

Relazione progetto

SMART LAUNDRY- RAD

CORSO DI LAUREA IN INFORMATICA

Ingegneria del software – A.A. 2022/2023

Componenti del gruppo:

Simone Palladino - 0124002316

Luca Tartaglia - 0124002294

Mattia Di Palma - 0124002448

Sommario

Introduzione.....	3
Sistema corrente	4
Sistema proposto	4
Panoramica (obiettivi del sistema).....	5
Requisiti funzionali (FR).....	6
Requisiti non funzionali (NFR).....	7
Vincoli	8
Modelli del sistema	9
Scenari.....	13
Scenario “impostaParametriLavaggio”	13
Scenario “controllaStatoDispositivo”	13
Scenario “esecuzioneTecnicheOttimizzazione”	13
Scenario “richiedeAnnullamentoProgramma”	14
Scenario “controllaCondizioniMeteo”	14
Scenario “attivaCiclo”	14
Scenario “calcoloPesoBucato”	15
Modello casi d’uso	16
Tabelle casi d’uso	16
Schema casi d’uso	34
Modello ad oggetti.....	35
Dizionario dei dati	35
Diagramma delle classi	37
Modelli dinamici	38
Diagrammi delle attività	38
Diagrammi delle sequenze	41
Interfaccia utente.....	45
Obiettivi del test	45
Realizzazione del prototipo	45
Metodologia usata	58
Sintesi delle misure	60
Analisi dei risultati	62
Sintesi delle intervistate	63
Raccomandazioni finali.....	65
Allegati	66
Valutazione dell’usabilità	67
Glossario.....	69

Introduzione

Il progetto descritto richiede lo sviluppo di un software in grado di gestire le funzioni di un elettrodomestico che comprende quattro dispositivi. Il processo di sviluppo del software deve essere rapido e il software stesso deve essere altamente flessibile, in modo da poter gestire diverse funzioni in modo efficace.

Le funzioni che il software deve gestire sono:

- Permettere la gestione singola di ogni funzione
- Permettere la gestione automatica che andrà a controllare più funzioni attivate (due, tre o tutte e quattro)
- La gestione automatica deve gestire il funzionamento contemporaneo di più funzioni armonizzandole al fine di evitare il superamento di una certa soglia di assorbimento di energia elettrica impostata dall'utente, ad esempio 2300 watt/ora
- Permettere di impostare le date di fine lavaggio e fine asciugatura di due diversi carichi di bucato, messi l'uno a lavare e l'altro ad asciugare contemporaneamente.
- Gestire le attività contemporanee di lavaggio e asciugatura rispettando le date richieste dall'utente, con il minor consumo di energia.
- Valutare se le date richieste dall'utente sono realizzabili, in caso contrario, non accettare tali date ma proporre la data o le date più vicine a quelle richieste.
- Indicare per ogni soluzione scelta il consumo totale di energia previsto

Tramite appositi sensori, il software deve essere capace di:

- Controllare temperatura aria ambiente
- Controllare umidità relativa aria ambiente
- Stabilire la tipologia e il peso del bucato, sia quello da asciugare e sia quello da lavare

Sistema corrente

La scoperta dei requisiti e il conseguente sviluppo del sistema introdotto parte da zero e non si basa su sistemi precedentemente esistenti. Si parla quindi di Greenfield Engineering, ovvero:

- Il progetto viene sviluppato da una base vuota (“blank”).
- Nessun sistema precedente viene utilizzato per lo sviluppo.
- I requisiti vengono estratti dagli utenti.

Sistema proposto

In questo capitolo si prendono in considerazione gli obiettivi e gli scopi del sistema che verranno collegati in quelli che risultano essere i requisiti funzionali e i requisiti non funzionali, rispettivamente FR e NFR.

Gli FR riguardano risultati particolari del sistema quando un'attività viene eseguita, mentre gli NFR forniscono un comportamento complessivo del sistema stesso.

Verranno inoltre messi in evidenza aspetti riguardanti l'interfacciamento del sistema, i fattori umani, prestazionali, di sicurezza e di qualità. A questo punto saranno introdotti i vincoli del sistema, anche detti pseudo-requisiti; i modelli del sistema tra i quali figurano il diagramma dei casi d'uso, il diagramma delle classi, i diagrammi delle sequenze e i diagrammi delle attività, analizzando il tutto maniera esaustiva. Infine, verranno mostrati aspetti implementativi di un'ipotetica interfaccia utente, prototipi realizzati e la documentazione realizzata per i test di usabilità condotti.

Panoramica (obiettivi del sistema)

Gli obiettivi del sistema riguardano la gestione di un unico software che coordina quattro diversi dispositivi, ovvero:

- Dispositivo per il lavaggio bucato.
- Dispositivo per l'asciugatura bucato.
- Dispositivo per la deumidificazione dell'aria.
- Dispositivo per il riscaldamento dell'aria.

Il software permette la gestione singola o parallela di ogni funzione, l'utente ha la possibilità di impostare la tipologia di programma e la data di terminazione di ogni comando attraverso un'interfaccia grafica.

Gli input inseriti verranno valutati per controllarne la validità e la coerenza da un controller dedicato, chiamato smart controller, in caso di risposta negativa il software dovrà in modo automatico proporre un input realizzabile vicino a quello inserito dall'utente.

Il sistema deve inoltre essere in grado di accedere sia ai dati geografici, ambientali inseriti durante la configurazione e sia ai dati sensoriali riguardanti la temperatura e l'umidità dell'ambiente.

Il software sfruttando i sensori calcolerà in base alla tipologia di materiale, il peso del bucato fornendo un consumo preventivo di energia.

Infine, il software deve rilevare e gestire situazioni atipiche al fine di rendere ottimale l'esperienza dell'utente. L'obiettivo complessivo del sistema è quindi quello di fornire una gestione efficiente di ogni dispositivo minimizzando i costi energetici e ottimizzando i tempi produttivi garantendo all'utente la massima resa per ogni componente.

Requisiti funzionali (FR)

FR1: Il software deve consentire la gestione singola di ogni funzione dell'elettrodomestico, come il lavaggio, l'asciugatura, la deumidificazione e il riscaldamento.

FR2: Il software deve consentire la gestione automatica di più funzioni contemporaneamente.

FR3: Il software deve permettere la configurazione delle date di fine lavaggio e asciugatura di due diversi carichi di bucato messi a lavare e ad asciugare contemporaneamente.

FR4: Il software deve valutare la fattibilità delle date richieste dall'utente e proporre le date più vicine in caso contrario.

FR5: Il software deve controllare la temperatura e l'umidità dell'aria ambiente e utilizzare queste informazioni per regolare le funzioni di deumidificazione e riscaldamento.

FR6: Il software deve monitorare il consumo di energia elettrica dell'elettrodomestico e visualizzare il consumo totale di energia previsto per ogni soluzione scelta dall'utente.

FR7: Il software deve gestire le aperture nel locale in cui è collocato l'elettrodomestico, indicando quali aperture vanno considerate chiuse e quali aperte.

Requisiti non funzionali (NFR)

NFR1: Il software deve essere facile da usare, con un'interfaccia utente intuitiva e ben progettata.

NFR2: Il software deve essere affidabile e robusto, in grado di gestire possibili interruzioni di corrente o altre situazioni di emergenza.

NFR3: Il software deve essere veloce e reattivo, in grado di rispondere rapidamente alle richieste dell'utente.

NFR4: Il software deve essere compatibile con diversi dispositivi, come smartphone, tablet e computer.

Vincoli

- Il sistema deve essere progettato nel linguaggio di programmazione java.
- Il sistema deve essere in grado di effettuare più operazioni contemporaneamente.
- Il sistema deve essere in grado di valutare le richieste effettuate dall'utente in base all'inserimento dei parametri, se sono fattibili li esegue, se non sono fattibili non li accetta e ripropone parametri coerenti.
- Il sistema deve essere in grado di calcolare il consumo di energia totale previsto.
- Il sistema deve essere in grado di "salvaguardarsi", al superamento di una certa soglia di energia, ad esempio i 2300 watt/ora, viene provocata una disattivazione di energia volontaria.
- Il sistema attraverso i sensori deve essere in grado di calcolare le misure della stanza, misurare la temperatura dell'aria e altri parametri.
- Il sistema deve essere in grado di scegliere in automatico quale funzione attivare in base alla data di terminazione inserita dall'utente.

Modelli del sistema

Andiamo a questo punto ad analizzare il modello strutturale del sistema con i vari scenari, casi d'uso e classi da noi implementate.

Prima di fare questo inizieremo con una descrizione esaustiva degli attori interni ed esterni al sistema.

Analisi introduttiva delle figure attoriali da noi riconosciute con le relative responsabilità:

User: colui che si occupa della configurazione iniziale in fase di installazione del software, inserendo i **dati geografici/altimetrici, dati ambientali e dati temporali**.

Si occupa inoltre dell'inserimento delle funzioni specifiche da eseguire con le relative date di terminazione e le soglie di assorbimento di energia elettrica.

SmartController: esso si occupa di valutare il corretto inserimento dei dati da parte dell'utente e nel caso di un eventuale errore, proporre una soluzione quanto più vicina alla sua richiesta (ES. Date inserite non coerenti e/o soglie di assorbimento non supportate). Al fine di migliorare l'esperienza lato utente il software tramite il CONTROLLER si occupa anche dell'ottimizzazione dei consumi indicando per ogni soluzione/funzione l'utilizzo totale di energia previsto.

Scheduler: si occupa della parallelizzazione delle funzioni attive al fine di armonizzarle per ottimizzare i consumi di energia elettrica, rispettando le richieste inserite in input (ES. date terminazione lavoro) gestite dallo SMARTCONTROLLER.

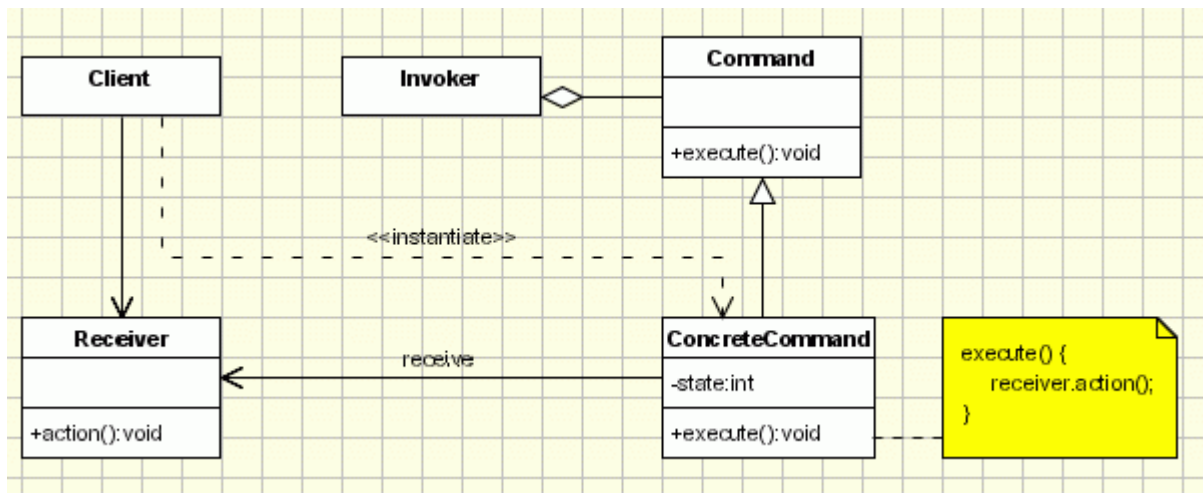
Sensor: quest'ultimo attore sfrutta i dati inseriti in fase di installazione al fine di ricavare informazioni utili come il peso, l'umidità e la tipologia di bucato.

Dopo aver descritto gli attori principali del nostro sistema, analizziamo adesso la struttura interna del nostro software.

Al fine di rispettare i cinque principi fondamentali (SOLID) per una corretta progettazione, abbiamo utilizzato un alto grado di astrazione per evitare un "contatto" diretto tra l'utente e le operazioni effettivamente eseguite. Per fare questo ci siamo serviti di alcuni modelli di risoluzione problemi come i **DESIGN PATTERN**.

Il software è progettato in modo tale da poter consentire all'utente di inserire varie tipologie di comandi.

Per incapsulare e parametrizzare le suddette richieste utilizziamo il **PATTERN COMMAND**.



Nel nostro caso abbiamo che:

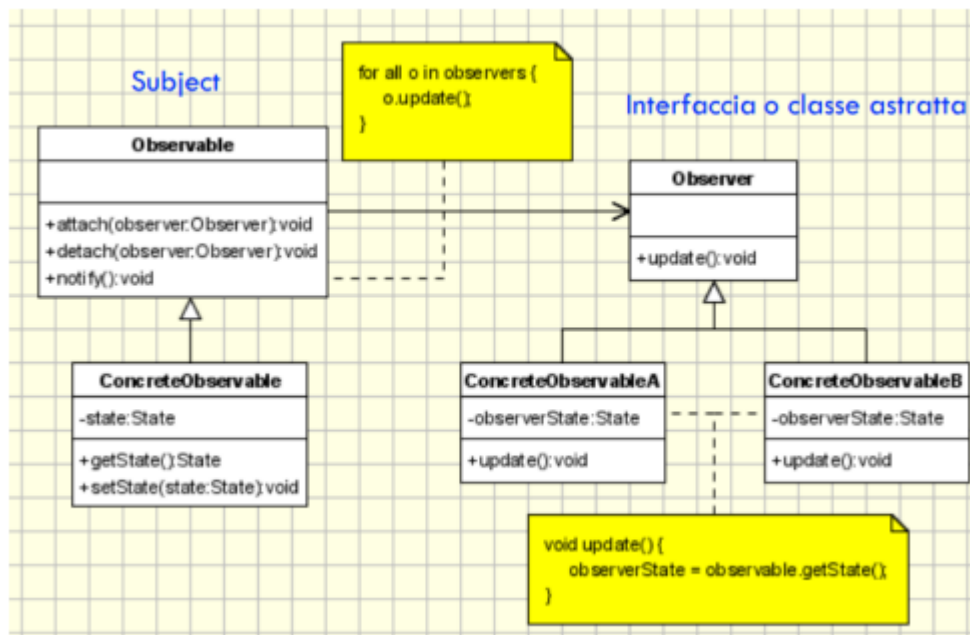
Invoker: Rappresentato dallo **scheduler** che si occupa della gestione singola e contemporanea di più comandi.

Command: Rappresentato dalla classe astratta **programming**.

ConcreteCommand: Rappresentati dalle classi che estendono **programming** ovvero **DryCommand** e **WashCommand**

Receiver: Rappresentato dalla classe **SmartController** che riceve il comando e lo smista a seconda del tipo ai dispositivi associati.

Il software inoltre deve poter costantemente controllare ed eventualmente notificare tramite una classe apposita le variazioni di temperatura dell'ambiente. Per fare questo utilizziamo il **PATTERN OBSERVER**



Nel nostro caso abbiamo che:

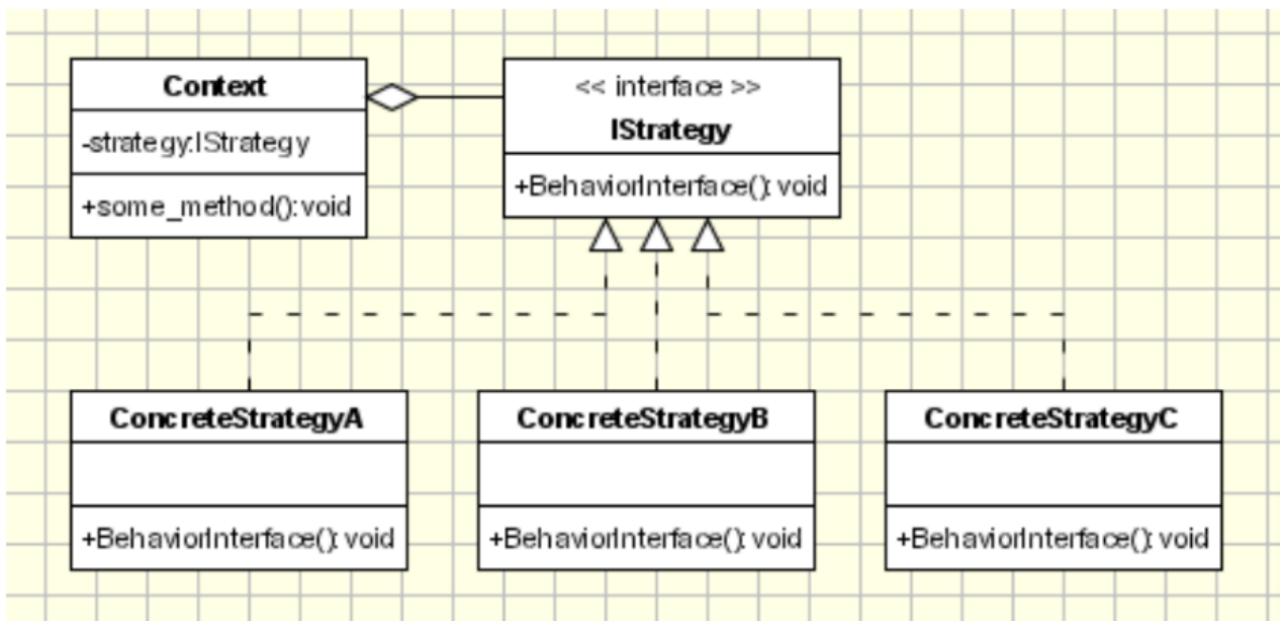
Observable: Rappresentato dalla classe astratta **Sensor**

ConcreteObservable: Rappresentati da **WetherSensor** e **InternalSensor** che si occupano di notificare i cambi di temperatura, per poi essere modificati nella classe **Environment** tenendo conto anche di una media per il salvataggio di eventuali abitudini.

Observer: Rappresentato dalla classe astratta omonima nel nostro diagramma.

ConcreteObserver: Rappresentato da **TemperatureObserver** che utilizza il metodo update per aggiornare la temperatura inviata poi alla classe Environment come precedentemente spiegato.

Utilizziamo inoltre, il **pattern Strategy** che ci permette di calcolare il peso e il consumo energetico a seconda della tipologia di bucato attuando una strategia dedicata.



Nel nostro caso abbiamo che:

IStrategy: Rappresentato dalla classe astratta **InternalSensor**

ConcreteStrategy: Rappresentato dalle classi **CottonEnergyCalculationStrategy**, **DelicateEnergyCalculationStrategy** e **SyntheticsEnergyCalculationStrategy** per le strategie riguardanti il calcolo dell'energia, mentre **CottonWeightCalculationStrategy**, **SyntheticsWeightCalculationStrategy** e **DelicateWeightCalculationStrategy** per il calcolo del peso.

Context: Rappresentato dalla classe **LaundryAppliance** ovvero il nostro elettrodomestico che "contiene" i vari dispositivi.

All'interno della classe **LaundryAppliance** avremo un attributo di tipo **InternalSensor** per comunicare la tipologia di strategia adottata.

Scenari

Scenario “impostaParametriLavaggio”

ISTANZA ATTORE PARTECIPANTE: Simone: User

FLUSSO DI EVENTI:

1. Simone accende l'elettrodomestico
2. Simone accede all'interfaccia grafica
3. Simone seleziona tramite l'interfaccia grafica la tipologia di programma
4. Simone seleziona la data di terminazione richiesta
5. Simone ha inserito i parametri di lavaggio

Scenario “controllaStatoDispositivo”

ISTANZA ATTORE PARTECIPANTE: Mattia: User

FLUSSO DI EVENTI:

1. Mattia accede all'interfaccia grafica
2. Mattia controlla lo stato di esecuzione del dispositivo tramite l'interfaccia grafica
3. Mattia sceglie se sospendere l'esecuzione o riprendere l'esecuzione in base allo stato del dispositivo.

Scenario “esecuzioneTecnicheOttimizzazione”

ISTANZE ATTORI PARTECIPANTI: Luca: User, Smart Controller, Scheduler, Sensor

FLUSSO DI EVENTI:

1. Luca accede all'interfaccia grafica
2. Luca seleziona la tipologia di programma e gli altri parametri richiesti tramite interfaccia grafica
3. Smart Controller riceve i parametri inseriti da Luca
4. Smart Controller richiama lo scheduler
5. Scheduler gestisce i comandi, armonizza i consumi e riduce le tempistiche
6. Sensor calcola i consumi generali
7. Luca visualizza tramite interfaccia grafica i consumi generali

Scenario “richiedeAnnullamentoProgramma”

ISTANZE ATTORI PARTECIPANTI: Luisa: User, Smart Controller

FLUSSO DI EVENTI:

1. Luisa accede all'interfaccia grafica
2. Luisa seleziona il programma da annullare
3. Smart Controller riceve la richiesta di annullamento del programma da parte di Luisa
4. Smart Controller annulla il programma

Scenario “controllaCondizioniMeteo”

ISTANZE ATTORI PARTECIPANTI: Angela: User, Sensor

FLUSSO DI EVENTI:

1. Angela utilizza l'interfaccia grafica
2. Angela attraverso l'interfaccia grafica vuole controllare la temperatura dell'ambiente
3. Il sensore riceve la richiesta di controllare le condizioni meteo da parte di Angela
4. Il sensore verifica le condizioni
5. Il sensore invia la risposta ad Angela
6. Angela visualizza la risposta sull'interfaccia grafica

Scenario “attivaCiclo”

ISTANZE ATTORI PARTECIPANTI: Marta: User, Smart Controller

FLUSSO DI EVENTI:

1. Marta accede all'interfaccia grafica
2. Marta attraverso l'interfaccia grafica seleziona la tipologia del programma e gli altri parametri
3. Smart Controller riceve i parametri inseriti da Marta
4. Smart Controller controlla la validità dei dati inseriti
5. Smart Controller attiva il ciclo se ci sono dati coerenti

Scenario “calcoloPesoBucato”

ISTANZE ATTORI PARTECIPANTI: Tony: User, Sensor

FLUSSO DI EVENTI:

1. Tony mette il bucato
2. Tony seleziona la tipologia di bucato
3. Tony seleziona il tipo di tessuto
4. Sensor riceve i dati inseriti da Tony
5. Sensor controlla se il bucato è bagnato o meno
6. Sensor pesa il bucato
7. Sensor restituisce peso del peso
8. Tony tramite l'interfaccia grafica visualizza il risultato

Modello casi d'uso

Tabelle casi d'uso

NOME CASO D'USO:	SetWashingParameters
Attori partecipanti:	User
Condizioni d'ingresso:	User imposta i parametri di lavaggio come la tipologia di materiale e la data richiesta di terminazione
Flusso di eventi:	<div><div>1. User utilizza l'interfaccia grafica per impostare i parametri.</div><div>2. User imposta manualmente i parametri di lavaggio.</div></div>
Condizioni di uscita:	Utente ha impostato i parametri di lavaggio

NOME CASO D'USO:	SetDryingParameters
Attori partecipanti:	User
Condizioni d'ingresso:	User imposta i parametri di asciugatura come la tipologia di materiale e la data richiesta di terminazione.
Flusso di eventi:	<ol style="list-style-type: none">1. User utilizza l'interfaccia per impostare i parametri2. User imposta manualmente i parametri di asciugatura.
Condizioni di uscita:	User ha impostato i parametri di asciugatura

NOME CASO D'USO:	SetHeatingParameters
Attori partecipanti:	User
Condizioni d'ingresso:	User imposta i parametri di riscaldamento durante la configurazione come i dati ambientali, geografici e temporali
Flusso di eventi:	<ol style="list-style-type: none">1. User utilizza l'interfaccia per impostare i parametri2. User imposta manualmente i parametri di riscaldamento.
Condizioni di uscita:	User ha impostato i parametri di riscaldamento

NOME CASO D'USO:	SetDehumidificationParameters
Attori partecipanti:	User
Condizioni d'ingresso:	User imposta i parametri di deumidificazione durante la configurazione come i dati ambientali, geografici e temporali
Flusso di eventi:	<ol style="list-style-type: none"> 1. User utilizza l'interfaccia per impostare i parametri 2. User imposta manualmente i parametri di deumidificazione
Condizioni di uscita:	User ha impostato i parametri di deumidificazione

NOME CASO D'USO:	CheckDeviceStatus
Attori partecipanti:	User
Condizioni d'ingresso:	User vuole controllare lo stato del dispositivo per eventualmente aggiornarlo
Flusso di eventi:	<ol style="list-style-type: none"> 1. User utilizza l'interfaccia per controllare lo stato di esecuzione del dispositivo. 2. User utilizza il tasto sospendi esecuzione per sospendere un'esecuzione del dispositivo. 3. User utilizza il tasto riprendi esecuzione per riprendere l'esecuzione del dispositivo.
Condizioni di uscita:	stato del dispositivo

NOME CASO D'USO:	CheckWeatherConditions
Attori partecipanti:	User, Sensor
Condizioni d'ingresso:	Bisogna avere un ambiente configurato in modo corretto e lo stato del dispositivo con l'apposito sensore in stato ON
Flusso di eventi:	<p>1) L'utente attraverso l'interfaccia grafica ha la possibilità di controllare tramite una richiesta l'umidità oppure la temperatura dell'ambiente.</p> <p>2) Il sensore ritorna il dato effettivo utile all'utente</p>
Condizioni di uscita:	L'utente non invia più la richiesta.

NOME CASO D'USO:	MonitorEnergyConsumption
Attori partecipanti:	User, Sensor
Condizioni d'ingresso:	User vuole monitorare il consumo energetico attraverso il sensore
Flusso di eventi:	<ol style="list-style-type: none"> 1. User utilizza l'interfaccia per monitorare il consumo energetico 2. Il sensore in automatico monitora il consumo energetico 3. Il sensore comunica il dispendio di energia 4. User visualizza il consumo energetico
Condizioni di uscita:	Il sensore monitora il consumo energetico

NOME CASO D'USO:	RequestCancellationProgram
Attori partecipanti:	User
Condizioni d'ingresso:	User vuole richiedere un annullamento del programma di lavaggio, asciugatura o altro.
Flusso di eventi:	<ol style="list-style-type: none">1. User utilizza l'interfaccia2. User annulla il programma.
Condizioni di uscita:	User annulla il programma.

NOME CASO D'USO:	ControlConsistencyParameters
Attori partecipanti:	SmartController
Condizioni d'ingresso:	Dopo che l'utente imposta i parametri per avviare il programma con le relative date di terminazione desiderate, lo SmartController si occupa di controllare la coerenza dei dati inseriti per evitare situazioni di errore
Flusso di eventi:	<p>1) L'utente imposta i parametri di input (tipologia del programma e data di terminazione).</p> <p>2) Lo SmartController controlla la validità dei dati inseriti e nel caso di un inserimento non coerente, tramite una sua estensione "propostaParametriCoerenti" invia ad esempio una data coerente vicina a quella inserita dall'utente</p>
Condizioni di uscita:	L'esito del controllo è positivo

NOME CASO D'USO:	CalculationWeightLaundry
Attori partecipanti:	Sensor
Condizioni d'ingresso:	Il sensore calcola il peso del bucato in base al tipo di materiale
Flusso di eventi:	<ol style="list-style-type: none"> 1. Il sensore seleziona la tipologia di bucato 2. Il sensore seleziona il tipo di tessuto 3. Il sensore pesa il carico.
Condizioni di uscita:	Il sensore ha calcolato il peso del bucato

NOME CASO D'USO:	CancelProgram
Attori partecipanti:	SmartController
Condizioni d'ingresso:	Lo SmartController deve annullare il programma
Flusso di eventi:	<ol style="list-style-type: none"> 1. L'utente (User) richiede l'annullamento del programma attraverso il caso d'uso "richiedeAnnullamentoProgramma" 2. Lo SmartController riceve la richiesta dell'utente (User) 3. Lo SmartController annulla il programma scelto dall'utente.
Condizioni di uscita:	Lo SmartController ha annullato il programma

NOME CASO D'USO:	IndicateExpectedConsumption
Attori partecipanti:	Sensor, SmartController
Condizioni d'ingresso:	Calcolo automatico dei consumi previsti
Flusso di eventi:	<ol style="list-style-type: none"> 1. Il sensore comunica i dati ambientali e ad altri allo smartController 2. Lo smartController in base ai dati ricevuti indica i possibili consumi previsti 3. Vengono comunicati all'utente (User) i consumi previsti attraverso l'interfaccia.
Condizioni di uscita:	Sono stati calcolati i consumi previsti

NOME CASO D'USO:	ExecutionTechniquesOptimization
Attori partecipanti:	Scheduler, SmartController, Sensor
Condizioni d'ingresso:	Per ogni comando inserito il software utilizza tecniche di ottimizzazione per la gestione parallela
Flusso di eventi:	<ol style="list-style-type: none"> 1. I parametri inseriti dall'utente vengono gestiti dallo smartController. 2. Una volta ricevuti i parametri, lo smartController utilizza lo scheduler per gestire i comandi al fine di armonizzare i consumi e ridurre le tempistiche. 3. Il sensore calcola i consumi generali di ogni funzione.
Condizioni di uscita:	Superamento soglia di wattora.

NOME CASO D'USO:	ParallelManagementCycles
Attori partecipanti:	Scheduler
Condizioni d'ingresso:	L'utente inserisce più operazione da effettuare nello stesso tempo. Lo scheduler deve poter gestire contemporaneamente più funzioni.
Flusso di eventi:	<ol style="list-style-type: none"> 1. L'utente richiede più funzioni in contemporanea tramite l'interfaccia 2. Lo scheduler si occupa della gestione di più funzioni in contemporanea armonizzando i consumi e i tempi
Condizioni di uscita:	Annullamento da parte dell'utente oppure soglia di consumi superata

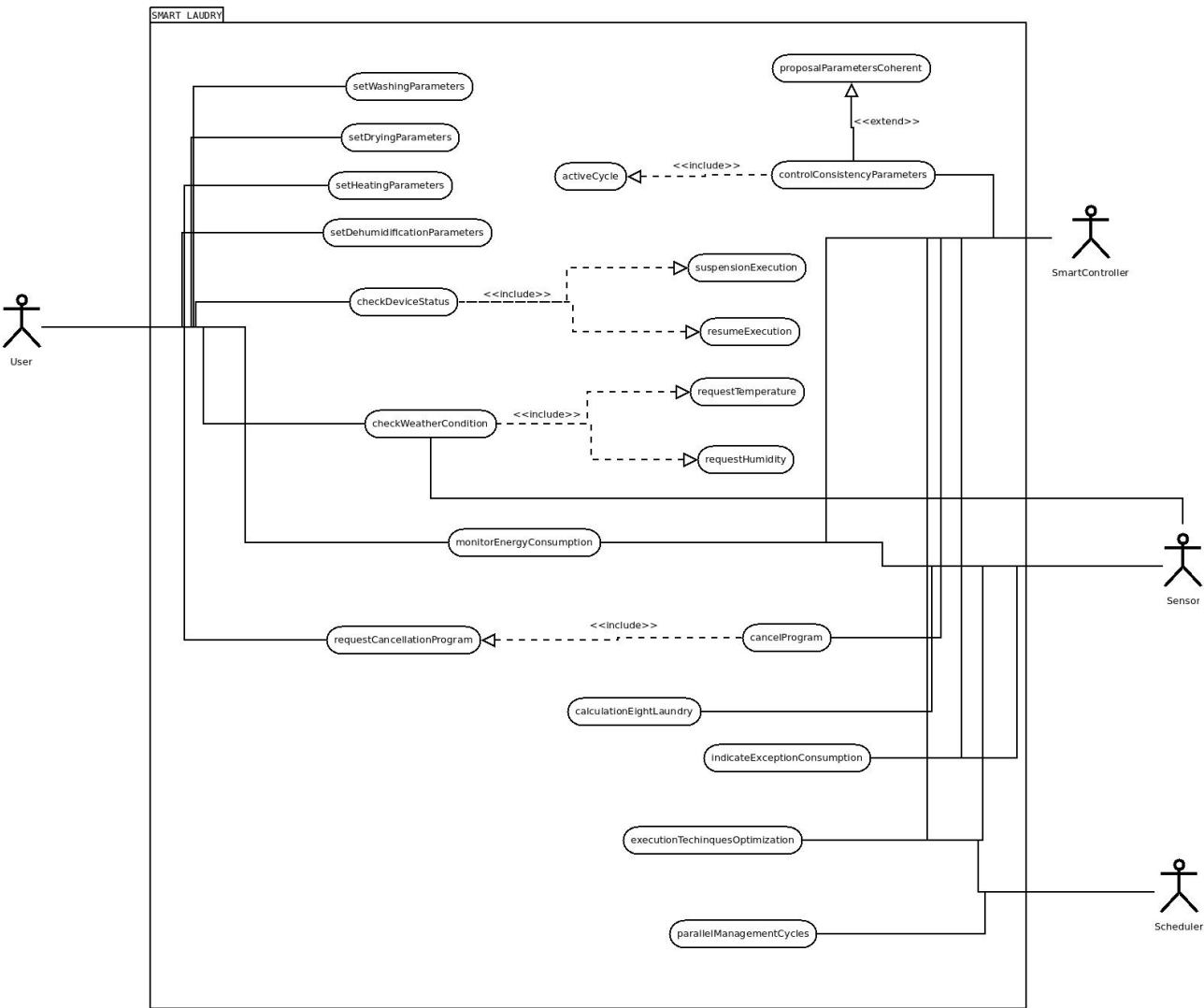
NOME CASO D'USO:	SuspensionExecution
Attori partecipanti:	User
Condizioni d'ingresso:	Lo user controlla lo stato del dispositivo e può accedere alla funzione di sospendiEsecuzione o riprendiEsecuzione.
Flusso di eventi:	<ol style="list-style-type: none"> 1. User utilizza l'interfaccia. 2. User controlla lo stato dell'esecuzione. 3. User sospende l'esecuzione della funzione se è in stato di elaborazione.
Condizioni di uscita:	

NOME CASO D'USO:	ResumeExecution
Attori partecipanti:	User
Condizioni d'ingresso:	Lo user controlla lo stato del dispositivo e può accedere alla funzione di riprendiEsecuzione
Flusso di eventi:	<ol style="list-style-type: none"> 1. User utilizza l'interfaccia. 2. User controlla lo stato dell'esecuzione. 3. User riprende l'esecuzione della funzione se lo stato è sospeso.
Condizioni di uscita:	

NOME CASO D'USO:	ActiveCycle
Attori partecipanti:	SmartController
Condizioni d'ingresso:	Entriamo in questa condizione quando il controller che gestisce la coerenza dei parametri inseriti dall'utente non trova nessun errore.
Flusso di eventi:	<ol style="list-style-type: none"> 1) L'utente imposta i parametri di input. 2) Lo SmartController controlla la validità dei dati inseriti. 3) Nel caso di un inserimento coerente, viene attivato il ciclo
Condizioni di uscita:	Dati inseriti dall'utente non idonei.

NOME CASO D'USO:	ProposalParametersCoherent
Attori partecipanti:	SmartController
Condizioni d'ingresso:	Entriamo in questa condizione solo quando il caso d'uso che gestisce la coerenza dei parametri inseriti dall'utente trova un errore.
Flusso di eventi:	<ol style="list-style-type: none"> 1) L'utente imposta i parametri di input (tipologia del programma e data di terminazione). 2) Lo SmartController controlla la validità dei dati inseriti. 3) Nel caso di un inserimento non coerente, viene inviata una proposta di dati coerenti vicina alla richiesta dell'utente.
Condizioni di uscita:	Dati ricevuti dall'utente

Schema casi d'uso



Modello ad oggetti

Dizionario dei dati

ApplianceController: Interfaccia che verrà implementata dallo SmartController e da LaundryAppliance successivamente.

LaundryAppliance: rappresenta l'elettrodomestico che comprende le quattro funzioni di lavaggio, asciugatura, deumidificazione e riscaldamento. Contiene i sensori necessari per rilevare la temperatura e l'umidità dell'aria e il peso del bucato, e comunica con il software per attivare le funzioni richieste dall'utente. La classe implementa l'interfaccia ApplianceController.

WashingProgram: rappresenta un programma di lavaggio, che definisce i parametri di lavaggio per una specifica tipologia di bucato (ad esempio, cotone, sintetico, delicato). Contiene informazioni sul tempo di lavaggio, la temperatura dell'acqua, la velocità di centrifuga, ecc. Viene utilizzata dalla classe LaundryAppliance per eseguire il lavaggio del bucato.

DryingProgram: rappresenta un programma di asciugatura, che definisce i parametri di asciugatura per una specifica tipologia di bucato (ad esempio, cotone, sintetico, delicato). Contiene informazioni sulla temperatura dell'aria, la velocità di rotazione del tamburo, ecc. Viene utilizzata dalla classe LaundryAppliance per eseguire l'asciugatura del bucato.

Humidifier: rappresenta la classe che gestisce la deumidificazione dell'aria. Contiene informazioni sulla potenza dell'apparecchio e sulle modalità di funzionamento. Viene utilizzata dalla classe LaundryAppliance per deumidificare l'aria nell'ambiente circostante.

Heater: rappresenta la classe che gestisce il riscaldamento dell'aria. Contiene informazioni sulla potenza dell'apparecchio e sulle modalità di funzionamento. Viene utilizzata dalla classe LaundryAppliance per riscaldare l'aria nell'ambiente circostante.

SmartController: rappresenta il software di controllo dell'elettrodomestico. Può attivare le funzioni di lavaggio, asciugatura, deumidificazione e riscaldamento in modo manuale o automatico. Implementa l'interfaccia ApplianceController.

EnergyMonitor: rappresenta un monitor di energia elettrica, che rileva il consumo di energia dell'elettrodomestico e segnala se viene superata una soglia predefinita.

Viene utilizzata dalla classe SmartController per controllare il consumo di energia durante il funzionamento dell'elettrodomestico.

WeatherMonitor: rappresenta un monitor delle condizioni meteorologiche, che riceve i dati di temperatura e umidità dal sensore esterno WeatherSensor e li utilizza per fare alcune valutazioni.

WeatherSensor: classe che rappresenta il sensore esterno.

LaundryCycle: classe astratta che rappresenta un ciclo di lavaggio o di asciugatura.

WashCycle: classe che estende LaundryCycle e implementa il ciclo di lavaggio.

DryCycle: classe che estende LaundryCycle e implementa il ciclo di asciugatura.

Environment: classe che rappresenta l'ambiente in cui si trova l'elettrodomestico.

LaundrySmart: classe principale che gestisce il software dell'elettrodomestico.

UserInterface: classe che rappresenta l'interfaccia utente.

Scheduler: classe che consente la gestione di più funzioni in contemporanea.

SchedulerCommand: esegue più funzioni in contemporanea.

Observer: Interfaccia che implementa il pattern observer che consente di aggiornare la temperatura dell'acqua o la velocità della centrifuga in base alla temperatura dell'aria.

Programmation: Interfaccia che viene implementata dalle classi che servono ad eseguire i comandi per le funzioni, in contemporanea o meno, di lavaggio o asciugatura come DryCommand e/o WashCommand.

Sensor: Interfaccia che viene implementata da InternalSensor.

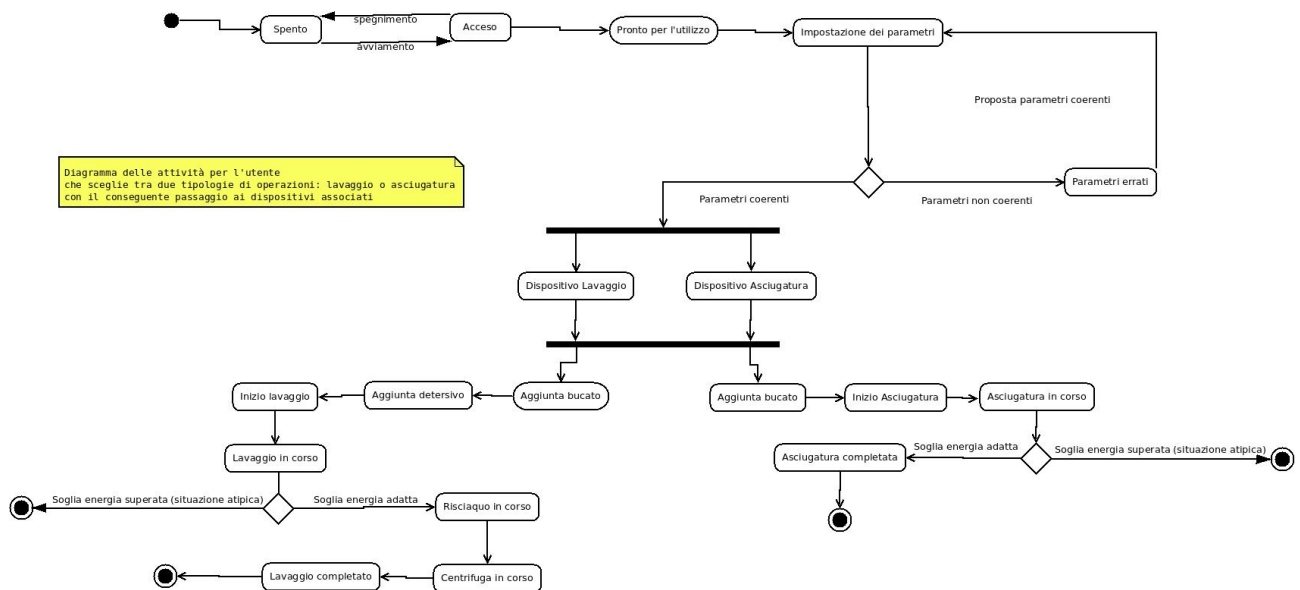
InternalSensor: Classe che implementa Sensor per il pattern Strategy, questa classe può calcolare grazie a dei metodi appositi il consumo di energia, il peso del bucato, indicare il consumo di energia preventiva e altro.

[illegible]

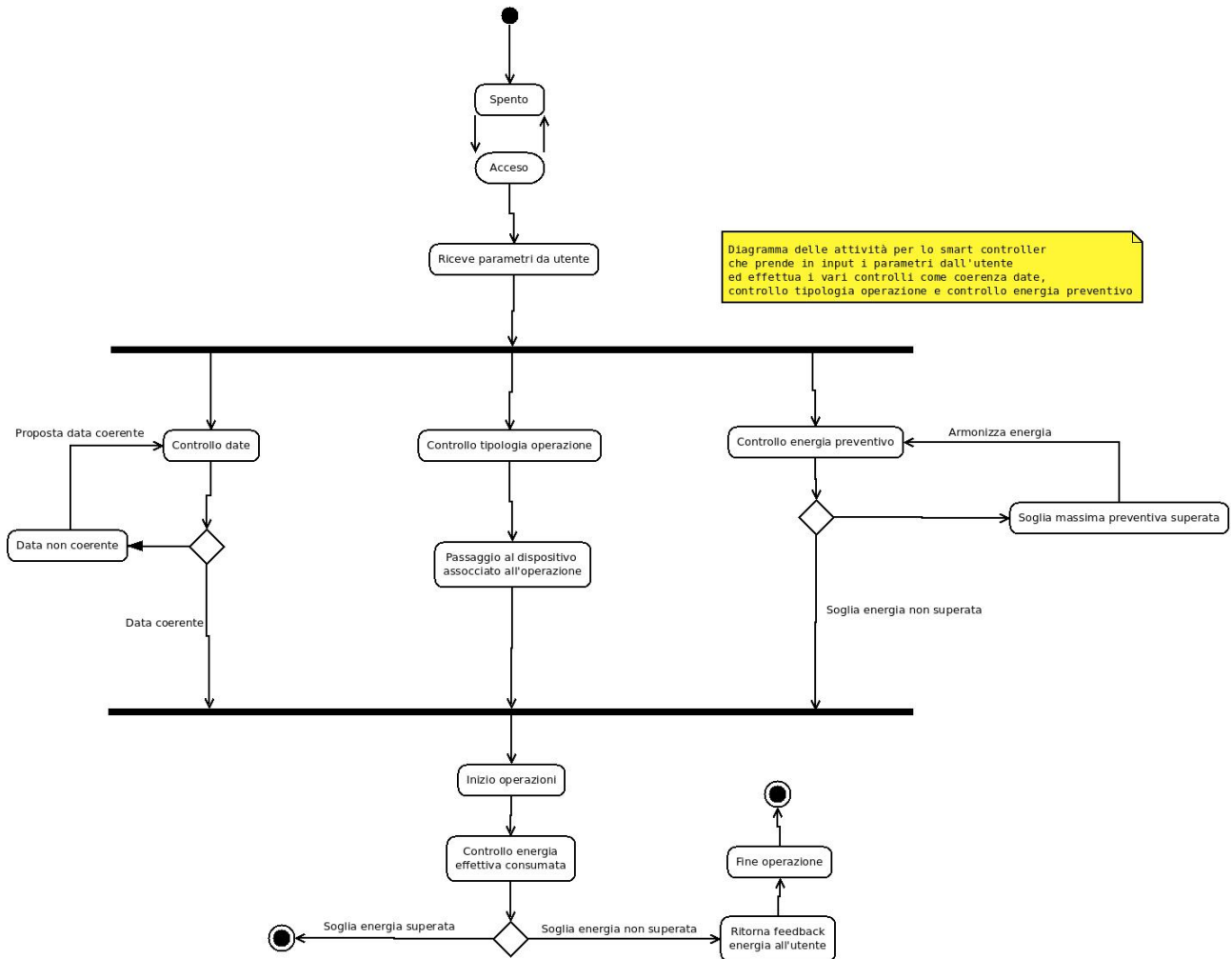
Modelli dinamici

Diagrammi delle attività

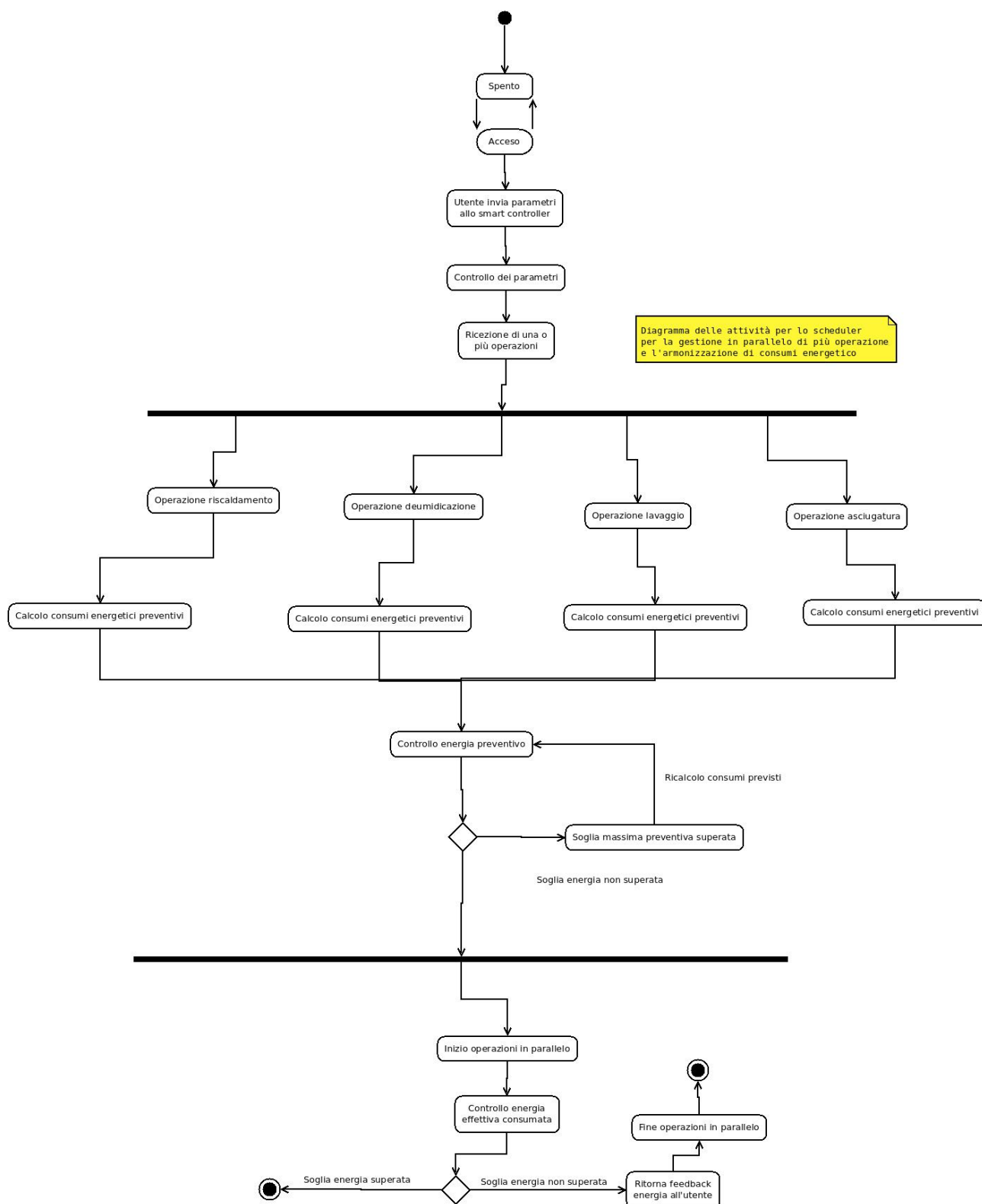
1. Diagramma delle attività per l'utente che sceglie due tipologie di operazioni: lavaggio o asciugatura con il conseguente passaggio ai dispositivi associati.



2. Diagramma delle attività per lo smart controller che prende in input i parametri dall'utente ed effettua i vari controlli come coerenza date, controllo tipologia operazione e controllo energia preventivo.

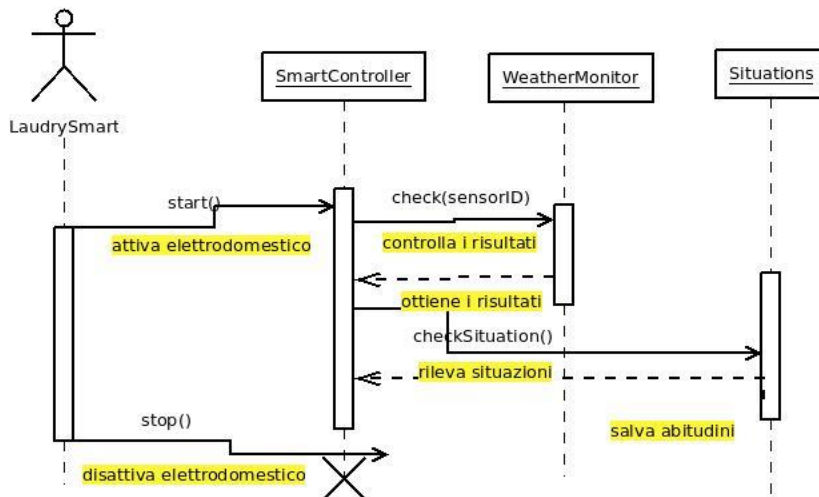


3. Diagramma delle attività per lo scheduler per la gestione in parallelo di più operazioni e l'armonizzazione del consumo energetico.

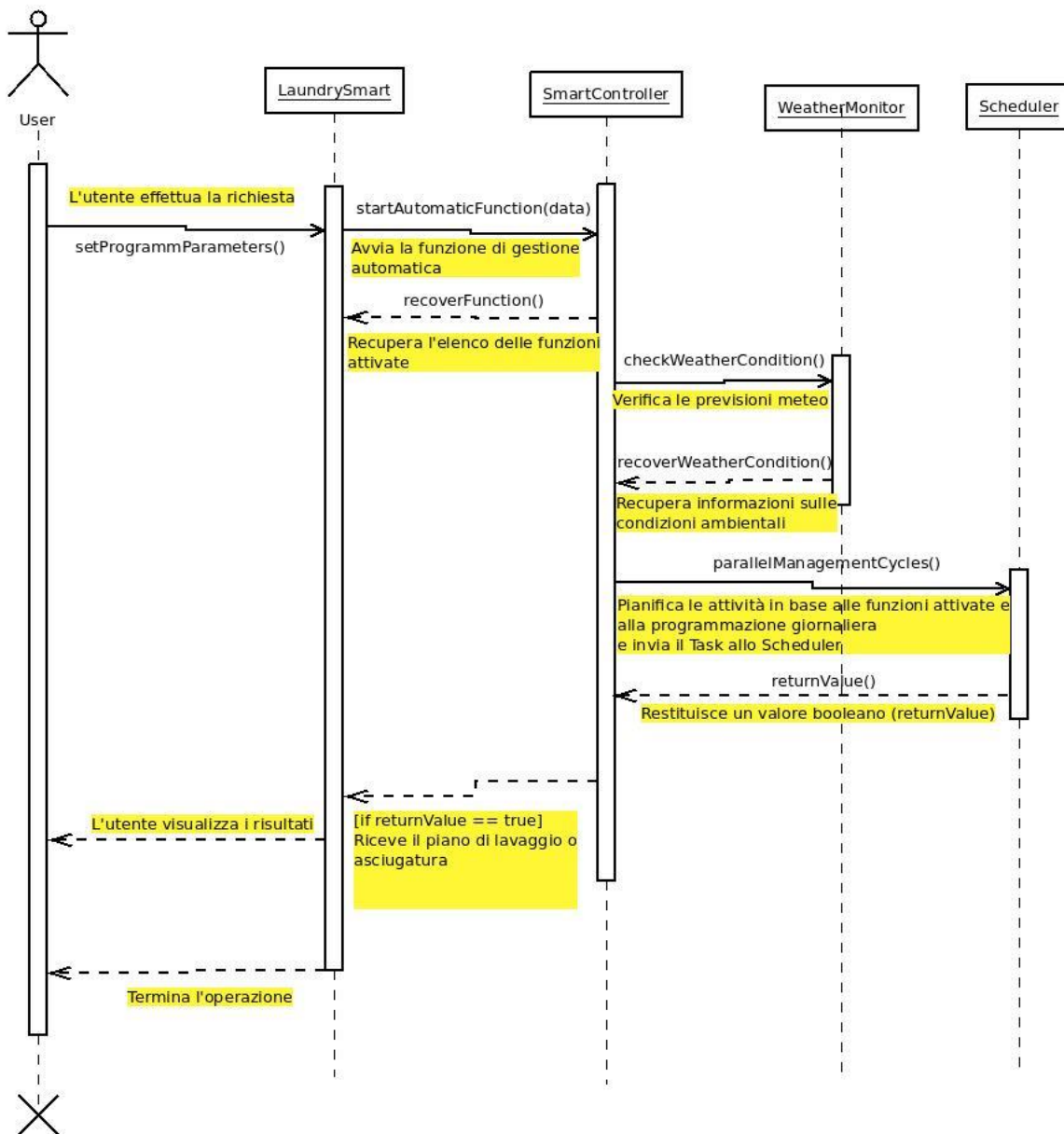


Diagrammi delle sequenze

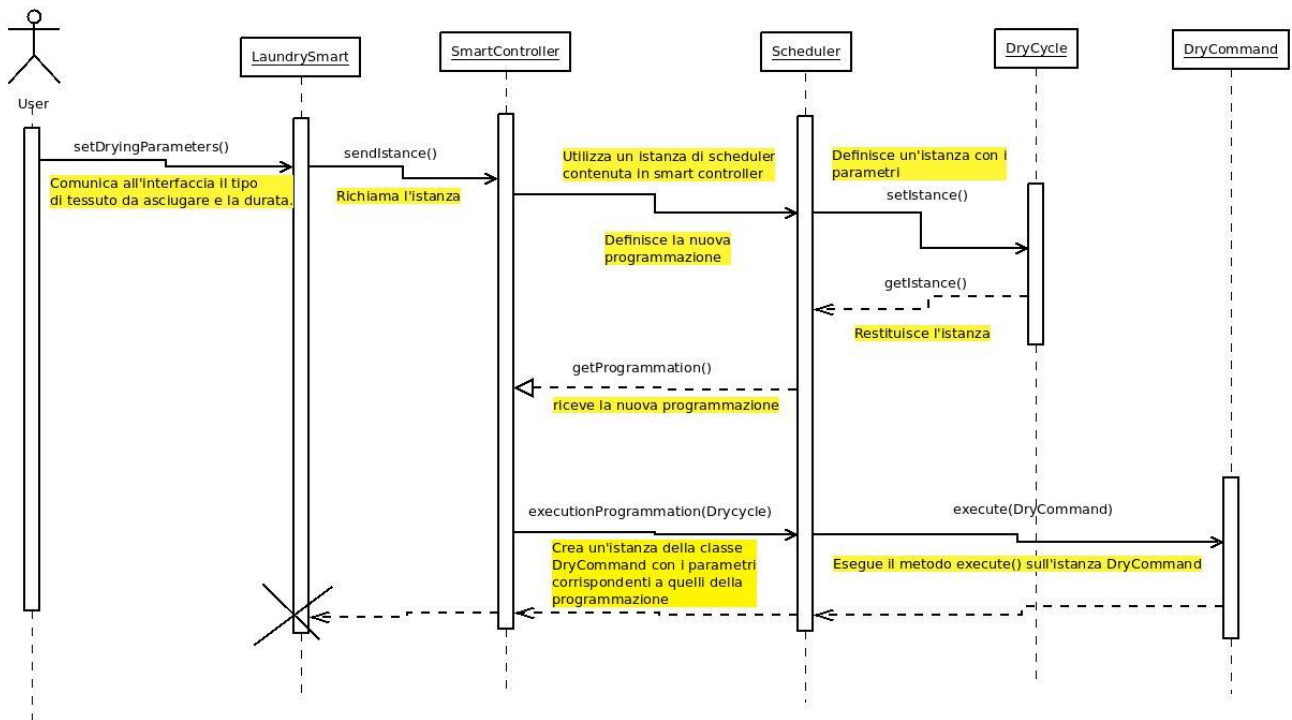
Lo SmartController si avvale dell'oggetto di tipo Situations per verificare le proprie abitudini (ad esempio, farsi la doccia tutte le sere alle ore 20:00)



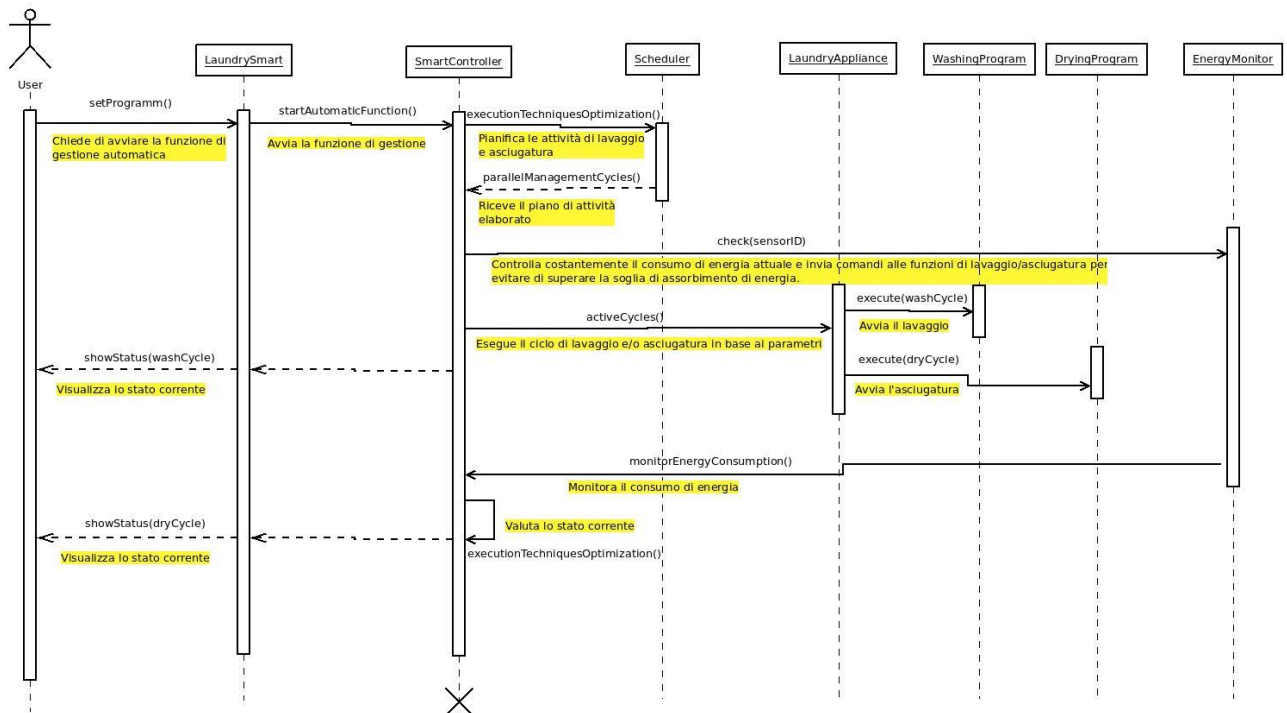
L'utente utilizza lo Scheduler per effettuare la richiesta per la gestione automatica dei programmi/a



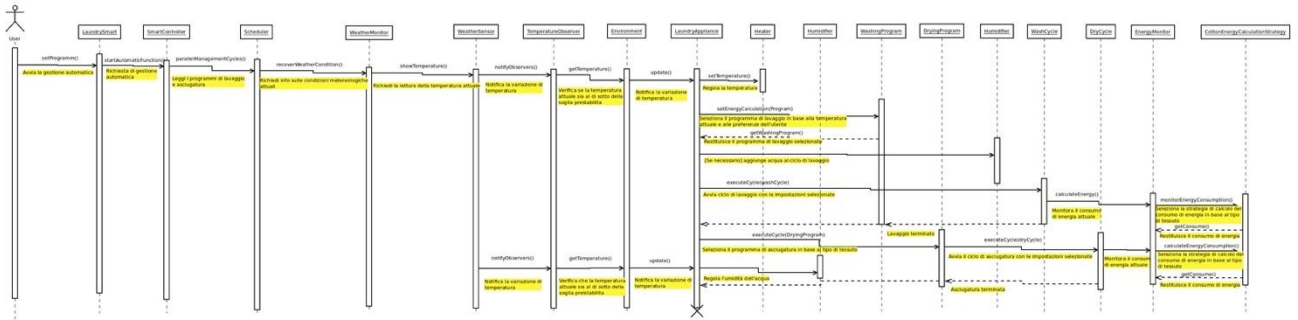
L'utente programma un'asciugatura grazie all'utilizzo dello Scheduler. Seleziona come "cotone" come tipo di tessuto e imposta la durata dell'asciugatura a 60 minuti.
N.B. Consultare lo schema delle classi per le funzioni e le aggregazioni.



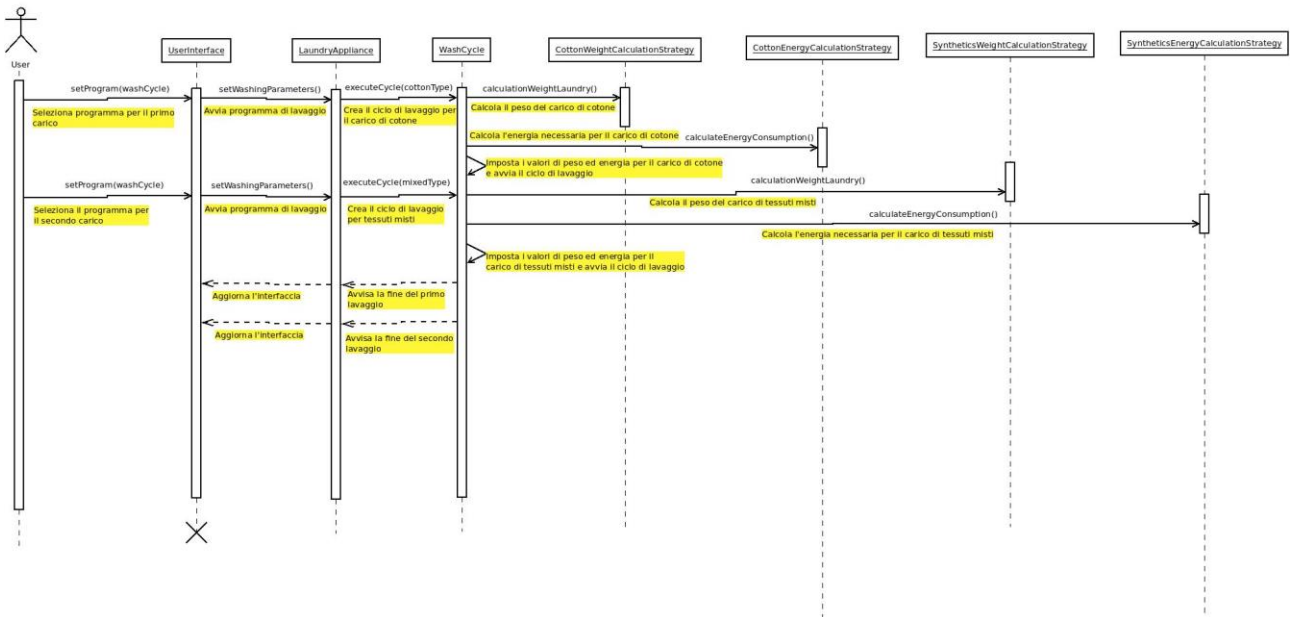
L'utente invia una richiesta e SmartLaundry gestisce le attività di lavaggio e asciugatura contemporaneamente.



L'utente invia una richiesta, le condizioni ambientali cambiano e l'elettrodomestico regola i parametri per adattarsi alle nuove condizioni ambientali.



Ho appena lavato un carico di 5 chilogrammi di cotone, centrifugati a 1.000 giri, e lo metto ad asciugare, contemporaneamente metto a lavare un secondo carico di 4 chilogrammi di tessuti misti, con centrifuga a 800 giri.



Interfaccia utente

Obiettivi del test

Prima di essere consegnata all'utente finale oppure essere lanciata sul mercato, la piattaforma necessita di essere testata. Se il test produce un esito positivo è possibile considerare il prototipo attendibile altrimenti bisognerà modificare le caratteristiche che non rispettano gli obiettivi del test.

Gli obiettivi prefissati per il superamento del test da parte della nostra piattaforma sono i seguenti:

- **Assenza di bug nel sistema:** l'utente deve utilizzare il software in modo fluido senza riscontrare problemi di alcun tipo.
- **Visualizzazione corretta delle diverse dashboard:** l'utente, deve visualizzare la dashboard corretta rispetto alla tipologia di device che vuole utilizzare.
- **Corretto funzionamento dei vari servizi:** bisogna verificare che i servizi offerti dalla piattaforma debbano funzionare correttamente, rispettando i vincoli analizzati nei capitoli precedenti.
- **Sicurezza:** la sicurezza in questo caso riguarda l'affidabilità della gestione delle varie operazioni del software rispettando i limiti dei dispositivi fisici. Tutto ciò per evitare ad esempio casi come il vettore Ariane 5 che esplose durante il volo inaugurale del 4 giugno 1996 dopo 40 secondi dal decollo per una avaria del sistema di controllo. Nel nostro caso quindi lo scheduler deve controllare le molteplici operazioni contemporanee evitando cortocircuiti.

Realizzazione del prototipo

Il prototipo del progetto SmartLaundry ha lo scopo di fornire una simulazione dell'esperienza utente, con una logica di base implementata, posizionandosi nella parte alta dello spazio tridimensionale dei prototipi, vicino al vertice dell'interfaccia. L'utente può interagire con l'applicazione utilizzando i pulsanti e i campi di testo per inserire i dati, sperimentando in parte il ruolo del prodotto nella propria vita.

Il prototipo ha un alto livello di fedeltà, riproducendo fedelmente l'interfaccia utente e l'interazione con il prodotto finale. Sebbene il prototipo non presenti funzionalità complete del sistema, la sua completezza funzionale è orizzontale, permettendo all'utente di valutare l'interfaccia nella sua interezza e di comprendere l'interazione con essa.

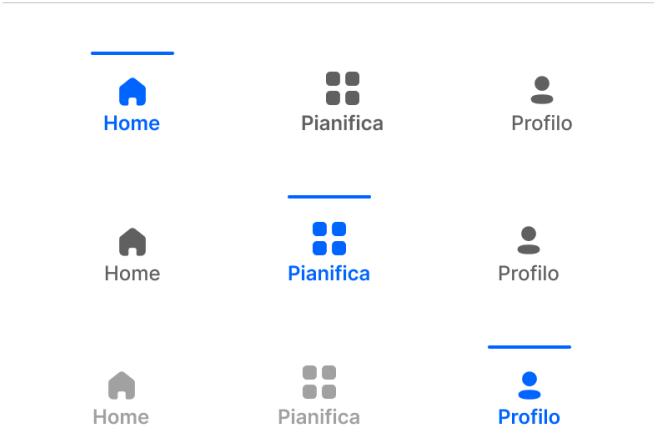
Esso è stato realizzato con lo strumento di progettazione di interfacce Figma.

Il nostro prototipo possiede un'interfaccia generica ma coerente, facile da usare e con un design estetico accattivante. L'interfaccia si presenta con una navigazione dal

basso, che suddivide l'app in sezioni con titoli intuitivi, rendendo facile la comprensione della posizione dell'utente all'interno dell'applicazione. I pulsanti di azione sono collocati nella parte bassa della schermata, per facilitare l'interazione da parte dell'utente.

Inoltre, abbiamo cercato di utilizzare icone supportate da scritte, per rendere l'interazione ancora più istintiva e immediata.

Barra di navigazione



Campi di testo

Campo di testo principale

Bottoni

Bottoni primari



Bottoni secondari



Interruttore / Disabilitato



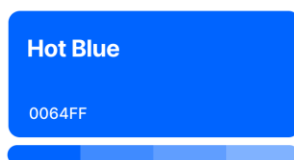
Interruttore / Abilitato



Per quanto riguarda i **colori**, abbiamo scelto i colori blu, rosso, verde, nero, grigio e bianco (colori primari o neutri comunemente utilizzati nell'ambito del design e della grafica).

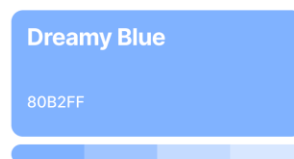
Nel nostro prototipo, utilizziamo tonalità chiare di questi colori per enfatizzare la bellezza del tema chiaro e trasmettere un'idea di pulito. Il bianco, ad esempio, è stato utilizzato come colore di sfondo per la maggior parte delle schermate per creare un'atmosfera di pulizia e leggerezza. Il blu e l'azzurro, invece, sono stati utilizzati per evidenziare i pulsanti di azione e i titoli delle schermate, in quanto questi colori sono associati alla calma, mentre il verde per marcare il successo di un'operazione e per esprimere sicurezza. Il grigio è stato utilizzato per creare contrasto tra il testo e lo sfondo e per enfatizzare le aree di contenuto, mentre il rosso per mettere l'accento su errori.

Colori



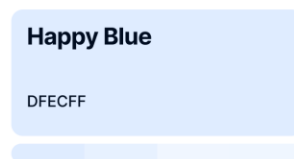
Colore stato attivo

Colore primario



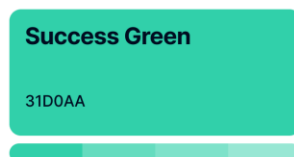
Colore stato attivo

Colore base



Colore d'accento

Colore chiaro



Notifiche di completamento /
operazioni completate

Colore di completamento

<div>Error Red</div> <div>ff0000</div>	Operazioni invalide / errori / problemi	Colore di errore
<div>Dark</div> <div>000E24</div>	Testo del corpo	Colore dei testi
<div>Grey</div> <div>A1A1A1</div>	Campi di testo / sottotitoli	Grigio dei sottotitoli
<div>White</div> <div>#FAFCFE</div>	Colore di sfondo	Colore chiaro

In generale, l'utente si troverà di fronte ad un'interfaccia familiare, già sperimentata in altre applicazioni.

Per quanto riguarda la **tipografia**, abbiamo scelto di utilizzare il font "Inter" nelle sue diverse sfaccettature, poiché è un carattere moderno e leggibile, progettato appositamente per l'uso sul web.

In particolare, abbiamo utilizzato il peso "Bold" per il testo principale dell'applicazione, come i titoli delle schermate, i titoli nelle sezioni e i pulsanti, in modo da renderli maggiormente distinguibili dal resto del testo.

Il peso "Medium" è stato utilizzato per i sottotitoli, in modo da avere un carattere meno pesante rispetto al testo principale, da non risultare troppo invadente.

L'uso coerente del font "Inter" in diverse sfaccettature ci ha aiutato a creare una sensazione di coerenza e uniformità all'interno dell'interfaccia, migliorando così l'esperienza utente complessiva.

Titolo	Benvenuto in SmartLaundry
H2	Benvenuto in SmartLaundry
H3	Benvenuto in SmartLaundry
Sottotitolo	Benvenuto in SmartLaundry
Testo regolare	Benvenuto in SmartLaundry
Testo grassetto	Benvenuto in SmartLaundry
Testo piccolo	Benvenuto in SmartLaundry
Testo dei bottoni	Welcome To My World

Coppie di testo

Titolo & H2

This Is A Random Text

Il nostro progetto di ingegneria del software è carino perché combina innovazione tecnologica e design accattivante. Siamo orgogliosi di presentare il risultato del nostro lavoro e speriamo che soddisfi le vostre aspettative.

Titolo & Testo del corpo

This Is A Random Text

Il nostro progetto di ingegneria del software è carino perché combina innovazione tecnologica e design accattivante. Siamo orgogliosi di presentare il risultato del nostro lavoro e speriamo che soddisfi le vostre aspettative.

H2 & Testo del corpo

This Is A Random Text

Il nostro progetto di ingegneria del software è carino perché combina innovazione tecnologica e design accattivante. Siamo orgogliosi di presentare il risultato del nostro lavoro e speriamo che soddisfi le vostre aspettative.

Testo del corpo & Testo piccolo

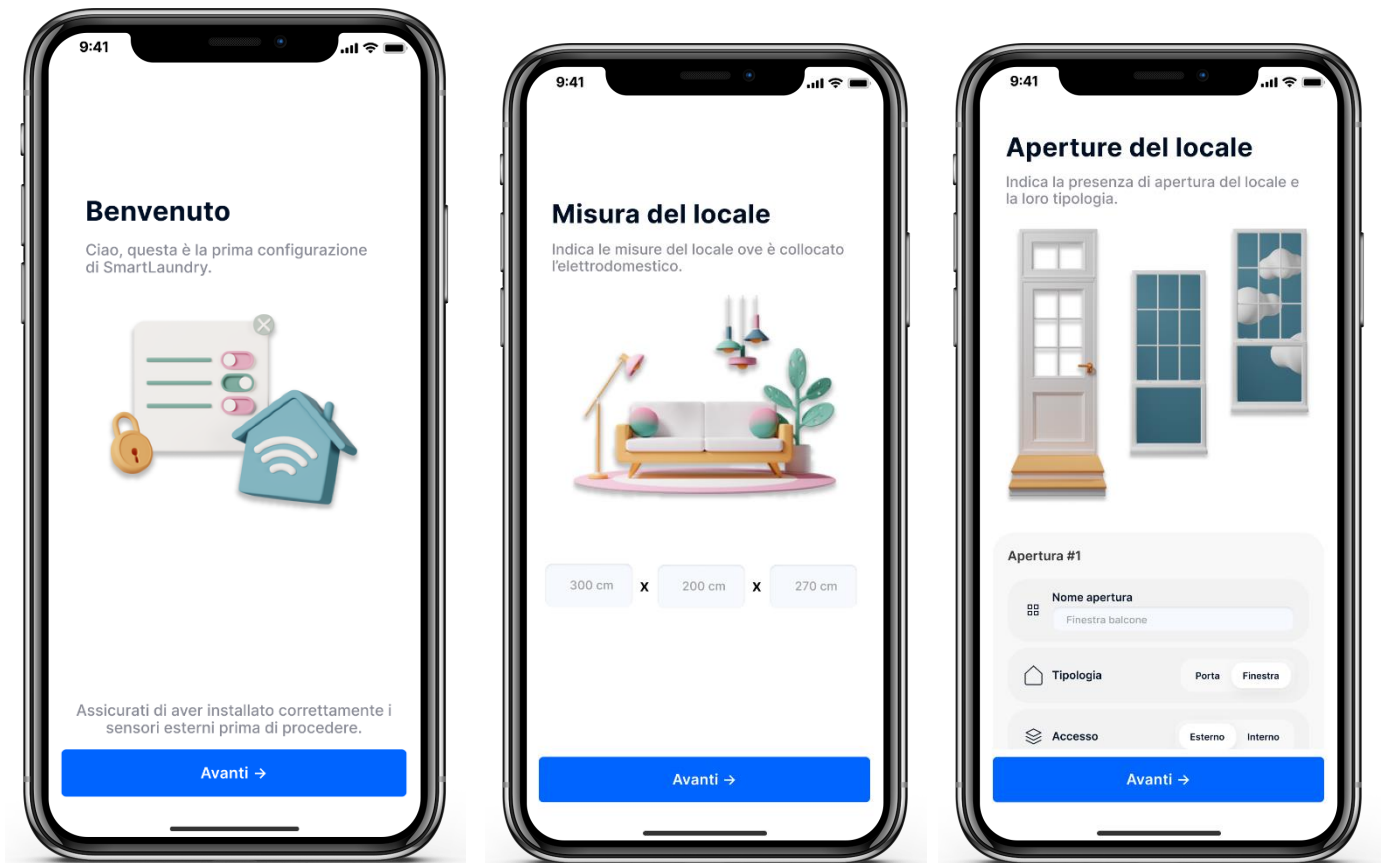
This Is A Random Text

Il nostro progetto di ingegneria del software è carino perché combina innovazione tecnologica e design accattivante. Siamo orgogliosi di presentare il risultato del nostro lavoro e speriamo che soddisfi le vostre aspettative.

Titolo	Font: Inter Weight: Bold Size: 24px Letter Spacing: -2%	Titolo
H2	Family: Inter Weight: Bold Size: 20px Letter Spacing: -2%	Header 2
H3	Family: Inter Weight: Bold Size: 18px Letter Spacing: -2%	Header 3
Sottotitolo	Family: Inter Weight: Medium Size: 18px Letter Spacing: 0%	Sottotitolo
Testo regolare	Family: Inter Weight: Medium Size: 16px Line Height: 140%	Testo regolare
Testo grassetto	Family: Inter Weight: Bold Size: 16px Line Height: 140%	Testo grassetto
Testo piccolo	Family: Inter Weight: Bold Size: 14px	Testo piccolo
Testo dei bottoni	Family: Inter Weight: Bold Size: 10px Letter Spacing: 3%	Testo Dei Bottoni

Il nostro prototipo di SmartLaundry si presenta con una **configurazione iniziale** intuitiva e ben progettata per l'utente. Al primo avvio dell'applicazione, l'utente viene guidato attraverso una serie di semplici passaggi per configurare l'ambiente in cui si trova l'elettrodomestico. In particolare, l'utente viene invitato ad inserire le dimensioni della stanza, il numero di porte e finestre presenti nella stanza e il loro stato.

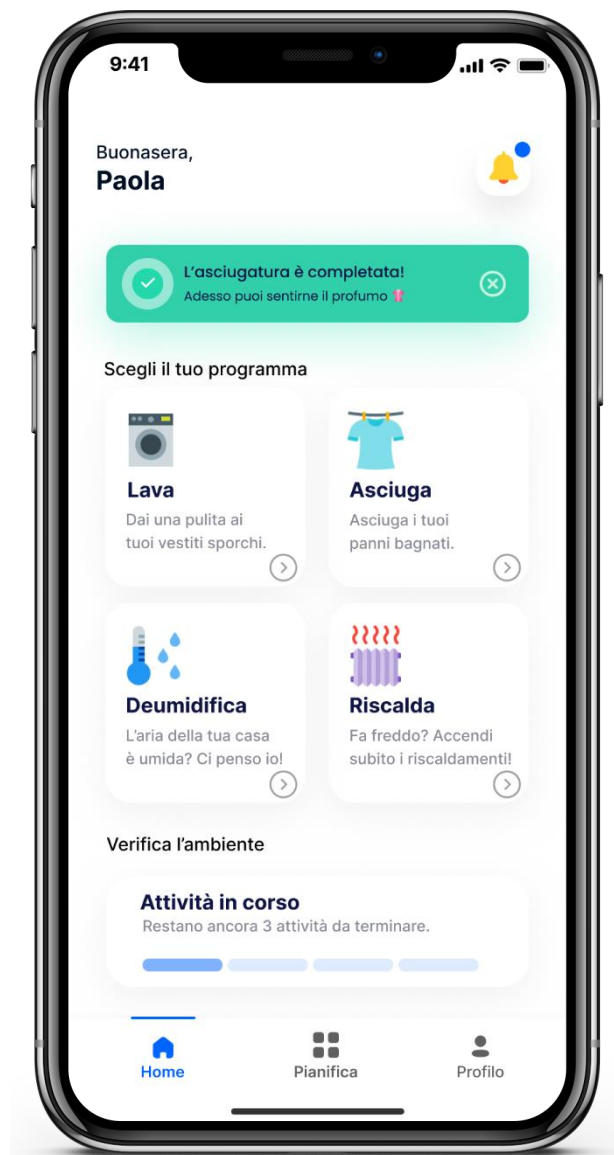
Questa configurazione iniziale è stata progettata con l'obiettivo di rendere l'esperienza dell'utente il più semplice possibile, chiedendo solo le informazioni necessarie per garantire il corretto funzionamento dell'applicazione. Abbiamo prestato molta attenzione all'aspetto estetico della configurazione iniziale, utilizzando colori e font in modo armonioso e coerente con il tema dell'applicazione. In questo modo, siamo riusciti a creare un'esperienza utente accattivante e allo stesso tempo facile da utilizzare, che rende l'interazione con l'applicazione molto più piacevole e intuitiva.



La **schermata principale** del nostro prototipo, la Home, è stata pensata per consentire all'utente di avere un'ampia panoramica sulle informazioni più importanti riguardanti il suo elettrodomestico. In particolare, la schermata visualizza le informazioni in tempo reale provenienti dai sensori della stanza, permettendo all'utente di tenere sotto controllo la temperatura e l'umidità e di intervenire se necessario.

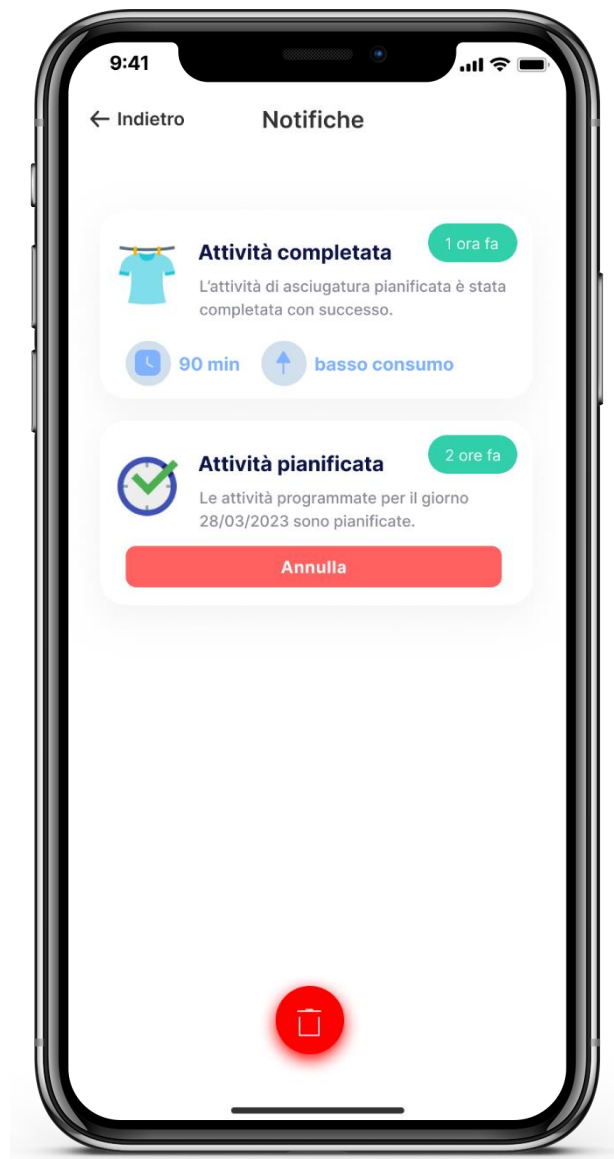
Inoltre, la schermata Home consente di visualizzare le informazioni meteorologiche aggiornate, in modo che l'utente possa adattare il suo programma di lavaggio in base alle condizioni climatiche esterne. Tutte queste informazioni sono presentate in modo chiaro e immediato, consentendo all'utente di avere un'idea precisa della situazione in un solo sguardo.

Oltre a questo, la schermata Home consente all'utente di interagire con le funzioni principali del nostro programma: lavaggio, asciugatura, deumidificazione e riscaldamento. Queste funzioni sono presentate come pulsanti grandi e ben visibili, con icone intuitive che consentono di identificarle immediatamente. Cliccando su uno dei pulsanti, l'utente viene portato alla schermata corrispondente, dove può impostare le opzioni di lavaggio o avviare la funzione desiderata.



Abbiamo inoltre incluso una funzionalità utile che consente agli utenti di tenere traccia delle attività pianificate e delle notifiche relative allo stato del sistema (**notifiche**). Per accedere a questa funzionalità, l'utente può semplicemente fare clic sull'icona della campanella situata nella parte superiore dell'applicazione.

Una volta che l'utente accede alla sezione delle notifiche, può visualizzare tutti gli avvisi relativi alla pianificazione di attività e al completamento delle operazioni. Questa sezione fornisce all'utente informazioni dettagliate su eventuali attività che devono essere svolte, come il lavaggio di un carico di biancheria.

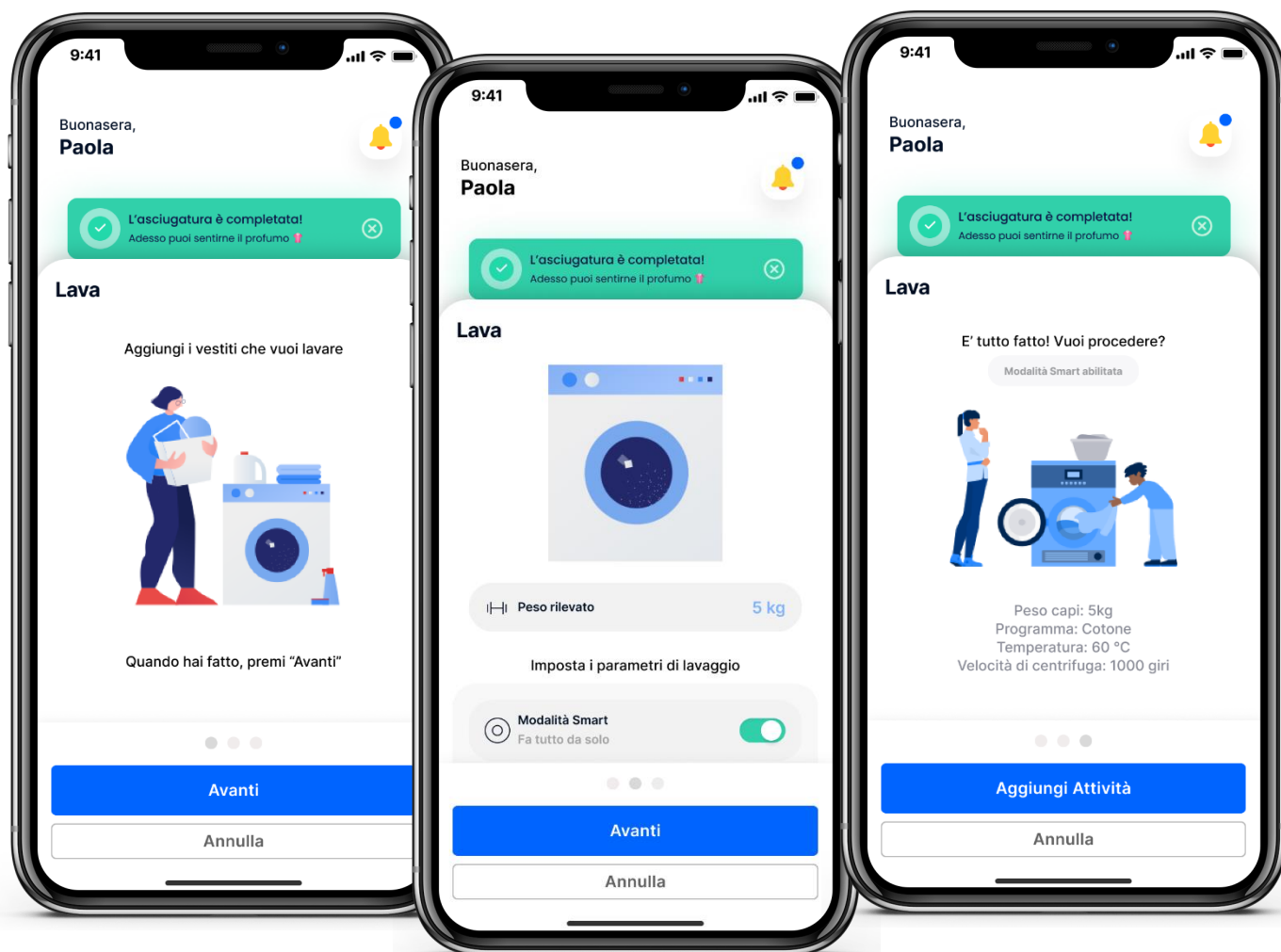


Quando si preme uno dei pulsanti principali della schermata Home, l'interfaccia si trasforma in una sorta di guida virtuale, che accompagna l'utente attraverso i passaggi necessari per eseguire l'operazione desiderata.

Ad esempio, se l'utente sceglie di avviare la procedura di lavaggio, l'applicazione può mostrare una sequenza di immagini animate che indicano come aprire il cassetto del

detersivo, inserire la biancheria, selezionare il programma di lavaggio e così via. Questa procedura guidata rende l'esperienza dell'utente estremamente facile e intuitiva, anche per coloro che non hanno mai utilizzato una lavatrice.

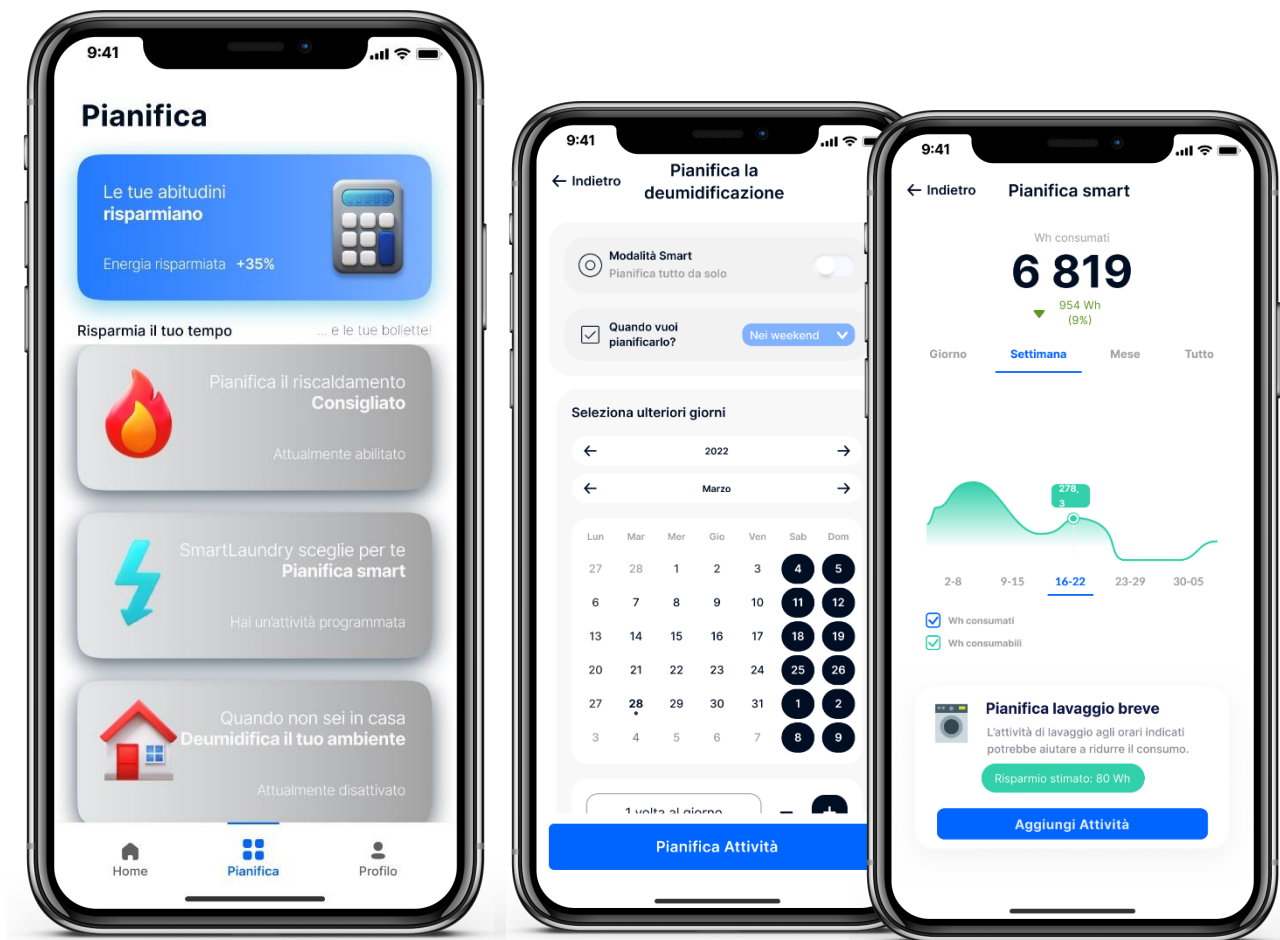
Inoltre, la **funzionalità Smart** offre all'utente un'esperienza ancora più personalizzata e conveniente. Grazie all'uso di algoritmi avanzati di machine learning, l'apprendimento automatico, e sensori avanzati, l'applicazione può raccogliere informazioni sulle abitudini dell'utente, come ad esempio la quantità di biancheria solitamente lavata e la durata dei cicli di lavaggio. In base a queste informazioni, l'applicazione può suggerire automaticamente le impostazioni di lavaggio più **efficienti** dal punto di vista energetico e comode per l'utente, senza la necessità di dover configurare manualmente ogni volta le impostazioni di lavaggio. In questo modo, l'esperienza dell'utente è ancora più personalizzata e conveniente, poiché l'applicazione si adatta alle esigenze specifiche di ciascun utente.



Cliccando sul pulsante delle **attività in corso** in basso alla schermata Home, l'utente può avere accesso a una schermata dettagliata che mostra le operazioni attualmente in corso o completate dall'elettrodomestico. In questa schermata, l'utente può visualizzare i consumi energetici della funzione selezionata e del complesso, in modo da avere sempre sotto controllo quanto sta consumando l'elettrodomestico. Inoltre, vengono mostrati anche il tempo necessario per il completamento dell'operazione e la possibilità di modificare alcuni parametri, come ad esempio la temperatura dell'acqua o la velocità di centrifuga della lavatrice.



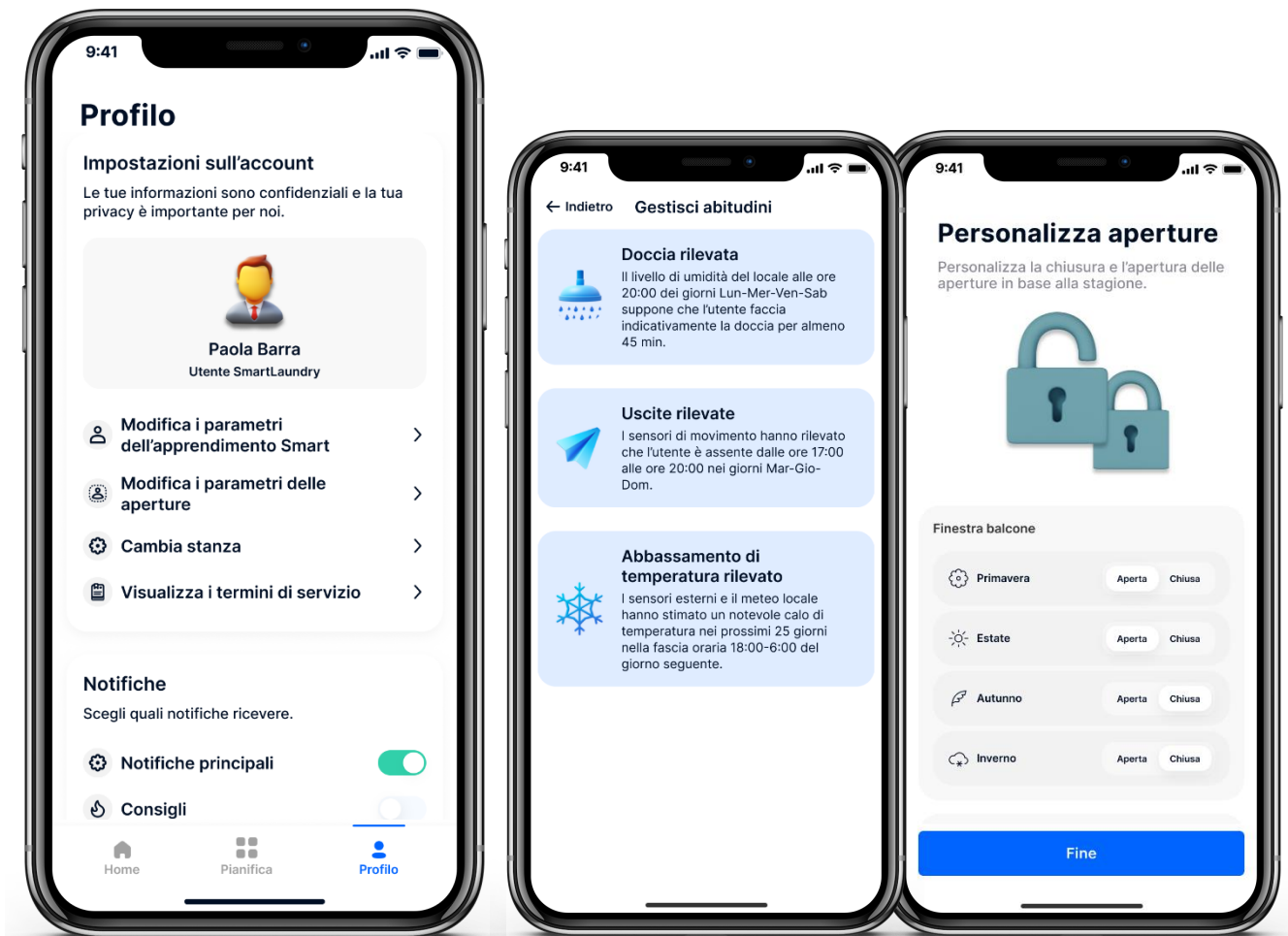
La **schermata Pianifica** è un'ottima risorsa per l'utente per sfruttare al massimo la funzionalità smart del software. Qui, l'utente può trovare suggerimenti utili per risparmiare energia, come l'utilizzo dei sensori di movimento per attivare il deumidificatore quando l'utente esce di casa, o la pianificazione dei tempi di lavaggio e asciugatura del bucato in base agli orari di picco di consumo energetico. Inoltre, l'utente può programmare la temperatura desiderata per il riscaldamento durante gli orari di riposo, in modo da risparmiare energia durante il giorno.



Il terzo pulsante della barra di navigazione consente di accedere alla **schermata Profilo**, dove l'utente può personalizzare ulteriormente il suo utilizzo del nostro elettrodomestico intelligente. Qui, l'utente può modificare le impostazioni principali come la configurazione della stanza, in modo da garantire il corretto funzionamento dei sensori e degli algoritmi di apprendimento automatico.

Inoltre, questa schermata consente all'utente di visualizzare e cancellare i parametri di auto apprendimento del software smart, che consentono all'elettrodomestico di adattarsi alle abitudini dell'utente e di ottimizzare i consumi energetici.

Qui è anche possibile modificare altri piccoli settaggi come le notifiche, che l'utente può personalizzare in base alle proprie preferenze.



Il nostro obiettivo principale nella creazione di questo prototipo è stato quello di realizzare un'interfaccia utente che sia intuitiva ed esteticamente piacevole. Abbiamo lavorato per creare un'esperienza utente semplice e diretta, che permetta agli utenti di accedere facilmente alle funzioni principali e di utilizzare l'applicazione senza difficoltà.

Per quanto riguarda l'aspetto estetico, abbiamo prestato particolare attenzione alla scelta delle forme, dei colori e del font utilizzato. Abbiamo utilizzato un design minimalista, con forme geometriche pulite e colori neutri, che creano un'atmosfera rilassante e accogliente per l'utente.

Metodologia usata

Il testing del software è stato condotto con la partecipazione di 5 utenti che hanno effettuato diverse operazioni. Il software non ha un target di clientela specifico, può essere utilizzato da qualsiasi persona indipendentemente dal sesso, età o qualsiasi altra categoria. In linea con questo principio, il testing è stato effettuato da cinque utenti che hanno età differente, sesso differente e svolgono lavori differenti in modo tale da rispecchiare quanto più possibile il principio di universalità promosso dall'applicazione. Il numero di utenti non è scelto a caso, rispetta la regola dettata da Jacob Nielsen per avere risultati coerenti.

Il test è di tipo informale quindi senza un controllo rigido, ad ogni candidato sono stati affidati diversi tipi di compiti e scenari, affinché però il test risulti essere efficace, lo stesso compito e scenario è stato affidato anche ad altri candidati in modo tale da poter confrontare la metodologia di esecuzione e la tempistica di risoluzione tra i vari soggetti, che, abbiamo detto appartenere a diverse categorie sociali. Questo processo ci aiuta a comprendere in che modo è possibile migliorare il prototipo per facilitare l'uso del software agli utenti.

Di seguito gli oggetti dei test:

- 1) Mario, ragazzo di 20 anni che si è appena trasferito in una casa indipendente per intraprendere il nuovo percorso universitario. Mario non ha mai svolto nessun tipo di faccenda domestica e vorrebbe un software che gli spiegasse in modo semplice i vari passi da eseguire, al fine di rendere meno impegnative operazioni per lui tediose come lavare ed asciugare il bucato.

Tasks:

- Accedere all'applicazione
- Inserire la tipologia di comando(lavaggio/asciugatura) e data di terminazione
- Controllare il peso del bucato restituito dal sensore
- Accettare la notifica per la terminazione del comando.

- 2) Stefania, donna di 67 anni, casalinga di professione, dichiara apertamente di testare il software unicamente per capire se può essere più produttivo di lei ma ne dubita.

Tasks:

- Accedere all'applicazione
- Inserire due comandi in contemporanea senza inserire una data di terminazione

- L'applicazione fornisce la data di terminazione più vicina possibile
- Controllare peso
- Controllare temperatura ambiente
- Accettare la notifica per la terminazione del comando uno.
- Accettare la notifica per la terminazione del comando due.

3) Gennaro, 38 anni, proprietario di un hotel vuole capire se questo nuovo software può aumentare i ritmi di lavaggio/asciugatura per la biancheria della sua struttura.

Tasks:

- Accedere all'applicazione
- Inserire quattro comandi in contemporanea (lavaggio/asciugatura/deumidificazione/riscaldamento) con una data non coerente per quanto riguarda il lavaggio e l'asciugatura.
- L'applicazione fornisce la data di terminazione più vicina possibile a quella richiesta dall'utente
- Controllare peso bucato lavaggio
- Controllare umidità ambiente
- Accettare la notifica per la terminazione del comando uno.
- Accettare la notifica per la terminazione del comando due.

4) Carlo 56 anni, dipendente statale.

Tasks:

- Accedere all'applicazione
- Inserire il comando di deumidificazione aria
- Controllare la temperatura dell'aria
- Accettare la notifica di terminazione comando

5) Emanuela 18 anni, studentessa, non ha mai utilizzato elettrodomestici gestiti da un software.

Tasks:

- Accedere all'applicazione
- Inserire la tipologia di comando(lavaggio/asciugatura) e data di terminazione
- Controllare il peso del bucato restituito dal sensore
- Accettare la notifica per la terminazione del comando.

Sintesi delle misure

Dopo aver descritto i vari utenti che effettueranno il test, vediamo adesso i risultati ottenuti.

SCENARIO 1:

	Task 1	Task 2	Task 3	Task 4	Task 5	Tasso successo	Tempo di esecuzione
Mario	S	S	S	S	P	90%	4 min
Stefania	S	S	S	S	F	80%	6 min
Gennaro	S	S	S	S	S	100%	5 min
Carlo	F	S	S	S	F	0%	8 min
Emanuela	S	S	S	S	S	100%	4 min

Leggenda: S= successo; P = successo parziale; F = fallimento

TASSO DI SUCCESSO = $(21 + (1 * 0,5)) / 25 = 89\%$

SCENARIO 2:

	Task 1	Task 2	Task 3	Task 4	Task 5	Tasso successo	Tempo di esecuzione
Mario	S	S	F	S	P	90%	4 min
Stefania	F	S	S	S	F	80%	9 min
Gennaro	S	S	S	S	P	90%	7 min
Carlo	F	S	S	S	F	80%	8 min
Emanuela	S	S	F	S	F	80%	6 min

Leggenda: S= successo; P = successo parziale; F = fallimento

TASSO DI SUCCESSO = $(16 + (2 * 0,5)) / 25 = 70\%$

SCENARIO 3:

	Task 1	Task 2	Task 3	Task 4	Task 5	Tasso successo	Tempo di esecuzione
Mario	S	S	S	S	S	100%	3 min
Stefania	F	S	S	S	F	80%	7 min
Gennaro	S	S	S	S	P	90%	5 min

Carlo	P	S	S	S	S	90%	8 min
Emanuela	S	S	S	S	P	90%	5min

Leggenda: S= successo; P = successo parziale; F = fallimento

$$\text{TASSO DI SUCCESSO} = (20 + (3 * 0,5)) / 25 = 86\%$$

SCENARIO 4:

	Task 1	Task 2	Task 3	Task 4	Task 5	Tasso successo	Tempo di esecuzione
Mario	S	S	S	S	S	100%	4 min
Stefania	S	S	S	S	P	90%	4 min
Gennaro	S	S	S	S	F	80%	8 min
Carlo	S	S	S	S	F	80%	4 min
Emanuela	P	S	P	S	F	60%	7 min

Leggenda: S= successo; P = successo parziale; F = fallimento

$$\text{TASSO DI SUCCESSO} = (19 + (3 * 0,5)) / 25 = 82\%$$

SCENARIO 5:

	Task 1	Task 2	Task 3	Task 4	Task 5	Tasso successo	Tempo di esecuzione
Mario	S	S	S	P	S	90%	4 min
Stefania	S	S	S	P	F	70%	9 min
Gennaro	S	S	S	S	S	90%	6 min
Carlo	F	F	F	S	F	10%	10 min
Emanuela	S	S	S	S	P	90%	5 min

Leggenda: S= successo; P = successo parziale; F = fallimento

$$\text{TASSO DI SUCCESSO} = (17 + (3 * 0,5)) / 25 = 74\%$$

Analisi dei risultati

Dopo aver effettuato il test ed aver portato a termine ogni scenario e relativi task assegnati, ogni utente ha rilasciato un proprio feedback sulla piattaforma, inserendo nella sezione apposita le problematiche riscontrate, proponendo anche le possibili personali soluzioni.

Le criticità indicate verranno poi analizzate dagli sviluppatori cercando di creare il giusto compromesso con quello richiesto dall'utente e quello effettivamente realizzabile. Tutto questo viene fatto al fine di aiutare i produttori a migliorare la piattaforma in base alle criticità evidenziate dagli utenti finali cercando di rendere ogni parte del software quanto più intuitiva possibile creando una universalità di utilizzo.

Di seguito una tabella dei problemi riscontrati:

	DESCRIZIONE PROBLEMATICATA	INDICE DI PRIORITA'
PROBLEMATICATA1	Dopo aver inserito più ordini in contemporanea, l'utente ha trovato difficoltà nel comprendere come visualizzare gli avanzamenti di stato di ogni singolo ordine inserito.	A
PROBLEMATICATA2	Durante la fase di inserimento date, in caso di non coerenza di quest'ultime, il software proponeva delle date non abbastanza vicine a quelle richieste dall'utente.	A
PROBLEMATICATA3	L'aggiornamento delle temperature per alcuni utenti era ritenuto troppo lento rispetto ad altre applicazioni.	M

Legenda: A = priorità alta; M = priorità media

Sintesi delle intervistate

Dopo aver analizzato le problematiche riscontrate in fase di testing, andiamo adesso a soffermarci sulle possibili soluzioni proposte dagli utenti finali.

Di seguito una raccolta di “interviste” degli utenti:

- 1) Mario Esposito, studente universitario di 20 anni, in generale ha riportato un feedback positivo, data la sua dimestichezza con l'utilizzo di applicazioni. La sua esperienza con la piattaforma è stata intuitiva e fluida, riuscendo a portare a termine la maggior parte dei tasks senza troppi problemi. Una segnalazione da lui effettuata riguarda l'operazione di proposta delle date di terminazione dei comandi inseriti. Secondo l'utente le proposte da parte del software sono comunque troppe distanti da quelle inserite in input. Tra le possibili sue soluzioni troviamo quella di migliorare i tempi medi di ogni operazione per rendere il tutto ancora più efficiente.
- 2) Stefania Marino, casalinga di 67 anni, la sua esperienza può essere considerata nel complesso buona, le principali difficoltà da lei riscontrate riguardano l'utilizzo dell'interfaccia. Una possibile soluzione da lei proposta per questa problematica è l'inserimento di un tutorial avanzato al primo accesso al fine di rendere più comprensibile ogni sezione dell'applicazione.
- 3) Gennaro Rossi, imprenditore di 38 anni, la sua esperienza è nel complesso positiva. L'utente è riuscito a “muoversi” all'interno dell'applicazione in modo agile e intuitivo definendo l'interfaccia ottima per il sistema proposto. Come l'utente Mario Esposito, anche lui vorrebbe una maggior celerità di esecuzione da parte di tutti i device, proponendo un miglioramento degli algoritmi di smistamento richieste.
- 4) Carlo Politano, dipendente statale di 56 anni, la sua esperienza è definita discreta, anche egli come l'utente Stefania Marino ha riscontrato delle difficoltà nell'utilizzo dell'interfaccia proponendo una più semplice e intuitiva progettazione della suddetta.
- 5) Emanuela Fiore, studentessa universitaria di 18 anni, il suo feedback è positivo, è riuscita a svolgere tutte le task senza problemi riuscendo a sfruttare l'interfaccia per “muoversi” attraverso le varie sezioni, definendola molto semplice e apprezzandone il design. L'unica problematica da lei riscontrata riguarda l'aggiornamento delle condizioni metereologiche non sempre

eseguito in modo rapido. Una sua possibile soluzione è quella di inserire una sezione per calibrare i sensori al fine di rendere più rapido e preciso il processo di calcolo sia della temperatura che dell'umidità.

Raccomandazioni finali

In questa sezione analizzeremo gli interventi migliorativi per le problematiche riscontrate dagli utenti nel capitolo precedente.

Di seguito la tabella esplicativa:

NUMERO PROBLEMA	MIGLIORAMENTI	INDICE DI PRIORITA'
Problema 1	Realizzare un'interfaccia grafica ancora più intuitiva e inserire al primo accesso un tutorial avanzato per descrivere le varie sezioni dell'applicazione.	1
Problema 2	Rendere ancora più efficiente la fase di parallelizzazione degli ordini, al fine di restituire le date di terminazione previste quanto più vicine a quelle inserite dall'utente	1
Problema 3	Inserire sezioni di configurazione dei sensori con appositi video tutorial per eventuali modifiche.	3

Legenda: 1 = Intervento indispensabile; 3 = intervento auspicabile

Allegati

Di seguito la tabella anagrafica dei tester:

NOME	COGNOME	ETA'	LUOGO DI NASCITA	SESSO
Mario	Esposito	20 anni	Salerno	M
Stefania	Marino	67 anni	Napoli	F
Gennaro	Rossi	38 anni	Napoli	M
Carlo	Politano	56 anni	Avellino	M
Emanuela	Fiore	18 anni	Caserta	F

Valutazione dell'usabilità

Una volta che il software supera con successo la fase di test, prima di essere rilasciato sul mercato oppure consegnato al cliente, gli sviluppatori devono valutare l'usabilità del sistema basandosi su uno standard preciso: ISO 9241-110. Questo standard descrive sette caratteristiche riguardanti interazione tra utente e sistema.

- 1) **Adeguatezza al compito:** Questo principio afferma che la modalità di esecuzione di ogni singola operazione deve essere adeguata al compito che l'utente deve portare a termine.

Voto: 4

- 2) **Auto-Descrizione:** Questo principio afferma che il sistema deve comunicare all'utente, in ogni momento, quali sono le possibili operazioni che può effettuare e il suo status attuale. Tutto questo nel nostro sistema è rispettato tramite le varie operazioni di monitoraggio di stato rispetto ogni parametro interessato, come ad esempio il tempo rimanente alla fine di un operazione, la temperatura e l'umidità dell'aria.

Voto: 3.5

- 3) **Conformità alle aspettative:** Questo principio afferma che il dialogo deve essere coerente con quanto specificato durante l'analisi iniziale. Il nostro Software risulta rispettare questo principio anche se alcuni utenti nella fase di test hanno rilevato alcune problematiche di comprensione al primo accesso sulla piattaforma. Alcuni tester, infatti, hanno suggerito dei tutorial introduttivi.

VEDI CAPITOLO ANALISI DEI RISULTATI

Voto: 3

- 4) **Adeguatezza all'apprendimento:** Questo principio afferma che il dialogo tra utente e sistema deve essere in grado di aiutare l'utente nell'apprendimento del sistema. Questo principio è legato nel nostro caso a quello della conformità delle aspettative, anche in questo caso il nostro sistema rispetta in parte questo principio dato il riscontro di alcune difficoltà al primo accesso dell'utente tester.

Voto: 3

- 5) **Controllabilità:** Questo principio afferma che deve essere l'utente a guidare il sistema e non il contrario. Deve essere l'utente a scegliere quali operazioni effettuare, quando interromperle e se modificare gli input impartiti al sistema. Nel nostro caso questo principio è rispettato date le varie funzioni semplici da utilizzare per la modifica, interruzione e ripresa di una o più operazioni in esecuzione. L'utente però sarà comunque indirizzato in modo semiautomatico

verso un utilizzo corretto e sicuro dell'intero software attraverso l'attore precedentemente descritto, ovvero lo **scheduler**.

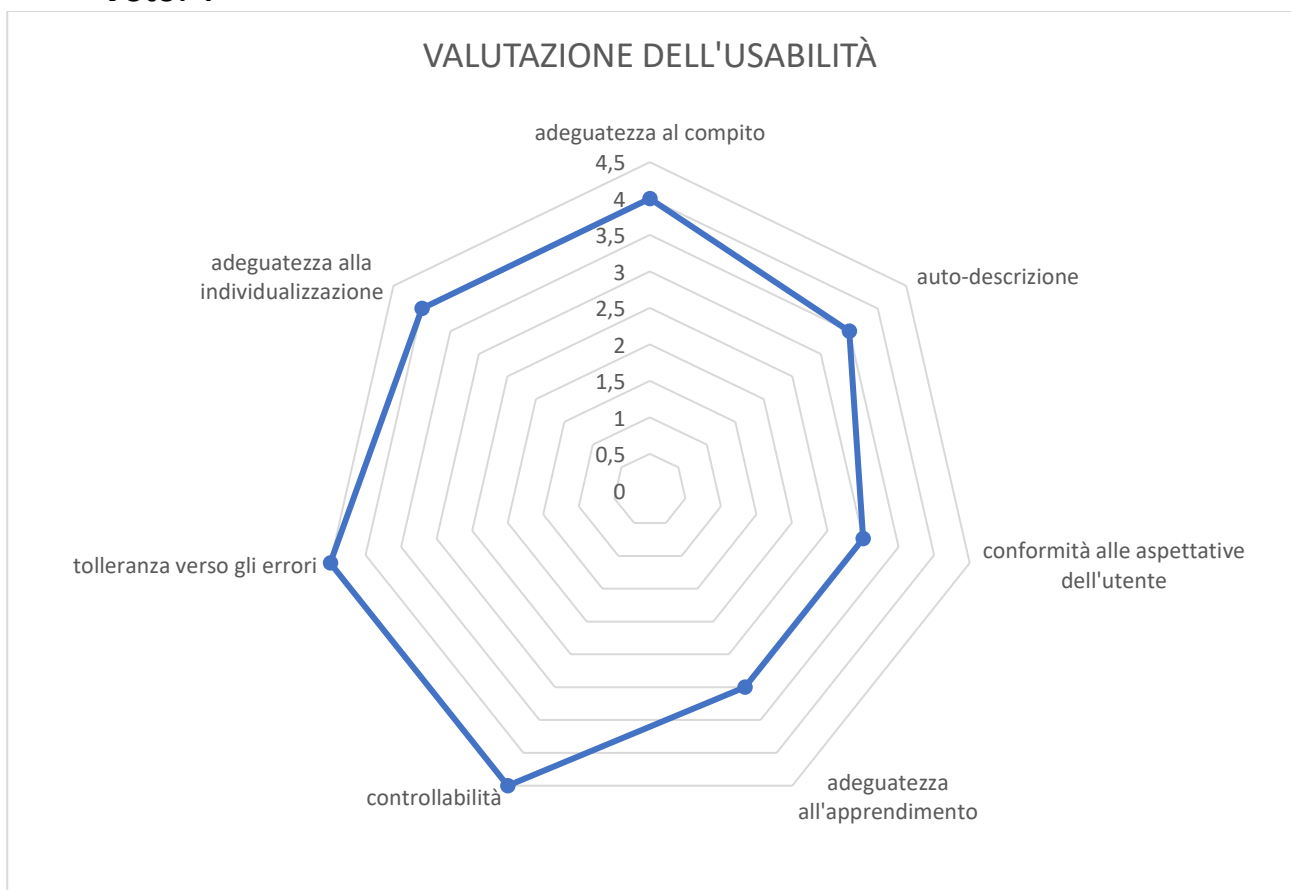
Voto: 4,5

- 6) Tollerabilità all'errore:** Questo principio afferma che il sistema deve essere in grado di fornire i risultati richiesti anche in presenza di errori da parte dell'utente. Questo principio è rispettato tramite l'utilizzo da parte del software di funzioni automatiche, che si occupano di gestire casi in cui ci sono errori negli inserimenti in input. Il sistema quindi si occupa anche di produrre in casi eccezionali parametri coerenti quanto più vicini a quelli inseriti dagli utenti. In fase di test però sono stati segnalati dei parametri leggermente distanti da quelli inseriti.

Voto: 4

- 7) Adeguatezza all' individualizzazione:** Questo principio afferma che Il sistema deve presentare funzioni e informazioni in modo accattivante e motivante, cercando di suscitare una continua interazione con il sistema. Questo principio è in parte rispettato attraverso l'utilizzo di notifiche ricorrenti basate sullo studio dei dati inseriti volta per volta dall'utente. Il sistema salverà i dati creando delle "abitudini" circa l'esecuzione delle varie operazioni al fine di rendere più efficiente la sua esperienza.

Voto: 4



Glossario

User: utente che usufruisce del sistema.

SmartController: gestore di elettrodomestici

Scheduler: gestore di funzioni simultanee

SmartLaundry: è il software che gestisce le funzioni