Graphical user interface

Description automatically generated

* Authors :
  + Victor Covalciuc (4428854)
  + Sean-David Brokke
  + Andre Sanao (4425553)

|  |
| --- |
| System Design Document  Smart Washing Machine |
| |  |  |  | | --- | --- | --- | | GROUP #3 | 28-Oct-22 | T-CB-PRO3-T3-CB01 Design Project | |

Table Of Contents

[List of Figures 2](#_Toc124514691)

[List of Tables 2](#_Toc124514692)

[Document history 3](#_Toc124514693)

[WDW [Who Did What] 4](#_Toc124514694)

[Terms, Abbreviations, Acronyms 5](#_Toc124514695)

[1. Introduction 6](#_Toc124514696)

[2. System description 6](#_Toc124514697)

[1.1 Washing programs description 6](#_Toc124514698)

[1.2 Safety 8](#_Toc124514699)

[3. System Design 9](#_Toc124514700)

[1.1 Use cases 9](#_Toc124514701)

[1.2 User requirements 12](#_Toc124514702)

[1.3 Requirements traceability matrix 14](#_Toc124514703)

[4. System structure 16](#_Toc124514704)

[5. System behaviour 17](#_Toc124514705)

[1.4 Sequence Diagrams 17](#_Toc124514706)

[1.5 State Machines 22](#_Toc124514707)

[1.5.1 Main controller 22](#_Toc124514708)

[1.5.2 Control panel 25](#_Toc124514709)

[1.5.3 Door controller 25](#_Toc124514710)

[1.5.4 Drum controller 25](#_Toc124514711)

[1.5.5 Water inlet valve controller 25](#_Toc124514712)

[1.5.6 Heating controller 27](#_Toc124514713)

[1.5.7 Drain controller 28](#_Toc124514714)

[1.5.8 Detergent dispenser controller 28](#_Toc124514715)

[6. Recommendations and conclusion 29](#_Toc124514716)

[7. References 30](#_Toc124514717)

# List of Figures

[Figure 1 - Use case diagram of Smart washing machine system 9](#_Toc124514674)

[Figure 2 - Sequence Diagram Scenario 1 from UC\_001 17](#_Toc124514675)

[Figure 3 - Sequence Diagram Scenario 2 from UC\_001 18](https://stichtingfontys-my.sharepoint.com/personal/477577_student_fontys_nl/Documents/Year%20II/SEM%20III/Design%20Project%20Cloud%20Storage/Group%233%20-%20System%20Design%20Document.docx#_Toc124514676)

[Figure 4 - Sequence Diagram Scenario 1 from UC\_002 19](#_Toc124514677)

[Figure 5 - Sequence Diagram Scenario 2 from UC\_002 20](#_Toc124514678)

[Figure 6 - Sequence Diagram Scenario 3 from UC\_002 21](#_Toc124514679)

[Figure 7 - Sequence Diagram Scenario 4 from UC\_002 22](#_Toc124514680)

[Figure 8 – Behaviour of the Main Controller Module 22](#_Toc124514681)

[Figure 9 - Behaviour of the Water Inlet Valve Controller Module 25](https://stichtingfontys-my.sharepoint.com/personal/477577_student_fontys_nl/Documents/Year%20II/SEM%20III/Design%20Project%20Cloud%20Storage/Group%233%20-%20System%20Design%20Document.docx#_Toc124514682)

[Figure 10 – Behaviour of the Heating Controller Module 27](https://stichtingfontys-my.sharepoint.com/personal/477577_student_fontys_nl/Documents/Year%20II/SEM%20III/Design%20Project%20Cloud%20Storage/Group%233%20-%20System%20Design%20Document.docx#_Toc124514683)

[Figure 11 - Behaviour of the Detergent dispenser controller 28](https://stichtingfontys-my.sharepoint.com/personal/477577_student_fontys_nl/Documents/Year%20II/SEM%20III/Design%20Project%20Cloud%20Storage/Group%233%20-%20System%20Design%20Document.docx#_Toc124514684)

# List of Tables

[Table 1 – Version history of the document 3](#_Toc124514718)

[Table 2 – Acronyms through the document 5](#_Toc124514719)

[Table 3 - User Requirements 12](#_Toc124514720)

[Table 4 - Functional Requirements 13](#_Toc124514721)

[Table 5 - Non-functional Requirements 14](#_Toc124514722)

[Table 6 - Use cases vs user requirements coverage 14](#_Toc124514723)

[Table 7 - Functional requirements vs user requirements coverage 15](#_Toc124514724)

[Table 8 - System structure modules and their responsibilities 16](#_Toc124514725)

[Table 9 - Messages communicated by the Detergent dispenser controller 28](#_Toc124514726)

# Document history

Table 1 – Version history of the document

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Status** | **Author** | **Description** | **Remarks** | |
| 1.0.0 | 28.10.2022 | Draft | All Auth | * Formatting of the document * Cover Page * Use Cases * User & System Requirements * Traceability Matrix | DONE | |
| 1.1.0 | 14.11.2022 | Draft | All Auth | * Sequence Diagrams | DONE | |
| 1.2.0 | 15.11.2022-  26.11.2022 | Final Draft | All Auth | * State Machine Diagrams * Use Case Diagram * Intermediate Delivery Feedback Improvement * Final Touches | DONE |
| 1.3.0 | 06.12.2022-  11.01.2022 | Final | All Auth | * Main Controller Diagram * Repair and Improvement on Feedback * WDW (Who Did What) Table * Final Touches and Formatting | DONE |

* ­­ Highlighted in green is the current version on which the document is on.

# WDW [Who Did What]

|  |  |  |
| --- | --- | --- |
| Victor  Covalciuc | Sean-David  Brokke | Andre  Sanao |
| 1. Table 1 | 1. Figure 2 | 1. Figure 5 |
| 1. Table 2 | 1. Figure 8 | 1. Figure 6 |
| 1. Formatting of the document | 1. FR\_005 | 1. Figure 7 |
| 1. Cover Page | 1. UC\_007 | 1. Figure 10 |
| 1. Figure 1 | 1. UR\_006 | 1. UC\_006 |
| 1. UC\_008 |  | 1. UR\_005 |
| 1. Remade Figure 2 |  | 1. FR\_025 |
|  |  | 1. Table 6 |
|  |  | 1. Table 7 |

# Terms, Abbreviations, Acronyms

Table 2 – Acronyms through the document

|  |  |
| --- | --- |
| Term | Enlarged Version of said Term |
| SDD | System Design Document |
| HW | Hardware |

# Introduction

The Smart Washing Machine G2025 (SWM G2025) is a product developed by HomeSmartCo.

company which is specialized in production of smart home devices. This document contains only parts of the system design. The design of other parts and aspects of the system should be made and documented accordingly in this document.

# System description

The SWM G2025 has the basic functions that all washing machines have such as, washing dirty laundry without damaging it and providing safe operation for the user. This Washing Machine system is configured with 4 predefined washing programs:

* quick washing
* dark washing
* cotton washing
* spinning

For **safety** purposes the program will not start if the door is open, or the machine is not connected to the water supply. It also offers the possibility to **add forgotten laundry in the middle of a wash** in a safe manner.

In addition to the primary function of the system, the secondary function is to allow an easier operation whenever user has needs for it. For this purpose, SWM G2025 can:

* **select water fill automatically**: it can determine the right amount of water fill based on the weight of the laundry load
* **select detergent amount automatically**: it can determine the right amount of detergent based on the weight of the laundry load

## Washing programs description

A washing program/cycle is a combination of four washing stages that the machine can do. The four washing stages are specified bellow and restricted to this specific number of stages defined bellow.

* **PREWASH**
  + water fill 40l
  + heating 40°C or no heating
  + detergent from the prewash detergent compartment
  + drum rotation: rotate clockwise 1 min., rotate counter clockwise 1 min; rotation time differs per program
  + rotation speed: regular
  + drain (dirty) water
* **MAIN WASH**
  + water fill 40l or 80l
  + heating 40°C, 60°C or 75°C
  + detergent from the main wash detergent compartment
  + drum rotation: rotate clockwise 1 min., rotate counter clockwise 1 min; rotation time differs per program
  + rotation speed: regular
  + drain (dirty) water
* **RINSE**
  + water fill 40l
  + no heating
  + no detergent
  + drum rotation: rotate clockwise 1 min., rotate counter clockwise 1 min; total rotation time differs per program
  + rotation speed: regular
  + drain (dirty) water
* **SPIN**
  + no water fill
  + keep draining water
  + drum rotation: rotate clockwise, highest speed, 30 seconds, rotate counter clockwise 30 seconds; total rotation time differs per program
  + rotation speed: regular or high

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Program | ProgID | Prewash | main wash | Rinsing | Spin |
| Quick wash | Q | X | - water fill 40l,  - heat 40°C,  - drum total rotation time 15min, | – drum total rotation time  10 min, | - drum total rotation time 5 min  - high speed |
| Dark wash | D | - no heating,  - drum total rotation time 10min | - water fill 40l,  - heat 40°C,  - drum total rotation time 20min, | – drum total rotation time  15 min, | - drum total rotation time 5 min  - high speed |
| Cotton wash | C | - heating 40°C,  - drum total rotation time 15min | - water fill 80l,  - heat 75°C,  - drum total rotation time 30min, | – drum total rotation time  20 min, | - drum total rotation time 5 min  - high speed |
| Spin | S | X | X | x | - drum total rotation time 10 min  - high speed |

In the table below the configurations of the predefined washing programs as made by the manufacturer are given. It is possible for the user to change the heating level, the water fill, and the spin speed of the predefined programs accordingly to the constraints given above. For example, the user can select the “Cotton wash” predefined program and make the following adjustments: main wash water fill 40l, main wash heat 40°C and regular spin speed.

## Safety

To guarantee safety of the user the manufacturer has stated several safety requirements:

* before a washing program starts the door must be locked
* the door cannot be open if the water fill is more than 20l
* the door cannot be open during spin step
* the door cannot be open if water temperature is above 60°C
* the water fill must not exceed 90l
* the water temperature must not exceed 80°C

# System Design

## Use cases

In this section use cases and corresponding requirements are described.

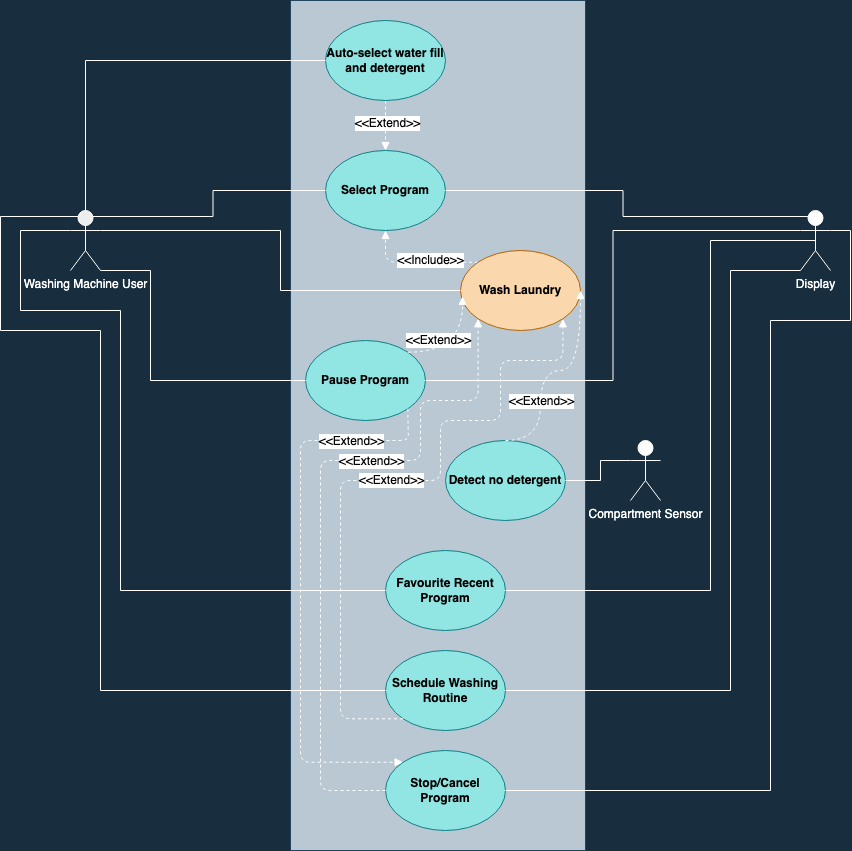


Figure 1 - Use case diagram of Smart washing machine system

**Error! Reference source not found.** depicts some of the use cases of Smart Washing Machine. It shows system boundary, specify how actors interact with system. Each use case will be described in a table.

|  |  |
| --- | --- |
| **Use Case ID:** UC[[1]](#footnote-2)\_001 | **Use Case:** Select a washing program |
| **Description** | When the user wants to use the washing machine, he chooses a washing program from the list of the predefined programs. |
| **Actor:** | A washing machine user, Display |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_002 | **Use Case:** Wash laundry |
| **Description** | The user wants to use the washing machine to wash laundry after a washing program has been selected |
| **Actor:** | A washing machine user |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_003 | **Use Case:** Auto-select water fill and detergent amount |
| **Description** | When the user wants to use the washing machine, he lets the machine choose the water fill based on the weight of the load. |
| **Actor:** | A washing machine user |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_004 | **Use Case:** Detect no detergent |
| **Description** | When the user forgets to put detergent in the pre-wash detergent compartment, or the main wash detergent compartment the washing machine notifies the user. |
| **Actor:** | Compartment Sensor |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_005 | **Use Case:** Pause a running washing program |
| **Description** | When the user wants to add an extra load, he pauses the running washing program. |
| **Actor:** | A washing machine user, Display |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_006 | **Use Case:** Favourite recent program |
| **Description** | When the user wants to favourite a predefined program, ran recently. |
| **Actor:** | A washing machine user, Display |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_007 | **Use Case:** Schedule washing routine |
| **Description** | A user can start a wash routine after a desired time, the user can select a desired time after selecting a washing program. |
| **Actor:** | A washing machine user, Display |

|  |  |
| --- | --- |
| **Use Case ID:** UC\_008 | **Use Case:** Stop/Cancel Program |
| **Description** | When a user wants to end the current running program, he stops the washing program. And it is cancelled automatically. |
| **Actor:** | A washing machine user, Display |

## User requirements

A user requirement describes what a user expects the system to do.

|  |  |  |
| --- | --- | --- |
| **Requirement ID** | **Description** | **Use Cases** |
| **UR**\_001 | A user shall be able to select a program from the predefined washing programs. | UC\_001 |
| **UR**\_002 | A user shall be able to adjust the spin speed, heating level or water fill of the main wash of the selected predefined program before the washing programs starts. | UC\_001 |
| **UR**\_003 | A user shall be able to cancel the just made selection of a washing program. | UC\_001 |
| **UR**\_004 | A user shall be able operate the washing machine in a safe manner. | UC\_001 |
| **UR**\_005 | A user shall be able to store favourite predefined washing program. | UC\_006 |
| **UR**\_006 | A user must be able to set a time to start a washing routine. | UC\_007 |

Table 3 - User Requirements

|  |  |  |
| --- | --- | --- |
| **Functional Requirement ID** | **Description** | **Use Case/UR** |
| **FR**\_001 | The predefined washing programs are pre-configured (temperature, time, …) and stored | UC\_001/UR\_001 |
| **FR**\_002 | The system must be able to execute all predefined programs as configured | UC\_001/UR\_001 |
| **FR**\_003 | The system must be able to cancel the already made selection of a washing program within 1 min after the selection has been done | UC\_001/ UR\_003/ UC\_007/ UR\_006 |
| **FR**\_004 | The system must be able to set and measure time | UC\_001/UR\_003  UC\_002/UC\_007/ UR\_006 |
| **FR**\_005 | The system must be able to start a washing routine after the washing routine timer has finished counting down. | UC\_001/UC\_007 |
| **FR**\_006 | The system must be able to set the spin speed |  |
| **FR**\_007 | The system must be able to change the spin speed of the pre-defined programs. | UC\_001/UR\_002 |
| **FR**\_008 | The system must be able to control the start, stop, speed and direction of the drum |  |
| **FR**\_009 | The system must be able to set and measure the rotation speed of the drum |  |
| **FR**\_010 | The system must be able to determine the rotation direction of the drum |  |
| **FR**\_011 | The system must be able to change the heating level of the pre-defined programs | UC\_001/UR\_002 |
| **FR**\_012 | The system must be able to set and measure the heating level |  |
| **FR**\_013 | The system must be able to control the heating level |  |
| **FR**\_014 | The system must be able to change the water fill level of the pre-defined programs | UC\_001/UR\_001 |
| **FR**\_015 | The system must be able to set and measure the water fill level |  |
| **FR**\_016 | The system must be able to control the water fill level |  |
| **FR**\_017 | The system must be able to control the detergent dispensers |  |
| **FR**\_018 | The system must be able to detect the absence of detergent in the detergent compartments | UC\_004/ |
| **FR**\_019 | The system must be able to control the door lock | UC\_001 |
| **FR**\_020 | The system must be able to detect the status of the door: open/closed |  |
| **FR**\_021 | The system must be able to control water drain | UC\_001/UR\_001 |
| **FR**\_022 | The system must be able to indicate when a washing program starts | UC\_002 |
| **FR**\_023 | The system must be able to indicate when a washing program is finished | UC\_002 |
| **FR**\_024 | The system must be able to indicate when a washing program is running | UC\_002 |
| **FR**\_025 | The system must be able to store the previous configurations of the user’s predefined washing programs. | UC\_006 |

Table 4 - Functional Requirements

|  |  |  |
| --- | --- | --- |
| **Non-functional Requirement ID** | **Description** | **Use Case/UR** |
| **NR**\_001 | The system clearly indicates when a washing program starts | UC\_001/UC\_002 |
| **NR**\_002 | The system clearly indicates when a washing program is finished | UC\_001/UC\_002 |
| **NR**\_003 | The system clearly indicates when a washing program is running | UC\_001/UC\_002 |
| **NR**\_004 | The predefined washing programs are clearly visible and easily selectable | UC\_001/UR\_001/ UC\_006 |
| **NR**\_005 | The pre-configuration of a predefined washing program is clearly visible | UC\_001/UR\_001 |
| **NR**\_006 | The selected washing program is clearly indicated | UC\_001/UR\_001/ UC\_006 |
| **NR**\_007 | The adjustments made by the user on the pre-defined program configuration are clearly visible | UC\_001/UR\_002 |

Table 5 - Non-functional Requirements

## Requirements traceability matrix

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | UC\_001 | UC\_002 | UC\_003 | UC\_004 | UC\_005 | UC\_006 | UC\_007 | UC\_008 | UC\_009 |
| UR\_001 | V |  |  |  |  |  |  |  |  |
| UR\_002 | V |  |  |  |  |  |  |  |  |
| UR\_003 | V |  |  |  |  |  |  |  |  |
| UR\_004 |  |  |  |  |  |  |  |  |  |
| UR\_005 |  |  |  |  |  | V |  |  |  |
| UR\_006 |  |  |  |  |  |  | V |  |  |

Table 6 - Use cases vs user requirements coverage

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | UR\_001 | UR\_002 | UR\_003 | UR\_004 | UR\_005 | UR\_006 |
| FR\_001 | V |  |  |  |  |  |
| FR\_002 | V |  |  |  |  |  |
| FR\_003 |  |  | V |  |  | V |
| FR\_004 |  |  | V |  |  | V |
| FR\_005 |  |  |  |  |  |  |
| FR\_006 |  |  |  |  |  |  |
| FR\_007 |  | V |  |  |  |  |
| FR\_008 |  |  |  |  |  |  |
| FR\_009 |  |  |  |  |  |  |
| FR\_010 |  |  |  |  |  |  |
| FR\_011 |  | V |  |  |  |  |
| FR\_012 |  |  |  |  |  |  |
| FR\_013 |  |  |  |  |  |  |
| FR\_014 | V |  |  |  |  |  |
| FR\_015 |  |  |  |  |  |  |
| FR\_016 |  |  |  |  |  |  |
| FR\_017 |  |  |  |  |  |  |
| FR\_018 |  |  |  |  |  |  |
| FR\_019 |  |  |  |  |  |  |
| FR\_020 |  |  |  |  |  |  |
| FR\_021 | V |  |  |  |  |  |
| FR\_022 |  |  |  |  |  |  |
| FR\_023 |  |  |  |  |  |  |
| FR\_024 |  |  |  |  |  |  |
| FR\_025 |  |  |  |  |  |  |

Table 7 - Functional requirements vs user requirements coverage

# System structure

|  |  |
| --- | --- |
| **Module name** | **Responsibilities** |
| Main controller | Executes the washing programs according to their pre-configuration or according to changed made by the user, and guarantees safe operation of the washing machine |
| Control panel | Receives external input and sends system output as described by the requirements |
| Door  controller | Opens/closes/locks the door and has monitoring functionality (door sensor) |
| Drum controller | Controls the direction and speed of the drum motor |
| Water inlet valve controller | Controls the water flow into the machine and has monitoring functionality (water sensor level) |
| Heating  controller | Controls the water heating element and measures water temperature |
| Drain  controller | Controls the water pump that sinks water from the machine (into the drain hose) |
| Detergent dispenser controller | Controls the clips of the prewash and main wash compartments and has monitoring functionality (detects presence of detergent and weights the amount of detergent in the compartments) |
| Timer | Counts time and generates timeouts; it is sub-module of other modules |

Table 8 - System structure modules and their responsibilities

# System behaviour

## Sequence Diagrams

In this section different scenarios are considered per use case and the corresponding interactions are described by sequence diagrams. The modules from the previous section are used as objects in these diagrams. In some sequence diagrams the actors from the use cases appear to be an object in a diagram, since they Interact with the system by sending or receiving messages.

UC\_001: Select a washing program

There are different scenarios for this use case depending on which washing program the user chooses, but they can be classified in two types of scenarios: (1) when the user of the washing machine selects a pre-defined program without changing any of the configured parameters, and (2) when the user decides to change spin speed and heating level of a selected pre-defined programs. Two concrete scenarios are chosen and described by a sequence diagram.

Scenario 1: The user selects cotton wash predefined program just as it is configured: prewash heating level of 40°C and high-speed spin. The sequence diagram of this scenario is given in Figure 2.

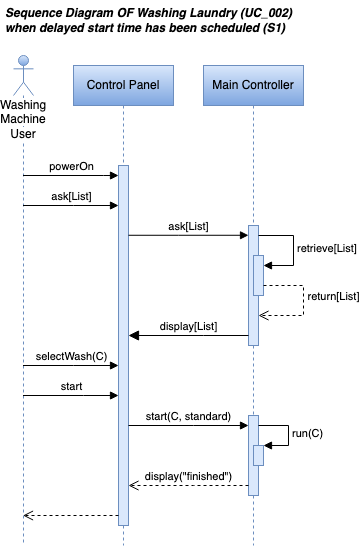


Figure 2 - Sequence Diagram Scenario 1 from UC\_001

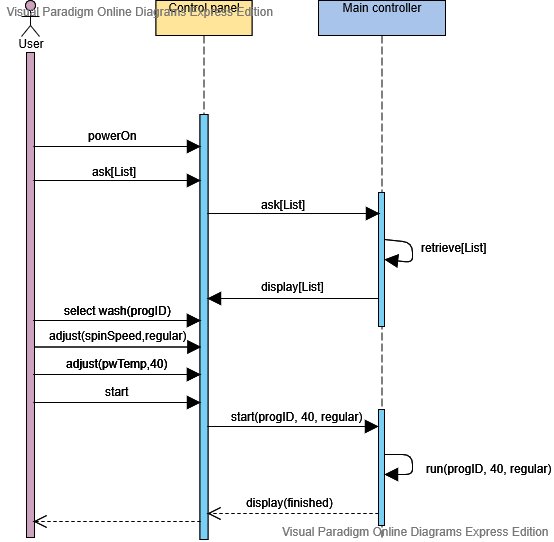
Scenario 2: The user selects cotton wash predefined program but makes the following changes: no heating in the prewash step, heating level of the main wash of 40°C and regular speed spin. The sequence diagram of this scenario is given in the bellow Figure 3.

Figure 3 - Sequence Diagram Scenario 2 from UC\_001

UC\_002: Wash laundry

Use Case 2 will contain different scenarios depending on which washing program the user chooses, therefore these can be good and bad scenarios. Given so, the “Wash Laundry” use case will contain five types of scenarios. Scenario (1) which is when the user selects a predefined washing program and schedules time to initiate the washing process later. Scenario (2) which is when the user selects a preconfigured favourite program without having to configuring all the settings again. Scenario (3) which is when the user starts to create a new program and saves it as a favourite one. Scenario (4) which is when the user selects the favourite program, but it has not been preconfigured yet.

Diagram, box and whisker chart

Description automatically generatedScenario 1: The user selects the cotton wash predefined program and schedule a timer to start the washing later the day. The sequence diagram is given below in Figure 4.

Figure 4 - Sequence Diagram Scenario 1 from UC\_002

Diagram

Description automatically generatedScenario 2: The user selects favorite predefined program just as it is configured by the users likings. The sequence diagram is given below in Figure 5.

Figure 5 - Sequence Diagram Scenario 2 from UC\_002

Diagram

Description automatically generatedScenario 6: The user creates a new favorite program just as it is configured: prewash heating level of 40°C, spin speed of regular and main wash heating level of 60°C. The sequence diagram is given below in Figure 6.

Figure 6 - Sequence Diagram Scenario 3 from UC\_002

Diagram

Description automatically generatedScenario 7: The user selects favourite predefined program, but the display shows that there is currently no saved favourite program. The sequence diagram is given below in Figure 7.

Figure 7 - Sequence Diagram Scenario 4 from UC\_002

## State Machines

In this section the behaviour of the previously identified system modules is described by means of state machines. The modules exchange information with each other, which is indicated as a received message or sent message. Messages can contain parameters that carry the data that are exchanged between different modules.

### Main controller

(SVG Format)  (PNG Format)

Figure 8 – Behavior of the Main Controller Module

Above is the state machine diagrams of the main controller, due to the size of the diagram inserting it would have made it unreadable, therefore it is inserted as a “link” in one of the icons.

The above Figure 8 shows a state machine diagram that outlines the Main Controller behaviour in relation with the other modules some of which are expanded on further in the current chapter.

|  |  |  |
| --- | --- | --- |
| Direction | Message name | Description |
| INCOMING MESSAGES | ask[List] | Command used to retrieve the list of predefined washing programs from the main controller. |
|  | | |
| OUTGOING  MESSAGES | display [List] | Sends a list of predefined programs to the control panel. |
|  | send(wivc,program.water\_lvl) | Sends **program. water\_lvl** (selected program’s water level) to **wivc** (water inlet valve controller) |
|  | send (hc, program.temp) | Sends **program.temp** (selected program’s heating temperature level) to **hc** (heater controller) |
|  | send  (dc,program.dr\_rotation,program.dr\_time) | Sends **program.dr\_rotation** (*set program’s drum rotation on*) and **program.dr\_time** (program’s *drum rotation time*) to **dc** (*drum controlle*r) |
|  | send (dc,program.stopDrumRotation()) | Sends **program.stopDrumRotation** (*set program’s drum rotation off*) to **dc** (*drum controlle*r) |
|  | send (hc,program.stopHeating()) | Sends **program.stopHeating** (*set program’s heater off*) to **hc** (*heater controlle*r) |
|  | send (cp, err\_msg) | Sends **err\_msg** (*send error message*) to **cp** (*controll panel*) |
|  | send (cp, goBackToMainMenuOption) | Sends message with option for user to go back to main menu |
|  | send(cp, finished) | Sends **Finished** (*completion message*) to **cp** (*controll panel*) informing user that the waching machine has finished washing cycle |
|  | | |
| Local msg | progID () | Check if a program was selected. |
|  | doorClosed() | Check if washing machine door is in closed state |
|  | cancelProgramDetected() | Condition that is true when user cancels washing program |
|  | program.dr\_TimeExpired() | Condidtion that is true when a **program.dr\_time** (program’s *drum rotation time*) has expired |
|  | backToMainMenu() | Condition that is true when user selects option to go to main menu from the control panel |
|  |  |  |

### Control panel

### Door controller

### Drum controller

### Diagram, schematic Description automatically generatedWater inlet valve controller

Figure 9 - Behaviour of the Water Inlet Valve Controller Module

The above Figure 9 shows a state machine diagram that outlines the Water Inlet Valve Controller behaviour.

|  |  |  |
| --- | --- | --- |
| Direction | Message name | Description |
| INCOMING MESSAGES | start (mc, water\_val) | The main controller initializes the water inlet valve controller from the main controller **(mc)** and passes the amount **(water\_val)** of water needed to be let in for the wash. |
|  | | |
| OUTGOING  MESSAGES | send (ack\_notification) | The water inlet valve controller acknowledges to the main controller after a message has been received. |
|  | send (yes\_water) | The water inlet valve controller acknowledges to the main controller that water is present. |
|  | send (no\_water) | The water inlet valve controller acknowledges to the main controller that there is no water present. |
|  | send (filled\_water) | The water inlet valve controller acknowledges to the main controller that the indicated amount of water i.e. *water\_val* has been let in. |
|  | | |
| Local msg | open\_valve () | The water inlet valve controller sends an open command to the HW component that controls the inlet part of the valve. |
|  | close\_valve () | The water inlet valve controller sends a close command to the HW component that controls the inlet part of the valve. |
|  | check\_valve () | The water inlet valve controller samples the flow of water from the valve, does calculations and returns the amount of water that has been let in. |
|  | senseWater () | The water inlet valve controller samples for water before the vales and returns the presence of water in the form of a truth value. |
|  |  |  |

### Heating controller

Diagram

Description automatically generatedThe state machine of the heating controller is given in the figure below. It communicates with the main controller as it follows:

Figure 10 – Behaviour of the Heating Controller Module

|  |  |  |
| --- | --- | --- |
| Direction | Message name | Description |
| In msg | start(mc, value) | The main controller **(mc)** initializes the heating controller and sets the selected value **(value)** needed for the wash. |
|  |  |  |
| Out (send) msg | set\_value | The heating controller acknowledge to the main controller, that the value is set. |
|  | received\_temp | The heating controller acknowledge to the main controller, that the set temperature has been received. |
|  |  |  |
| Local msg | cancel\_timer | The timer has been cancelled because a temperature value has been selected. |

### Drain controller

### Detergent dispenser controller

The state machine of the detergent dispenser controller is given in the figure below. It communicates with the main controller as it follows:

|  |  |  |
| --- | --- | --- |
| Direction | Message name | Description |
| In msg | start(pw,amount) | The main controller initializes the detergent disp. controller in the prewash mode **(pw)** and passes the amount **(amount)** of detergent needed for the wash |
|  | start(mw, amount) | The main controller initializes the detergent disp. controller in the main wash mode **(mw)** and passes the amount **(amount)** of detergent needed for the wash |
|  | close(pw) | The main controller sends a command to the detergent dispenser controller to close the prewash **(pw)** compartment |
|  | close(mw) | The main controller sends a command to the detergent dispenser controller to close the main wash **(mw)** compartment |
|  |  |  |
| Out (send) msg | opened | The detergent dispenser acknowledge to the main controller compartment is open |
|  | closed | The detergent dispenser acknowledge to the main controller compartment is closed |
|  |  |  |
| Local msg | open\_clip1 | The detergent dispenser controller sends open/close command to the HW component that controls the clip of the compartment 1 |
|  | close\_clip1 |
|  | open\_clip2 | The detergent dispenser controller sends open/close command to the HW component that controls the clip of the compartment 2 |
|  | close\_clip2 |

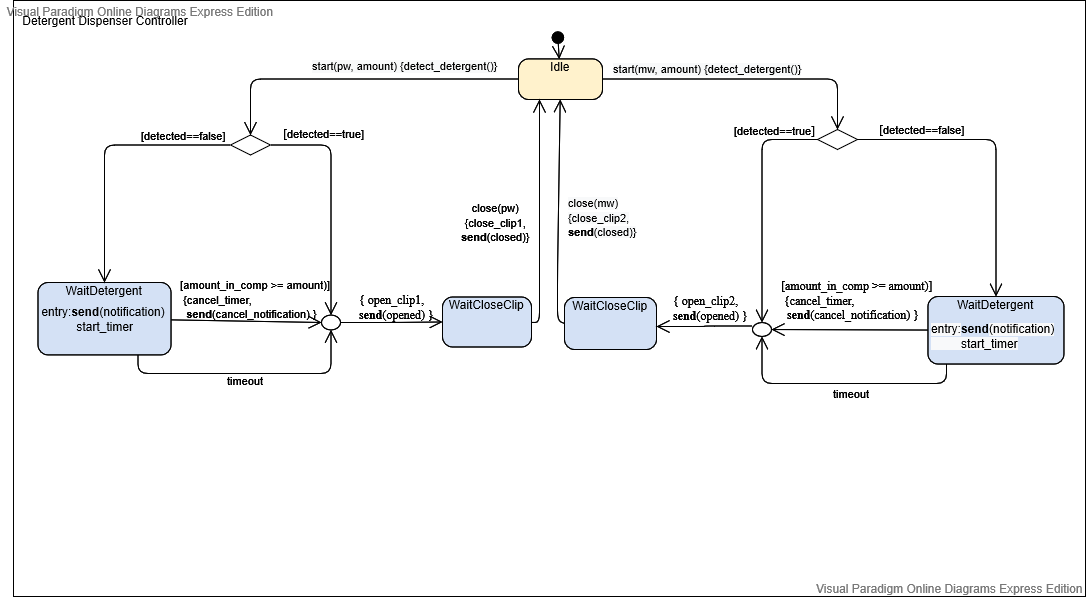
Table 9 - Messages communicated by the Detergent dispenser controller

Figure 11 - Behaviour of the Detergent dispenser controller

# Recommendations and conclusion

To conclude, after brainstorming to expanding the system design document for a smart washing machine, we put our ideas in motion and created new use cases that would be fit to the current ones. Given so and with the guidance of the teacher, we updated the relation of the user requirements and the functional requirements, fixed some type errors during the process. Moreover, we mapped some of the current behaviour of the product through diagrams such as state machine and sequence diagrams.

At last, following the execution of this product, and design of it, as much as we would like to believe that perfect is the word par excellence to describe our assignment, improvement can still be made and will always be there to be made. As such one of the few points which we would have preferred to have improved on, was to have mapped more of the washing machine’s behavior and to have found more use-cases, functional requirements and user requirement. Also, to maybe have expanded the functionality of the product. Nevertheless, this point ought to be achieved sometime in the future, as it is of great value to our skills and to the efficiency of constructed apparatus.

# References

1. *UML Sequence Diagram Tutorial*
   * <https://www.lucidchart.com/pages/uml-sequence-diagram>
2. *Introduction to UML*, Sparx System Training,
   * <https://www.sparxsystems.eu/resources/project-development-with-uml-and-ea/>
3. *Visual Paradigm Online*
   * <https://online.visual-paradigm.com/>
4. *How to Code a State Machine in C or C++*, Barr Group Software Experts,
   * <https://barrgroup.com/embedded-systems/how-to/coding-state-machines> (posted on 2016-05-04 by Miro Samek)
5. *UML @ Classroom: An Introduction to Object-Oriented Modeling,* M. Seidl, et.al
   * <https://link.springer.com/content/pdf/10.1007%2F978-3-319-12742-2.pdf>
6. *Diagrams Online*
   * [*https://app.diagrams.net/*](https://app.diagrams.net/)

1. UC = Use Case [↑](#footnote-ref-2)