|  |
| --- |
|  |
| System Design Document  Smart Washing Machine  (Add name of the students) |
| |  |  |  | | --- | --- | --- | | Xxxx | 6/29/20 | Project 1 | |

Contents

[Document history 3](#_Toc45886307)

[Terms, Abbreviations 3](#_Toc45886308)

[1. Introduction 4](#_Toc45886309)

[2. System description 4](#_Toc45886310)

[2.1 Washing programs description 4](#_Toc45886313)

[2.2 Safety 5](#_Toc45886314)

[3. System Design 6](#_Toc45886315)

[3.1 Use cases (Left to the students to extend/complete it) 6](#_Toc45886319)

[3.2 User requirements: 7](#_Toc45886320)

[3.3 Requirements traceability matrix (not complete \*It is left for students to complete it) 9](#_Toc45886321)

[4. System structure 10](#_Toc45886322)

[5. System behaviour 11](#_Toc45886323)

[5.1 Sequence Diagrams (Left to students to add more diagrams for other uc’s) 11](#_Toc45886326)

[5.2 State Machines (Left to students to work it further out) 12](#_Toc45886327)

[5.2.1 Main controller 12](#_Toc45886328)

[5.2.2 Control panel 12](#_Toc45886329)

[5.2.3 Door controller 12](#_Toc45886330)

[5.2.4 Drum controller 12](#_Toc45886331)

[5.2.5 Water inlet valve controller 12](#_Toc45886332)

[5.2.6 Heating controller 12](#_Toc45886333)

[5.2.7 Drain controller 12](#_Toc45886334)

[5.2.8 Detergent dispenser controller 12](#_Toc45886335)

[6. Recommendations and conclusion 13](#_Toc45886336)

[7. References 13](#_Toc45886337)

List of Figures

[Figure 1. Use case diagram of Smart washing machine system 6](#_Toc45886355)

[Figure 2. Sequence Diagram Scenario 1 from UC\_001 11](#_Toc45886356)

[Figure 3. Sequence Diagram Scenario 2 from UC\_001 12](#_Toc45886357)

[Figure 4. Behavior of the Detergent dispenser controller represented by a state machine 13](#_Toc45886358)

List of Tables

[Table 1. User Requirements 7](#_Toc45886369)

[Table 2. Functional Requirements 8](#_Toc45886370)

[Table 3. Non-functional Requirements 9](#_Toc45886371)

[Table 4: Use cases vs user requirements coverage 9](#_Toc45886372)

[Table 5: Functional requirements vs user requirements coverage 9](#_Toc45886373)

[Table 6. System structure modules and their responsibilities 10](#_Toc45886374)

[Table 7. Messages communicated by the Detergent dispenser controller 13](#_Toc45886375)

# Document history (update by every update of the document)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Date** | **Status** | **Author** | **Description** | **Remarks** |
| 0.1 | 2023-02-06 | Draft | Xxx | Creation |  |
|  |  |  |  |  |  |

­­

# Terms, Abbreviations

|  |  |
| --- | --- |
|  |  |
| SDD | System Design Document |

# 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 primary functionality that all washing machines have: washing dirty laundry without damaging it and providing safety operation to the user. It 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 functionality of the system, the secondary functionality 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 amount based on the weight of the laundry load



## Washing programs description

A washing program is a combination of 4 cycles the washing machine can do. The 4 cycles are specified and constrained as it follows:

* 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; 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; rotation time differs per program
  + rotation speed: regular or high

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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |
| Spinning | S | X | X | x | - drum total rotation time 10 min  - high 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 cycle
* 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 (Task: add 2 use cases UC\_006 and UC\_007)

In this section use cases and corresponding requirements are described.

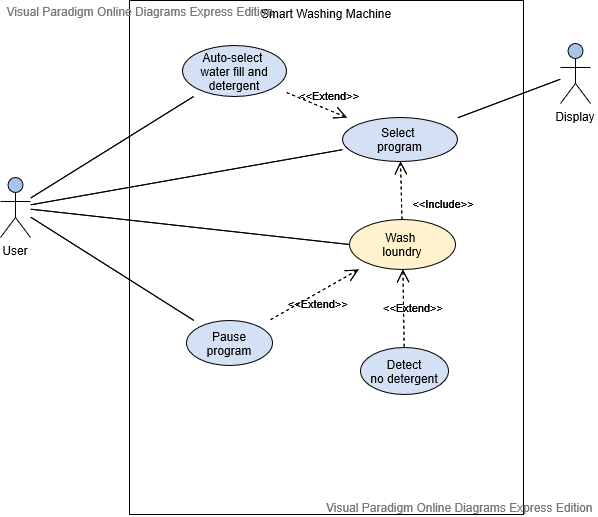


Figure 1. Use case diagram of Smart washing machine system

Figure 1 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\_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 |

|  |  |
| --- | --- |
| **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:** |  |

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

## User requirements (Task: add user, functional and non-functional requirements accordingly to the added use cases)

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**\_003 | A user shall be able operate the washing machine in a safe manner. |  |

Table 1. User Requirements

|  |  |  |
| --- | --- | --- |
| **Functional Requirement ID** | **Description** | **Use Case/UR** |
| **FR**\_001 | The predefined washing programs are pre-configured (temp, 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 |
| **FR**\_004 | The system must be able to set and measure time | UC\_001/UR\_003  UC\_002/ |
| **FR**\_005 | The system must be able to set the spin speed |  |
| **FR**\_006 | The system must be able to change the spin speed of the pre-defined programs. | UC\_001/UR\_002 |
| **FR**\_007 | The system must be able to control the start, stop, speed and direction of the drum |  |
| **FR**\_008 | The system must be able to set and measure the rotation speed of the drum |  |
| **FR**\_009 | The system must be able to determine the rotation direction of the drum |  |
| **FR**\_010 | The system must be able to change the heating level of the pre-defined programs | UC\_001/UR\_002 |
| **FR**\_011 | The system must be able to set and measure the heating level |  |
| **FR**\_012 | The system must be able to control the heating level |  |
| **FR**\_013 | The system must be able to change the water fill level of the pre-defined programs | UC\_001/UR\_001 |
| **FR**\_014 | The system must be able to set and measure the water fill level |  |
| **FR**\_015 | The system must be able to control the water fill level |  |
| **FR**\_016 | The system must be able to control the detergent dispensers |  |
| **FR**\_017 | The system must be able to detect the absence of detergent in the detergent compartments | UC\_004/ |
| **FR**\_018 | The system must be able to control the door lock |  |
| **FR**\_019 | The system must be able to detect the status of the door: open/closed |  |
| **FR**\_020 | The system must be able to control water drain | UC\_001/UR\_001 |
| **FR**\_021 | The system must be able to indicate when a washing program starts | UC\_002 |
| **FR**\_022 | The system must be able to indicate when a washing program is finished | UC\_002 |
| **FR**\_023 | The system must be able to indicate when a washing program is running | UC\_002 |

Table 2. 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 |
| **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 |
| **NR**\_007 | The adjustments made by the user on the pre-defined program configuration are clearly visible | UC\_001/UR\_002 |

Table 3. Non-functional Requirements

## Requirements traceability matrix (Make a traceability matrix for the added use cases)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | UC\_001 | UC\_002 | UC\_003 | UC\_004 | UC\_005 | … |
| UR\_001 | v |  |  |  |  |  |
| UR\_002 | V |  |  |  |  |  |
| UR\_003 | V |  |  |  |  |  |
| …. |  |  |  |  |  |  |

Table 4: Use cases vs user requirements coverage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | UR\_001 | UR\_002 | UR\_003 | … |
| FR\_001 | V |  |  |  |
| FR\_002 | V |  |  |  |
| FR\_003 |  |  | v |  |
| FR\_004 |  |  | v |  |
| FR\_005 |  |  |  |  |
| FR\_006 |  |  |  |  |
| FR\_007 |  |  |  |  |
| FR\_008 |  |  |  |  |
| FR\_009 |  |  |  |  |
| FR\_010 |  |  |  |  |
| FR\_011 |  |  |  |  |
| FR\_012 |  |  |  |  |
| FR\_013 |  |  |  |  |
| … |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table 5: 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 mainwash 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 6. System components and their responsibilities

# System behaviour



## Sequence Diagrams

## (Task: Describe 3 scenarios from use cases of your choice. 1 scenario is a “happy flow”, and 2 scenarios are “unhappy flows”. These scenarios can involve any components of the system.)

In this section different scenarios are considered and the corresponding interactions between the involved components are described by sequence diagrams. Some sequence diagrams show how an actor (for instance the user) interacts 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.

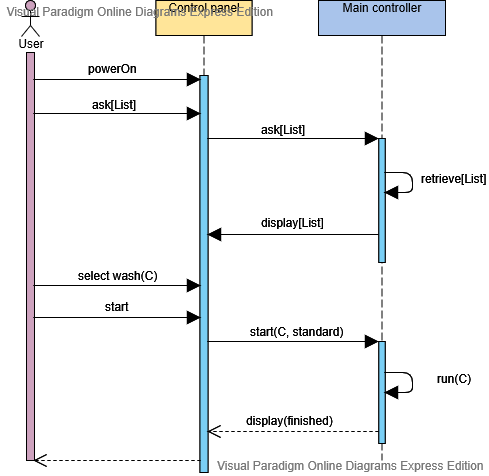


Figure 2. Sequence Diagram Scenario 1 from UC\_001

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 Figure 3.

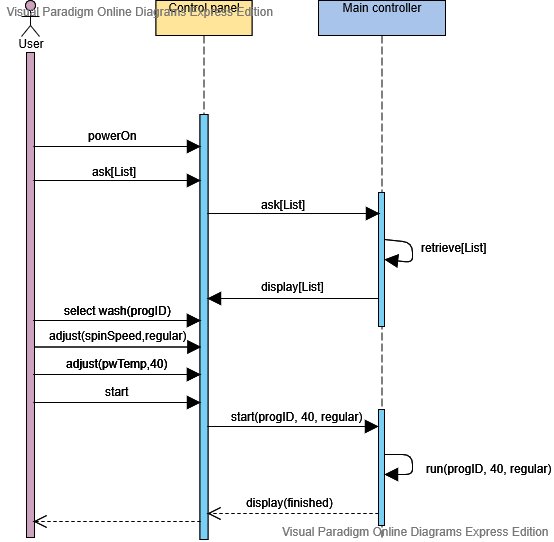


Figure 3. Sequence Diagram Scenario 2 from UC\_001

## State Machines

## (Task: First, analyse the given detergent dispenser example 5.5.8 and understand it properly, its state machine and the table of messages of this component. Reason how the whole system functions and how each component is involved in it. Then, sketch the state machine and the table of messages for the Main controller in 5.5.1. . Then, choose 2 of these components: Door controller, Drum controller. Water inlet valve controller and Heating controller and draw the state machines and tables of messages for the two chosen components in the corresponding sections below.

## Note: It is important that your design is **consistent**. This means that: 1. the scenarios described by sequence diagrams must be consistent with the behaviour of the state machines, 2. the messages in the tables must be consistent with the messages in the state machines. )

In this section the behaviour of the previously identified system components is described by means of state machines. The components exchange messages 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 components.

### Main controller

### Control panel

### Door controller

### Drum controller

### Water inlet valve controller

### Heating controller

### 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(pm,amount) | The main controller initialize 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 initialize 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 mainwash **(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 |
|  | notification | The detergent dispenser send notification the main controller, for instance “No detergent”. |
|  | cancel\_notification | The detergent dispenser cancels notification the main controller. |
|  |  |  |
| 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 7. Messages communicated by the Detergent dispenser controller

Keep in mind that this is one of many possible ways how the Detergent Dispenser controller may behave, and therefore one of many possible designs for this component.

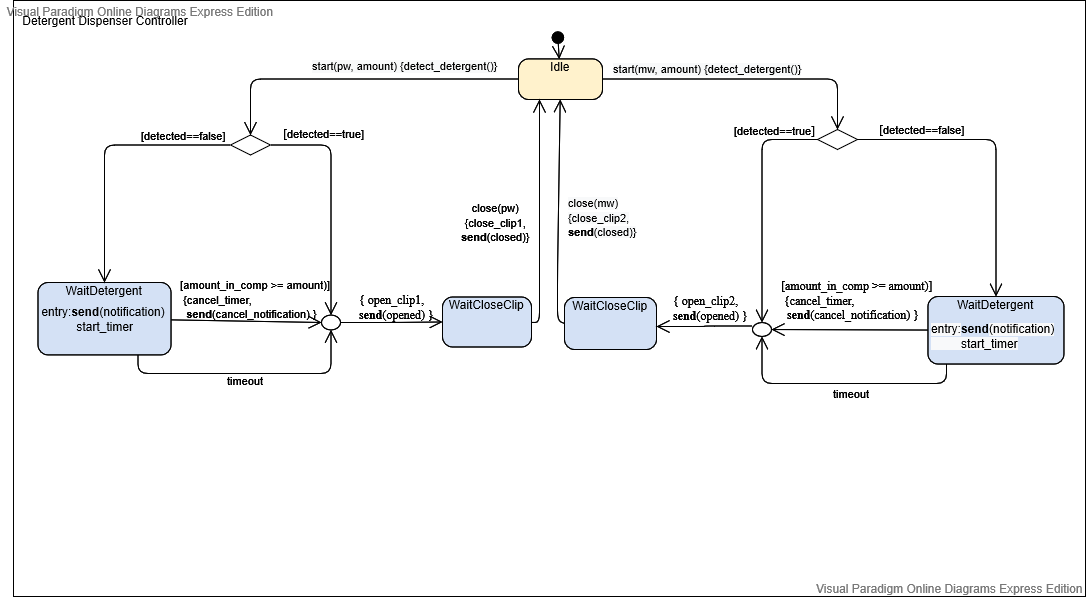


Figure 4. Behavior of the Detergent dispenser controller represented by a state machine

# Recommendations and conclusion (Task: Give your recommendations and conclusions regarding the design. )

# Reflection (Task: Every student in the group writes own reflection on what he/she has done and has learned from this project.)

# References (Task: Add any references that you have used for this assignment. )

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