	**M**	The skin must be able to turn yellow.
F1.2	**M**	The skin must be able to turn red.
F1.2	**M**	The skin must be able to turn dark blue.
F1.3	**M**	The skin must be able to turn pale.
F1.4	**S**	The skin shouldn't emit a lot of light.
F1.5	**S**	The skin should be able to accurately simulate petechiae. **Added on 10-10-23**
F2	**S**	**The skin should feel realistically.**
F2.1	**S**	The skin should not feel sticky.
F2.2	**S**	The flexibility of the skin should match the flexibility of real skin.
F3	**S**	**The skin should look realistically.**
F3.1	**S**	The skin shouldn't enter the uncanny valley*.
F3.2	**S**	The colour of the skin should be realistic.
F4	**C**	**The skin could react to being touched.**
F4.1	**C**	The touched spot could display capillary refill**.
F5	**S**	**The system should work on different skin colours.**
F5.1	**S**	White skin colour.
~~F5.2~~	~~**S**~~	~~Brown skin colour.~~ **Removed on 10-10-23**
~~F5.3~~	~~**S**~~	~~Dark brown colour.~~ **Removed on 10-10-23**
F6	**M**	**We must test our system on different skin colours.**
F6.1	**S**	White skin colour.
F6.2	**S**	~~Brown skin colour.~~ **Removed on 10-10-23**
F6.3	**S**	~~Dark brown colour.~~ **Removed on 10-10-23**

*Used in reference to the phenomenon whereby a computer-generated figure or humanoid robot bearing a near-identical resemblance to a human being arouses a sense of unease or revulsion in the person viewing it.

**The 'capillary refill' time is the time it takes for colour to return to a distal capillary network after pressure has been applied.

2 – Technical specifications

The table below presents the technical specifications, adhering to the same formatting principles we applied to the functional specifications table in paragraph 1.

Technical specifications		
#	Moscow	Description
T1	**M**	**The used microcontroller is the SAMD21~~J18A-M~~. Removed on 4-12-2023**
T1.1	M	The microcontroller must be placed on a PCB designed by us.
T2	**M**	**The used silicone is Eurosil 8 or 33 Transparant.**
T2.1	M	The silicone must be used to create the skin for the baby.
T2.2	M	The silicone must not tear apart.
T3	**M**	**The used pigments come from NEILL's.**
T3.1	M	X gram of pigment is used for one litre of silicone.
T3.2	M	X gram of flocking material is used for one litre of silicone.
T4	**M**	**The code must be written in CPP.**
T5	**S**	**We should make use of flex PCBs**
T5.1	C	We could use FFC connectors.
T5.2	C	We could use x amount of buttons.
T5.3	C	We could use LEDs.
T5.3.1	S	*We should supply the LEDs separately with power.*
T6	**M**	**The system must be modular.**
T6.1	M	The control PCB must be detachable from the main system.
T6.2	M	The flex PCBs must be detachable from the control PCB.
T7	**M**	**The control PCB must be connected to the main I2C bus to the rest of the system.**
T7.1	M	There must be two I2C ports. An input and output port.
T7.2	M	The standard I2C protocol must be used.
T8	**M**	**The PCBs must be made as compact as possible.**
T9	**M**	**The system must produce as little heat as possible.**
T10	**S**	**The status of the PCB must be made visible to the user.**

3 – User interface

A user interface (UI) serves as the medium through which a user interacts with a system or application. Examples of user interfaces include physical screens, on-screen applications, or even a basic indicator like a led, all of which enable users to engage with a machine or software.

In the following paragraph, we will detail the specific types of user interfaces we intend to use in our project, elaborate on the information they will present to the user, and outline our implementation strategy.

3.1 – On board interface / heartbeat

We intend to communicate vital information regarding our system's operation using six LEDs. These LEDs will serve as intuitive indicators, providing users with a clear understanding of the system's status.

In the following paragraph, we will explain the specific roles of these LEDs and describe how they can be effectively utilized to monitor the system's condition.

3.1.1 – Serial LEDs

The Serial LEDs are designed to provide a visual representation of both the RX (Receive) and TX (Transmit) signals between the microcontroller.

These LEDs will be placed next to each other. We did this to maintain a good overview of the serial communication status.

3.1.2 – Heartbeat LED

As the name suggests, this led blinks like a heart to indicate that the board is still 'alive.' The led will turn on and of every second.

3.1.3 – Sense LED

The sense led turns on when one of the buttons is pressed. This is extremely helpful while debugging.

3.1.4 – RGB LED

The RGB led indicates the state of the PCB. The distinct colours indicate the following states:

Colour:	State:
Off	Nothing is wrong, PCB is off.
White	The PCB went into hard fault.
Red	There is no connection between the PCB and the main board.
Green	INIT finished successfully.
Blue	The PCB is in test mode.

3.1.5 – 3V3 LED

The 3V3 led indicates when a 3.3 volt is present. This means that when the board is connected to power, the led turns on. This is not dependent on the chip.

4 – User interfacing

The system will interact with the user in a before specified manner, but not only the user will have impact. Our board will be largely controlled by another piece of hardware and software. This interaction is by touch in case of the user and by I2C in case of the other hardware.

A UML use case diagram is shown below, it describes what actions can be taken and how our system will react to them. In our case there are two primary actors, both depicted on the left.

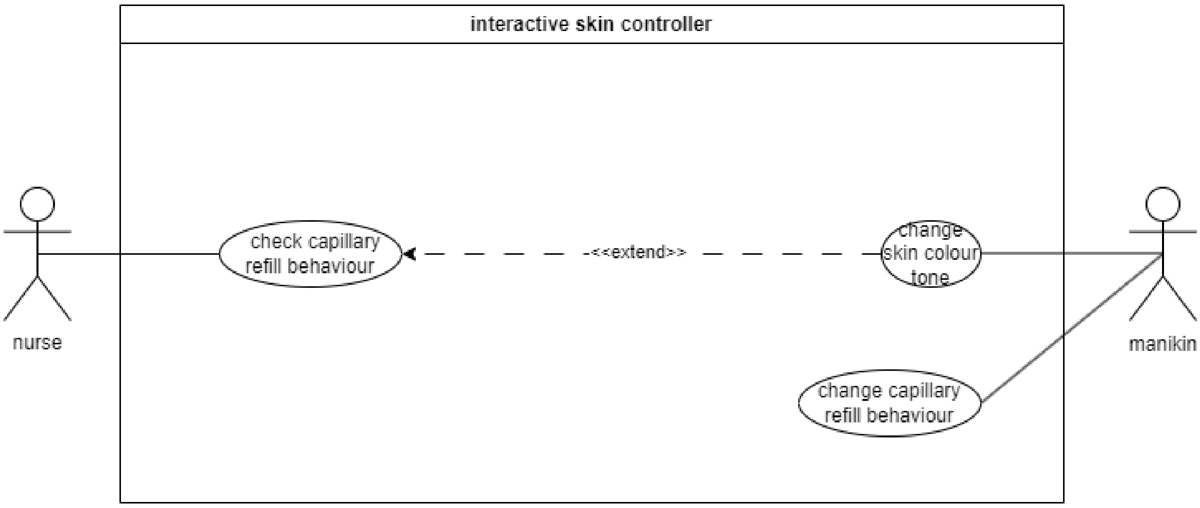

Figure 1 UML use case diagram