D4.1 - 1st Release of DIGITISE Cross-Sector Services and Applications for End-Users

Project number	101160671
Project acronym:	DIGITISE
Project title:	Digital Innovative cross-sector services for Greater citizen Integration in a just energy Transltion, and Societal Empowerment

Work Package	WP4: Cross-Sector Services and Applications for End Users
Responsible Partner	UCD
Official Submission Date	31/10/2025
Actual Submission Date	31/10/2025
Type	Other
Dissemination Level	Public
Reviewers	APPART, ARDEN
Version	v1.0

Funded by the European Union

Versioning and contribution history

Version	Date	Author(s)	Notes
0.1	5/8/25	UCD	ToC added
0.2	29/9/25	UCD	Health and Security Application added
0.3	13/10/25	UCD	Updated the applications' format and added Introduction
0.4	24/10/25	UCD, CIRCE, UNP, APPART	Finalized contribution from all partners, Introduction, Executive summary, and Conclusion
0.5	27/10/25	UCD, CIRCE, UNP, APPART	Addressed initial comments by all partners
0.6	30/10/25	UCD, CIRCE, UNP, APPART	Addressed final comments by all partners
0.7	30/10/25	UCD	Editing and minor content adjustment
1.0	31/10/2025	UBITECH	Final check, version for submission

Copyright notice

Disclaimer

Preface

Funded by the European Commission under Grant Agreement number 101160671, DIGITISE is a project focused on enhancing the digital literacy and empowerment of consumers and prosumers in the energy sector. By integrating advanced technologies and fostering active engagement in digital energy activities and markets, DIGITISE aims to play a crucial role in the global energy transition

Executive Summary

The transition towards smarter, more sustainable, and user-centric buildings is central in the EU's directives. This can be achieved by integrating digital solutions empowering consumers, which in turn enhances household performance across multiple dimensions. Via the use of artificial intelligence modules, the users can achieve optimization at personal and household levels. In the scope of DIGITISE four complementary applications are designed to optimize energy use inside the household and extend household intelligence beyond the energy domain.

First, energy savings and self-consumption application enables consumers to actively participate in self-consumption schemes, maximizing energy and cost savings through intelligent monitoring, analytics, and control mechanisms. Second, the asset sizing and investment optimization application provides evidence-based insights and tailored recommendations to facilitate informed decisions on energy-related investments, promoting financial accessibility and transparency. In parallel, the DIGITISE health and security application enhances the well-being and safety of occupants through data-driven intelligence and adaptive control over the devices of the household. Last, all the information and outputs of the aforementioned applications are displayed on a dashboard. This does not only provide a holistic view to prosumers on the functionalities offered but also enables controllability of their devices.

Together, these applications establish an interconnected ecosystem of smart services that drive positive behavioural changes, optimize resource usage, and promote active citizen participation in the evolving digital and sustainable energy landscape.

Funded by the European Union

Table of Contents

Funded by the European Union

List of Figures

Funded by the European Union

List of Tables

Funded by the European Union

Abbreviations

Abbreviation	Full Name
H-EMS	Home Energy Management Systems
RES	Renewable Energy Systems
PV	Photovoltaic
BESS	Battery Energy Storage System
KPI	Key Performance Indicator
MILP	Mixed-Integer Linear Programming
IRR	Internal Rate of Return
ROI	Return on Investment
AI	Artificial Intelligence
IAQ	Indoor Air Quality
SRI	Smart Readiness Indicator
DR	Demand Response
SPA	Single-Page Application
GDPR	General Data Protection Regulation
UI	User Interface
DER	Distributed Energy Resources
ADCE	Automated Device-Level Control Execution

Funded by
the European Union

1 Introduction and objectives

This deliverable (D4.1) provides details revolving around the first version of the DIGITISE cross-sector services and applications for end users. These are all designed and implemented under the scope of WP4 of the DoA and their details regarding components, features, and employed technologies are in line with D2.2. Deliverable D4.1 serves as an intermediate checkpoint in the realization of each individual application and includes information on the 1st version of the respective work performed in WP4. More specifically:

- 2.1 maps to task 4.1, Energy Savings and Self Consumption Optimization Application,
- 2.2 expands on task 4.2, Asset Sizing and Investment Optimization Application,
- 2.3 corresponds to task 4.3, the Health and Security Application,
- 2.4 showcases the Smart Home and DER Automated Control system, and
- 2.5 provides details for task 4.4, which embodies the Consumer Application for Personalized Empowerment and Capacity Building.

Overall, the aim is to develop a suite of services centered on maximizing the wellbeing and energy efficiency of the prosumers. More concretely, the following objectives are targeted:

1) Engage consumers in self-consumption while increasing energy and costs savings.
2) Guide consumers via a financing support tool in energy-related investments.
3) Enhance and extend the smartness of the household beyond the energy sector, ensuring health and security standards across the household.
4) Provide the consumers with a tool integrating both conceptually and visually all digital services and offering an omni-channel experience for citizens literacy enhancement and active engagement.

Based on the current status of the applications, the backbone infrastructure has been implemented alongside basic functionalities; crucial steps towards the realization of the task objectives. Last, the development plan has been specified, establishing a route towards the completion of all features to be delivered at the final 2nd version of the applications.

1.1 Structure of the document

The sections below follow a similar structure. First, a general overview is given followed by an analysis of the currently supported functionality of the respective applications and the technology stack behind it. Then, the assumptions and constraints in developing the 1st version of the applications are presented, and each section concludes with listing the development plan for the 2nd and final version of the application.

Funded by the European Union

2 DIGITISE End-User Applications

2.1 Energy Savings and Self Consumption Optimization Application

2.1.1 Overview

Description of the application. The energy savings and self-consumption optimization application is built on the principles of Home Energy Management Systems (H-EMS). Its core components include renewable energy systems (RES), specifically photovoltaic (PV) systems, for self-consumption and grid feeding, energy drawn from the grid while considering dynamic pricing, and a battery energy storage system (BESS). Energy demand is a central element of the system and must always be met. Energy can be drawn from the RES, the BESS, or the grid, but the primary goal is to ensure that demand is fully covered while maximizing self-consumption of renewable energy.

The optimization engine considers forecasted energy demand, expected renewable generation, and projected energy prices to determine the most effective strategy for energy storage and BESS usage. As outputs, this application reports key performance indicators (KPIs), presents key energy profiles, and provides actionable recommendations that require either user engagement or automatic control where system is enabled to system to translate these insights into real-time, actionable responses. In Annex I an in-depth analysis of the state of the art on EMS optimization algorithms is provided.

Components and Integration Features. As noted in deliverable D2.2, the application is designed to interact with other applications, including the data space and the dashboard, and these interactions will be reported in the next version of the deliverable. The application retrieves data from the data space via APIs employs in optimization processes, creating optimal outputs, and key profiling outputs which are shared with the dashboard application via a dedicated API.

Integration with external components is currently under development. Data exchange structures and API endpoints have been defined and tested in simulation, with full deployment planned for the next version.

Funded by
the European Union

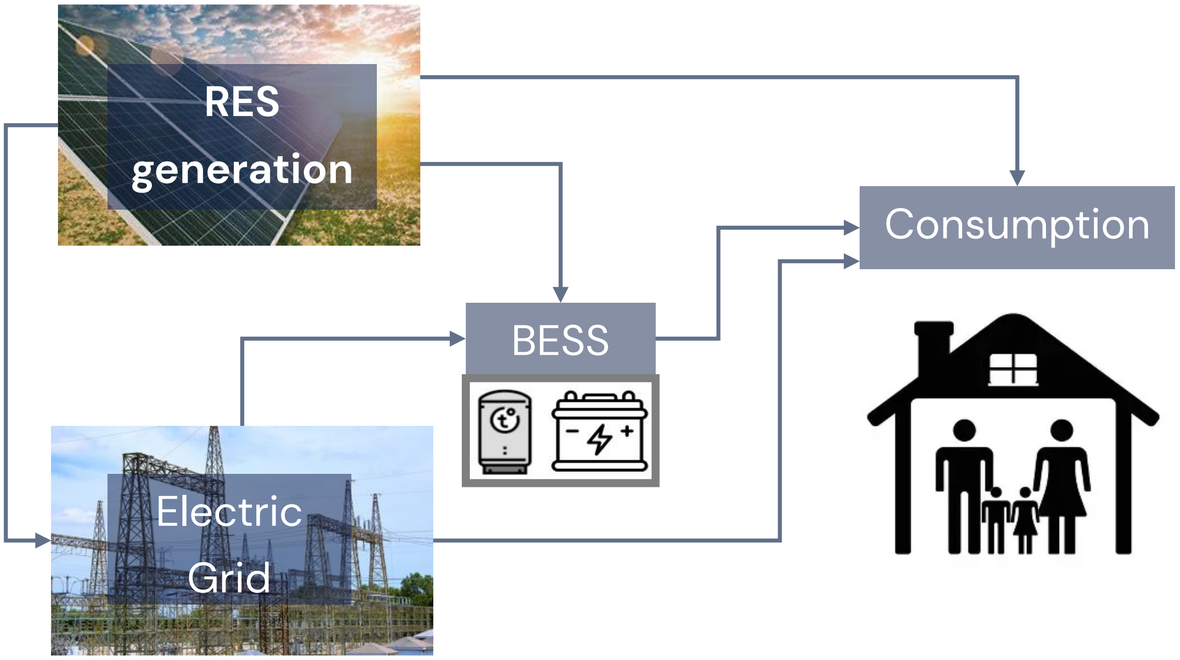

FIGURE 1. HOME ENERGY MANAGEMENT SYSTEMS COMPONENTS

2.1.2 Delivered Functionality

The first version of the Energy Savings and Self-Consumption Optimization Application focuses on defining the core functionalities of the optimization engine. The following table summarizes the features implemented and their current development status. Notably, the DER controller will be implemented in the next version of the application.

TABLE 1 – FEATURES OF ENERGY SAVINGS AND SELF-CONSUMPTION OPTIMIZATION APP FOR 1ST VERSION

Feature ID	Feature Name	Development status for v1
ESSCO – 1	Real-Time Monitoring data integration	Partially complete – started collecting pilot data using API.
ESSCO – 3	Battery model configuration	Partially complete – basic storage model implemented; parameter tuning and degradation modelling under development.
ESSCO – 4	Dynamic Control and Flexibility Management	Complete – functional battery model integrated into the MILP with charge/discharge limits, efficiencies, state-of-charge dynamics, and degradation.
ESSCO – 5	Comprehensive Energy Profiling	Partially complete – implemented using prototype data to validate system behaviour and KPIs.

2.1.2.1 Technology Stack

For the final foreseen application, the integrated architecture is structured around three synchronized layers, Data Space, Optimization Engine, and the User Interface (see Figure 2). It operates in continuous feedback to connect data acquisition, model execution, and connectivity visualization. Communication between these layers will be established through a RESTful API built on the HTTP protocol and JSON data exchange, ensuring a lightweight, interoperable, and secure channel for seamless integration across the system's data-value chain.

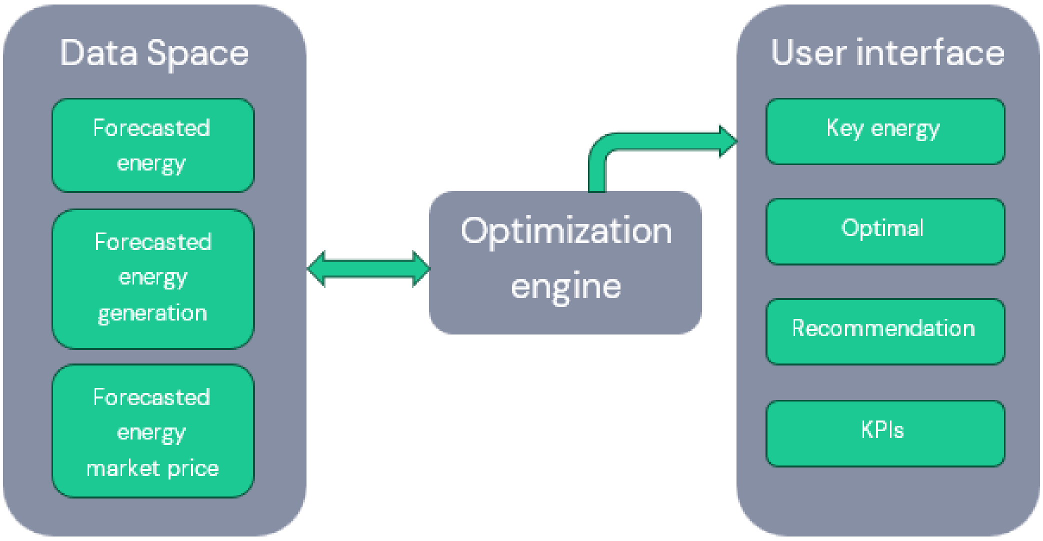

FIGURE 2. ARCHITECTURE OF THE APPLICATION

The optimization Engine acts as the system's intelligence layer. Starting to be developed in Python with advanced open-source libraries (CVXPY, HiGHS, ECOS_BB), it executes Mixed-Integer Linear Programming (MILP) within a 24-hour rolling horizon. Although not yet completed, the optimization framework is being developed with a modular architecture suitable for deployment in various environments. It enables scalable, near-real-time decision-making for energy scheduling, battery storage management, and price-based adjustments.

In the current state, a dedicated interface was implemented through web frameworks such as Dash, Streamlit, an open-source environment, which serves as the visualization and interaction layer at HTLM format. It delivers interactive dashboards that present optimization outcomes and energy system metrics. The optimization methodology was defined and tested to operate on a 24-hour rolling horizon, ensuring short-term forecasts inform daily energy scheduling decisions. The hourly results from each daily optimization are subsequently aggregated across the entire simulation period to determine total and cumulative energy costs. During the initial phase testing, some key profiles are being explored using historical data from the Spanish pilot site in Murcia, applying representative and configurable parameters to validate the proposed optimization methodology.

Figure 3 and Figure 4 illustrate two representative days, one in summer (June) and another in winter (December), used to validate the optimizer's behavior under different operating conditions. In both cases, the algorithm dynamically determines the optimal energy flows based on hourly electricity prices, demand profiles, and photovoltaic generation availability. As observed, battery charging can occur both from photovoltaic production and from the grid when prices are low, enabling stored energy to be used later during periods of higher costs or reduced generation.

Figure 5and Figure 6highlight the economic savings of the four scenarios in terms of cumulative costs and hourly savings, illustrating how the optimized control decisions influence the economic evolution of the system throughout the simulation period. The validation included three reference cases aimed at evaluating the benefits provided by the optimization, focusing the comparative analysis on the economic results and the maximization of PV consumption. The first scenario represents the baseline situation, in which the total electricity demand is supplied from the grid without local generation. The second scenario introduces photovoltaic generation, prioritizing self-consumption without including storage. The third scenario considers a system with photovoltaic generation and battery storage operated through predefined logic, without mathematical optimization. Finally, the optimized scenario applies the developed MILP-based control to the same technical configuration, determining the charging, discharging, and import/export strategies that minimize the total daily cost.

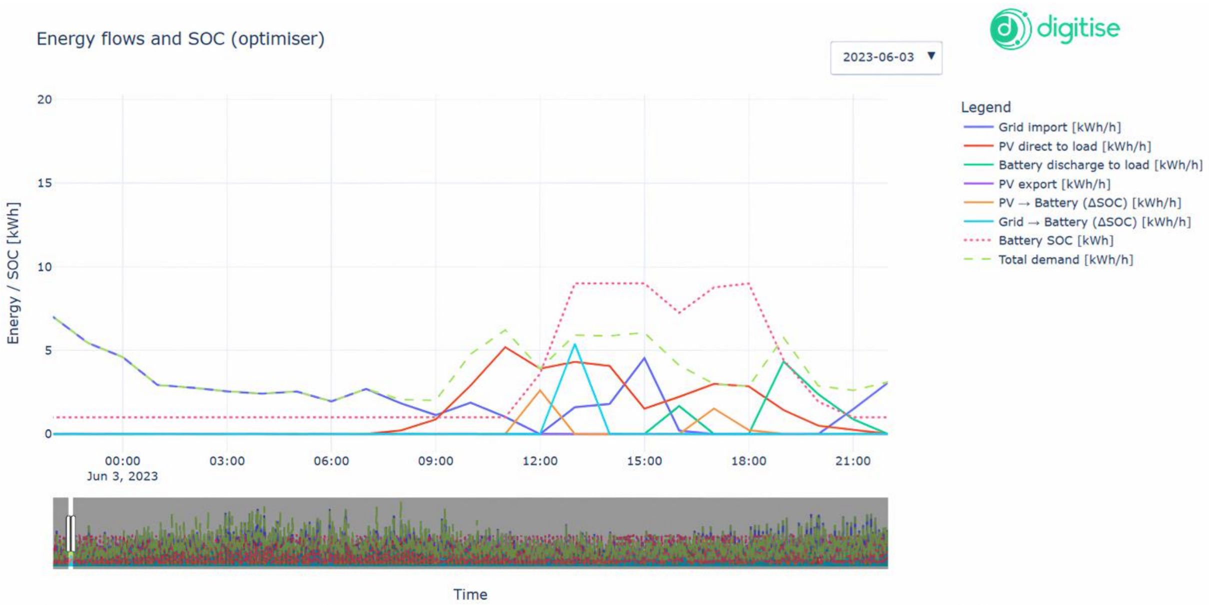

FIGURE 3. PROTOTYPE V.1.0 TESTS WITH SPANISH PILOT HISTORICAL DATA AS. 03 JUNE 2023 — 24-HOUR MILP OPTIMIZATION SHOWING ENERGY FLOWS AND BATTERY SOC (DIGITISE ENGINE)

Funded by
the European Union

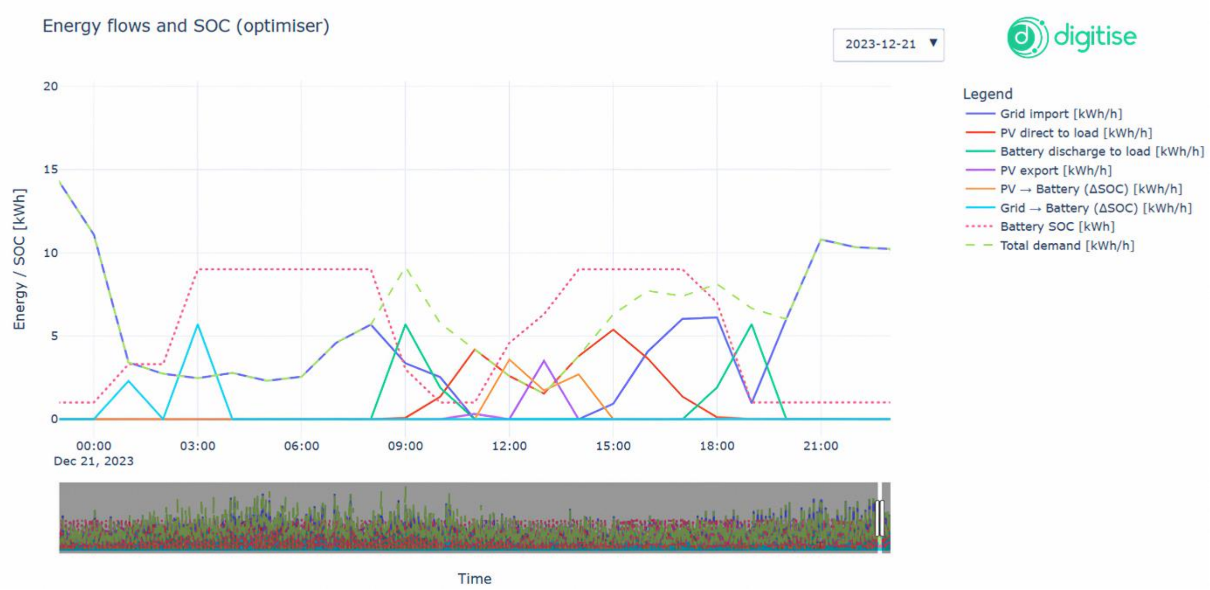

FIGURE 4. PROTOTYPE V.1.0 TESTS WITH SPANISH PILOT HISTORICAL DATA. 21 DECEMBER 2023 — 24-HOUR MILP OPTIMIZATION SHOWING ENERGY FLOWS AND BATTERY SOC (DIGITISE ENGINE)

FIGURE 5. *ECONOMIC SAVINGS. COSTS DAILY VIEW*

FIGURE 6. *ECONOMIC SAVINGS. COSTS* FULL-YEAR VIEW

2.1.3 Assumptions and Constraints

This version of the application has been built with several practical limits and starting conditions in mind:

1) **Connectivity:** The development of the application in v1 has been realized internally. No links to the Data Space or Dashboard are realized in this version, although some efforts for the integration have been initiated. That includes evaluating the characteristics of the pilots, defining application output features, obtaining data through dedicated APIs into a Python interface, analyzing and reporting key profiles and scenarios.
2) **Evaluation:** In all the tests, we applied the optimization logic while keeping real-world technical and operational limitations in mind. Key assumptions considered include:
    a. The model works perfectly for energy demand, solar (PV) generation, and electricity prices.
    b. The model meets the demand at every moment.
    c. The model stays within battery limits, i.e., capacity, and maximum/minimum charging/discharging of the battery.
    d. The system prioritizes using solar energy for immediate consumption.
    e. If there's any excess, it can either be stored in the battery or sold back to the grid, depending on current price signals.
    f. Charging from the grid is only allowed when prices are low.
    g. Battery discharge happens during expensive periods or when solar energy is scarce.

3) **Functionality:** The first version already includes a simple optimization algorithm for different scenarios. However, fine-tuning and improvement of the algorithm are required to adapt to the project´s need.

2.1.4 Development plan for next version

In the next version, the focus will be on developing the application further, including more features. That includes using refined forecasted data for the optimization, and together with enhancing the connectivity between the different applications in the DIGITISE application suite, i.e., the Data space and the user dashboard. More features to be reported in the next version are highlighted below.

TABLE 2 – FEATURES OF THE ENERGY SAVINGS AND SELF-CONSUMPTION OPTIMIZATION APP FOR 2ND VERSION

Feature ID	Feature Name
ESSCO – 1	Real-Time Monitoring data integration
ESSCO – 2	AI-Based Forecast Integration
ESSCO – 3	Battery model configuration
ESSCO – 4	Dynamic Control and Flexibility Management
ESSCO – 5	Comprehensive Energy Profiling
ESSCO – 6	Personalized Optimization Recommendations
ESSCO – 7	KPI Reporting
ESSCO – 8	Secure Data Exchange via Data Space

2.2 Asset Sizing and Investment Optimization Application

2.2.1 Overview

The Asset Sizing and Investment Optimization Application helps households/consumers make informed investment decisions to maximize self-consumption by integrating RES (Renewable Energy Sources) assets and enhance flexibility. Common examples of RES include solar power, wind energy, hydroelectric energy, and geothermal energy. The integration of RES aims to reduce greenhouse gas emissions, promote sustainable energy consumption, and enhance energy security.

This application aims to incorporate financial data to evaluate investments in RES, storage and electromobility. The Asset Sizing application objective is to analyse the pricing of each individual asset and provide the overall financial investment needed form the consumer point of view, but also some important investment metrics.

This application can provide a list of metrics that are described below:

- **Cost-effectiveness**: the economic efficiency of an investment, determining whether the financial outlay is justified by the benefits received. This involves comparing the costs of implementing RES with the savings generated over time through reduced energy bills and potential incentives or subsidies.

- **Payback period**: measures the time it takes for an investment to recoup its initial costs through the financial benefits it generates. A shorter payback period indicates a quicker return on investment, which is often more attractive to investors.
- **IRR (Internal Rate of Return)**: is a financial metric used to evaluate the profitability of an investment. A higher IRR indicates a more profitable investment.
- **ROI (Return on Investment)**: is a measure of the gain or loss generated by an investment relative to the amount of money invested. ROI provides a simple gauge of an investment's efficiency and is often used for comparing the profitability of multiple investments.

These metrics are crucial for evaluating the financial viability and timeline for returns on investments in renewable energy sources (RES) for households/consumers.

The Asset Sizing and Investment Optimization Application also can provide a list of technical key features that are described below:

- **Identify user objectives (IUO):** the app will be able to request from the User their goals and objectives for the asset sizing, in terms of what assets to invest or what type or limits they are envisioning.
- **Create alternative scenarios (CAS):** based on the user goals and objectives, the asset sizing application will create multiple alternative scenarios which will be optimized to understand which scenarios provide the best results.
- **Analise each alternative scenario (AEAS):** each of the scenarios will be analysed based on the new added assets. This will include the financial aspect of the added assets.
- **Provide reports based on the analysed scenarios (PRAS):** The asset sizing application will provide a comprehensive analysis of the scenarios, identifying the necessary investments to be made, its initial cost, cost-effectiveness, payback period, IRR and ROI.
- **User feedback and adjustments (UFA):** the users will be able to provide feedback to the recommendations from the asset sizing application, by requesting additional assets or changing some of the initial parameters (objectives and goals) to re-analyse possible investments in the household and its financial impact.

2.2.1.1 Technology Stack

The Asset Sizing and Investment Optimization application is developed using the React framework[1] which allows to build visual elements that can be combined into pages, screens or apps. The backend is developed using Meteor[2], which is a full-stack Javascript

[1] https://react.dev
[2] https://www.meteor.com

Funded by the European Union

platform with a key set of technologies for building connected-client reactive applications.

Despite this application should work as a backend server and interacting only with the Dashboard frontend, for development and testing purposes, a simple UI was developed to ensure that all the functionalities developed for the application can be tested and work as expected. This way facilitating all technical tests and validations before the integration work can be done between this App and the DIGITISE Dashboard.

Since this application needs to fetch prices from stores, is important to keep historical search data which can be useful to avoid repeated queries or whenever there is no internet connection to retrieve new updated data. These historical data are stored in a Meteor database and can be visualized by the User in the UI.

Also, as part of the application development, a list of APIs is being developed to allow interactions to and from the App. These APIs will be available by using the Express.js[3] which is a Node.js web application framework that can provide a set of methods to be consumed for any application. As a demonstration of the implementation of functionalities of the application, the Figure 7 represents a code snippet of the functionality of retrieving prices from the stores (CAS-02).

```
async _fetchItem(searchSite: string, searchItem: string): Promise<SearchResult[] | undefined> {

    const _str = `site:${searchSite} "${searchItem}"`
    const safeSearch = "active"
    const ui_language = "pt"
    const user_location = "pt"

    const url = "STORE_API_RUL"

    const headers = {
        'accept': 'application/json'
    }

    const config = {
        url: url,
        method: "GET",
        headers: headers
    }

    try {
        const res = await axios(config)

        const resultData = res.data

        if (!resultData.items) return

        const results = resultData.items.map((el: any) => {

            const _result = new SearchResult()
            _result.getObject(el)
            return _result
        })
        return results.filter((el: any) => el.price)
    } catch (e) {
        throw e
    }
}
```

[3] https://expressjs.com/

FIGURE 8 RETRIEVE PRICES FROM STORES: CODE EXAMPLE

```
API Object {
  kind: 'customsearch#result',
  title: 'Buy Solar Panel AIKO 600WP 144 CELLS N TYPE | Tienda Solar',
  htmlTitle: 'Buy <b>Solar Panel</b> AIKO 600WP 144 CELLS N TYPE | Tienda Solar',
  link: 'https://tienda-solar.es/en/solar-panels/1772-solar-panel-aiko-600wp-144-cells-n-type',
  displayLink: 'tienda-solar.es',
  snippet: 'Buy AIKO Solar Panel 600WP 144 CELLS N TYPE with 24% efficiency and low temperature coefficient | Top brands at best price.',
  htmlSnippet: 'Buy AIKO <b>Solar Panel</b> 600WP 144 CELLS N TYPE with 24% efficiency and low temperature coefficient | Top brands at best price.',
  formattedUrl: 'https://tienda-solar.es/.../solar-panels/1772-solar-panel-aiko-600wp-144-cell...',
  htmlFormattedUrl: 'https://tienda-solar.es/.../<b>solar-panel</b>s/1772-<b>solar-panel</b>-aiko-600wp-144-cell...',
  pagemap: {
    offer: [ [Object], [Object], [Object], [Object] ],
    cse_thumbnail: [ [Object] ],
    product: [ [Object], [Object], [Object], [Object] ],
    metatags: [ [Object] ],
    cse_image: [ [Object] ],
    hproduct: [ [Object], [Object], [Object], [Object] ]
  }
}
```

FIGURE 9 RETRIEVE PRICES FROM STORES: API RESULT

Since the API response has many information that is beyond the application needs, is important to model the response into a most user-friendly object to be used within the application. For application purposes, a new object is created by parsing some fields from the API result and therefore, create an object with the most relevant information that could be useful for the user. In the Figure 10 is represented an example of the object created with the information retrieved from the API.

```
App Object SearchResult {
  id: undefined,
  title: 'Buy Solar Panel AIKO 600WP 144 CELLS N TYPE | Tienda Solar',
  link: 'https://tienda-solar.es/en/solar-panels/1772-solar-panel-aiko-600wp-144-cells-n-type',
  name: 'LONGI Solar Panel 455Wp HPBC',
  description: 'Buy AIKO Solar Panel 600WP 144 CELLS N TYPE with 24% efficiency and low temperature coefficient | Top brands at best price',
  image: 'https://cdn.tienda-solar.es/7644-thickbox_default/solar-panel-aiko-600wp-144-cells-n-type.jpg',
  imageHeight: '1422',
  imageWidth: '1100',
  pricePreTax: '100',
  pricePreTaxCurrency: 'EUR',
  price: '121',
  priceCurrency: 'EUR',
  weight: '24.500000',
  weightUnit: 'kg'
}
```

FIGURE 11 RETRIEVE PRICES FROM STORES: APPLICATION OBJECT

The object generated by the application represents an example of a single solar panel retrieved through the API, but it can also represent the information related to a battery or other asset. This object contains all the necessary data used to create alternative scenarios and propose suitable solutions to the user. This information will be baseline for the recommendations provided by the app.

2.2.2 Delivered Functionality

In **Error! Reference source not found.**, the list of features of the first version of the application is provided, alongside their development status.

Funded by
the European Union

TABLE 3 – FEATURES OF ASSET SIZING AND INVESTMENT OPTIMIZATION APP FOR 1ST VERSION

Feature ID	Feature Name	Development status for v1
IUO-01	Receive user input from the Dashboard	Complete. The application already receives the expected input from an internal dashboard (our own version for testing purposes).
CAS-01	Identity RES assets	Complete. The application allows to select which assets to search for.
CAS-02	Retrieve prices from stores	Complete. The application can retrieve prices about solar panels and batteries from different online stores.
AEAS-01	Analyze the capabilities of the assets	Complete. The application can display the capability from each asset.
PRAS-01	Provide report based on the user parameters	Complete. The application can provide reports to the user.

The Asset Sizing and Investment Optimization application is developed using the React framework[4] which allows to build visual elements that can be combined into pages, screens or apps. The backend is developed using Meteor[5], which is a full-stack Javascript platform with a key set of technologies for building connected-client reactive applications. Despite, this application should work as a backend server and interact only with the Dashboard frontend, for development and testing purposes, a simple UI was developed to ensure that all the functionalities developed for the application can be tested and work as expected. This way facilitating all technical tests and validations before the integration work can be done between this App and the DIGITISE Dashboard.

Since this application needs to fetch prices from stores, is important to keep historical search data which can be useful to avoid repeated queries or whenever there is no internet connection to retrieve new updated data. These historical data are stored in a Meteor database and can be visualized by the User in the UI.

Also, as part of the application development, a list of APIs is being developed to allow interactions to and from the App. These APIs will be available by using the Express.js[6] which is a Node.js web application framework that can provide a set of methods to be consumed for any application.

[4] https://react.dev
[5] https://www.meteor.com
[6] https://expressjs.com/

Funded by
the European Union

2.2.3 Assumptions and Constraints

The Asset Sizing and Investment Optimization application relies on the data availability upon each household. These data should include information about the consumption of each household, the capacity to produce energy, the available area for installing any solar solution. The application generates scenarios based on the details of each household and offer possible solutions to optimize its consumption and therefore reduce the costs. But to offer any scenarios, the available data from each household should be available via the Digital Twin Platform, which enables the creation of more accurate solutions for each household. Having this data unavailable or inaccessible through the Digital Twin platform, could lead to more "generic" solutions that would have lower impact in the household.

Another constraint is the lack of available APIs from stores for retrieving prices from RES assets. As a result, the application may have limited capacity to retrieve prices from more stores that expected and therefore offer less solutions to the User. On another aspect, the application may also face challenges when generating complex solutions. These solutions typically include solar panels, inverters, batteries, and the installation itself. Installation costs cannot be defined with a fixed price, as they depend on several factors—the amount of labor needed to adapt the household to the system, ensuring sufficient electrical capacity, the available space for installing all the products, etc. Additionally, proper integration of all connections within the household can further influence the cost. Due to these variables, the 1st version of the application estimates installation costs as a percentage of the total proposed solution.

2.2.4 Development plan for next version

For the next version of the application, all planned functionalities will be implemented, including the integration with the other DIGITISE components, including the DIGITISE Dashboard for retrieving the user-based parameters, and the DIGITISE Digital Twin Platform, to retrieve the necessary household data and push the recommendation reports generated by the App. Also relevant to v2, is the integration with the optimization application, so to be able to use the optimization capabilities of DIGITISE, in order to enhance the recommendations to be provided to the users. All feedback gathered after the release of version v1 will be reviewed and integrated in the next version to better align with the User needs and improve the application functionalities. In Table 4 the list of features of the application for the 2nd version is showcased.

TABLE 4 – FEATURES OF ASSET SIZING AND INVESTMENT OPTIMIZATION APP FOR 2ND VERSION

Feature ID	Feature Name
IUO-01	Receive user input from the Dashboard
IUO-02	Receive household data

CAS-01	Identity RES assets
CAS-02	Retrieve prices from stores
CAS-03	Retrieve RES based on household forecasts
CAS-04	Generate multiple scenarios based on actual household (household + RES)
AEAS-01	Analyze the capabilities of the assets
AEAS-02	Analyze the impact of the asset in the household forecast (and its location)
AEAS-03	Perform optimizations on multiple scenarios
PRAS-01	Provide report based on the user parameters
PRAS-02	Provide multiple reports based on the household forecasts
UFA-01	Provide feedback on reports and request updates

2.3 Health and Security Application

2.3.1 Overview

The scope of the work in Task 4.3 is to design health and security services for building occupants. More specifically, the aim of this task is to deliver a bundle of services whose purpose is twofold: (i) identify and mitigate poor indoor conditions (e.g., air quality, thermal discomfort, lighting quality, pathogen risk factors), and (ii) detect abnormal behaviors or contextual conditions (e.g., intrusion risks, fire, gas, or water leaks, unauthorized access) and enable rapid and reliable control actions. First, a key aspect in the design and implementation of this service is identifying risk conditions related to health and security aspects of households in real time. Additionally, it should feature an alert system that informs the user of potential risks and recommends actions in a personalized manner towards combating unpleasant or even hazardous conditions. In cases where the user has permitted autonomous actions to be carried out, the application should execute these actions directly. Importantly, all the above should be delivered via non-intrusive ways to the users.

The Health and Security application has three main components. The user interacts with the application through an Interface that provides historical and real-time views for a variety of indoor and environmental sensor data. The outcomes of the app are made visible to the users to view alerts, receive smart recommendations, and review automation actions taken by the system. The AI Profiling and Control Engine is the most integral part of the application, as it serves as its intelligence core. It analyses user behaviour and household usage patterns, and occupancy trends. The engine also makes decisions to issue contextual recommendations aimed at improving both comfort and safety. Last, the engine takes as input data from the Health and Security Data Layer. This essentially collects and manages data from a range of sources throughout the home (e.g., consumption and environmental sensor data).

During the 1st release, the Health and Security application realizes a subset of the functionalities expressed in D2.2. First, it supports streaming device-generated data in the form of time series and can account for user-provided input. Additionally, based on the occupancy and comfort settings the application extracts patterns that enable the system to make informed decisions on how to adjust the environment for maximum comfort and health benefits. The AI engine detects deviations from user-specified comfort levels and outputs recommendations to adjust the living conditions accordingly. This feature ensures that any potential health or security risks are quickly flagged for user attention. Additionally, the application exposes API endpoints for its outputs; allowing for other components/applications to utilize them for other functionalities.

TABLE 5 – FEATURES OF HEALTH AND SECURITY APP FOR 1ST VERSION

Feature ID	Feature Name	Development status for v1
HSI - 01	Real-time IAQ and device data monitoring via the interface	Partially complete. Real-time and SRI calculation is missing
HSD-01	Integration with indoor sensors and sub-metering streams	Partially complete. Missing connection to the Data Space
AIP - 01	Behavioural analytics using AI	Partially complete. Pattern recognition is missing
AIP - 02	Anomaly detection and alert triggering based on deviations from norm	Partially complete. Occupancy deviations missing
AAM - 03	Fallback recommendation system if automation is not applicable	Partially complete. Recommendations for temperature setpoints and ventilation setting are implemented

2.3.1 Delivered Functionality

2.3.1.1 Technology Stack

For the initial version, a standalone application for the Health and Security application has been developed. This comprises of both front and back-end components, utilizes an internal database for the training and evaluation of the engine components, and lies atop a specific architecture. Below, an overview is provided for the respective components, and a more elaborate documentation of the application details is offloaded in Annex I.

Funded by the European Union

Frontend: HTML has been used to showcase how the outputs of the application will be shown to the user. All this information will later be visualized through the dashboard. Figure 7 depicts the user interface features that are supported in the 1st version.

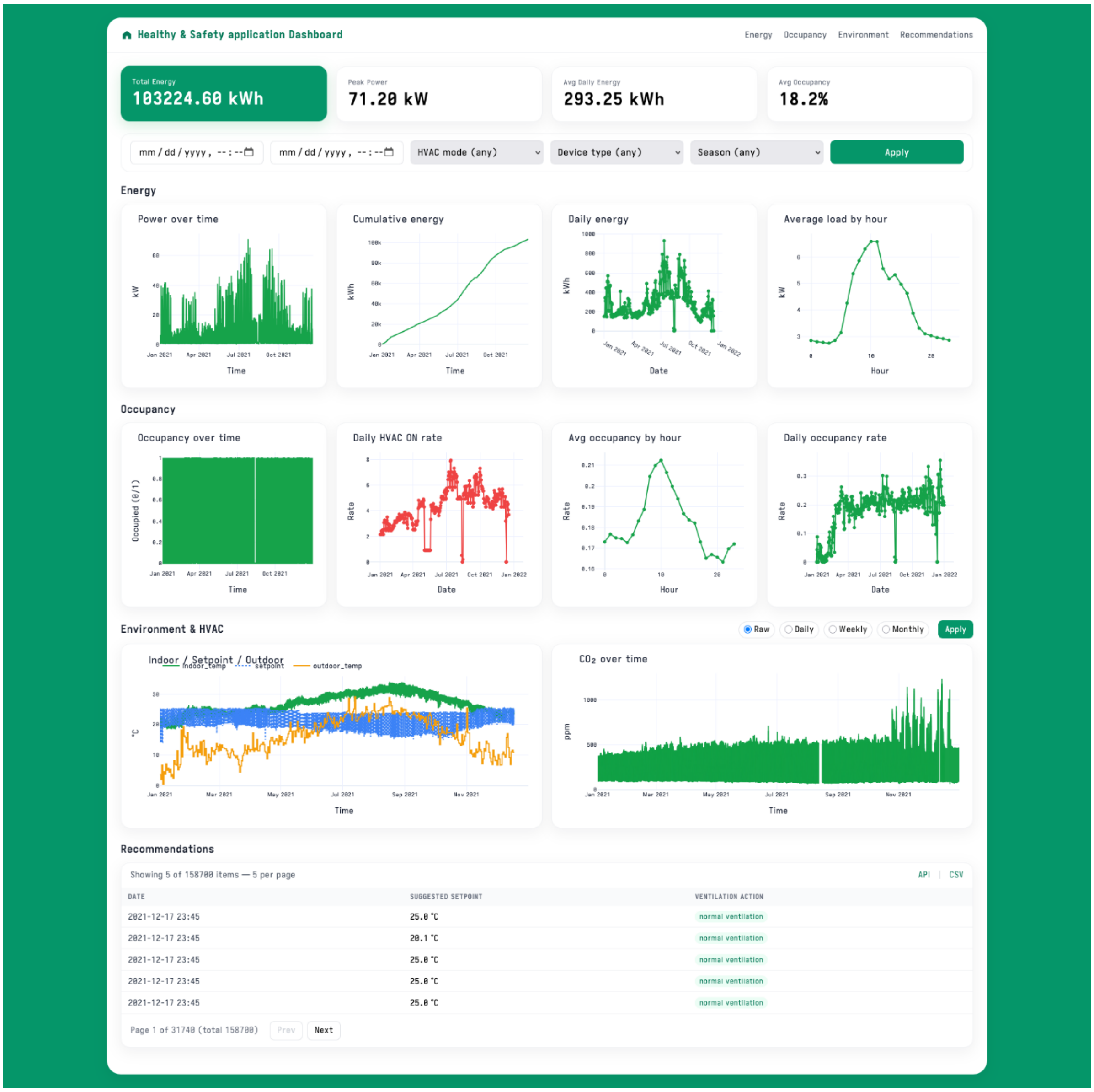

FIGURE 12. USER INTERFACE FOR THE HEALTH AND SECURITY APPLICATION

Backend: All logical components of the application were developed in Python and the connectivity with other DIGITISE components will be facilitated via REST API endpoints. In terms of the ML techniques employed in the implementation, XGBoost and RandomForest were tested with various indicative results.

Database: To realize the application, the PLEIADA dataset was utilized, which contains various measurements from different devices inside three building blocks (A, B, and C). This data is in the form of timeseries; it is ingested via Apache Airflow, preprocessed with PySpark, and both the input and output are managed via PostgreSQL.

Architecture: As depicted in Figure 8, the application currently follows a microservices architecture for modularity and scalability purposes. Each microservice is developed as an individual container, and orchestration is performed through Kubernetes.

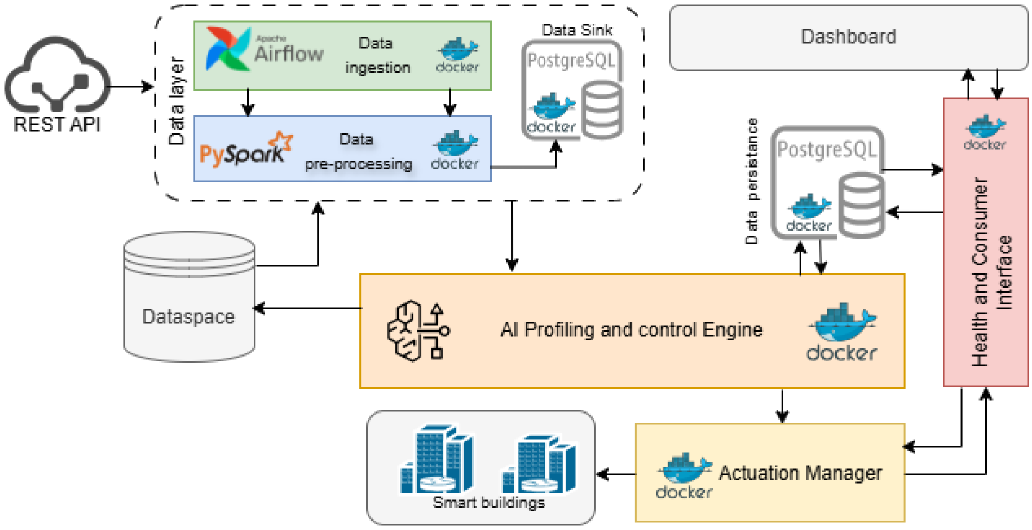

FIGURE 13. ARCHITECTURAL VIEW OF THE HEALTH AND SECURITY APPLICATION

2.3.2 Assumptions and Constraints

The first version (v1) of the application has been developed with the following assumptions and constraints:

1) **Connectivity:** Integration with the Data Space and the Dashboard is missing currently.
2) **Evaluation:** The application has been tested for scalability and performance based on its supporting initial-version operations on for data sourced from an existing dataset. Further testing is needed upon acquiring access to real-time data.
3) **Functionality:** The engine of the application provides the user with essential functionality in terms of monitoring and outputs initial recommendations.

2.3.3 Development plan for next version

For the application to be fully functional and connected to the DIGITISE ecosystem, the components need to be fully developed, enabling the realization of the remaining features as well. As specified in D2.2, first, the Interface needs to be able to support personalized recommendations for improving health/security of the household residents. Second, the automated control component needs to be developed, allowing for the adjustment of environmental factors like ventilation or lighting, responding to real-time conditions to maintain a healthy, secure environment. Notably, this can be done automatically with the consent of the user, or via the application manually, e.g., after receiving a respective recommendation. Third, the Health and Security data layer needs to be finalized to output an enriched SRI score by incorporating data related to IAQ and security indicators and support data exchange with external services.

For technological aspects of the implementation, the following observation can be made: while both RandomForest and XGBoost performed well, their differing strengths suggest potential for further exploration. More models will be explored, and the multi-objective controller model for user comfort will be defined. These predictions can be used as input for comfort-level modeling and assist in generating personalized suggestions to improve user comfort. Below, the full list of features to be supported is listed (as per D2.2).

TABLE 6 – FEATURES OF HEALTH AND SECURITY APP FOR 2ND VERSION

Feature ID	Feature Name
HSI – 01	Real-time IAQ and device data monitoring via the interface
HSI-02	Provision of personalized recommendations for improving health/security
HSD-01	Integration with indoor sensors and sub-metering streams
HSD-02	Calculation of enhanced SRI score based on IAQ and security indicators
HSD-03	Data exchange with external services
AIP – 01	Behavioural analytics using AI
AIP – 02	Anomaly detection and alert triggering based on deviations from norm
AAM – 01	Automated control of environmental conditions
AAM – 02	Smart actuation commands based on AI-driven decisions or IAQ thresholds
AAM – 03	Fallback recommendation system if automation is not applicable

2.4 Smart Home and Distributed Energy Resources (DERs) Automated Control

2.4.1 Overview

As outlined in D2.2 this automation-driven system enables consumer participation in energy activities without requiring any direct user involvement. The system's base functionality is to convert strategic energy management plans into specific automated device instructions, resulting in both energy savings and financial advantages. The system operates through Automated Device-Level Control Execution (ADCE), enabling target applications to access devices directly without human involvement. Each device is operated at its best performance level since it receives continuous top-level commands from DIGITISE applications, which ultimately aim to reduce energy waste, operational expenses, and keep users comfortable.

Funded by
the European Union

The system performs automated tasks which eliminate consumer responsibilities while making energy-saving programs user-friendly. The system provides effective energy-saving solutions which operate in a way that fits naturally into daily routines to build user confidence in automated energy management systems. Components and Integration Features. The application operates through two essential parts, which include the Smart DER Optimizer Component and the Automated Device-Level Control Execution (ADCE). These components enable detailed interactions with other system modules within DIGITISE.

The **Smart DER Optimizer Component** effectively serves as an interface, translating high-level energy strategies into commands for household devices. It integrates with the:

1. **VPP Configuration Component:** The Smart DER Optimizer Component receives disaggregated flexibility requests from the VPP Configuration Component (which processes contract information, actual flexibility data, and grid requests). This enables the optimizer to initiate specific device-level demand response actions.
2. **Energy Savings and Self-consumption Optimization Application:** Insights from this application guide the Smart DER Optimizer Component in translating energy efficiency strategies into precise control signals for individual devices, promoting optimal energy usage and increased self-consumption.

The **Automated Device-Level Control Execution (ADCE)** component facilitates seamless communication and control between end-user devices and the broader energy management system. It acts as the gateway for routing control actions to the physical assets. The ADCE interacts with:

1. **Smart DER Optimizer Component:** Receives translated energy strategies and demand response commands for execution.
2. **Automation Deployment Component (from the Health and Security Application):** Receives direct control actions for execution.

2.4.2 Delivered Functionality

The first version (M18) of the Smart Home and DER Automated Control application focuses on establishing the core architecture for strategy translation and device-level command execution. Table 7 summarizes the features and their current development status.

TABLE 7 – FEATURES OF SMART HOME AND (DERS) AUTOMATED CONTROL FOR 1ST VERSION

Feature ID	Feature Name	Development Status for v1
SHC-1	Modeling for Energy Efficiency Strategy Execution	Partially complete. Basic translation logic implemented to convert simple 'save energy' directives into generic device-off commands.
SHC-2	Modeling for Demand Response Strategy Implementation	Partially complete. Prototype converts high-level 'reduce load' requests from the VPP stub into simulated device-level actions. Still pending the delivery of the VPP as part of the work in WP5
SHC-3	Handling of Control Events Triggered by Business Applications	Complete – Internal API endpoints for receiving directives from applications have been defined and tested with mock data.
SHC-4	Execution of Control Events	Partially complete – Control actions are logged to a database but not yet routed to physical hardware (pending the demo assets' integration)

2.4.3 Technology stack

The system serves as an integration and execution hub, with its technology stack chosen to support interoperability and real-time communication. Communication between this component and the other DIGITISE applications (VPP, Energy Savings) will be established through a RESTful API built on the HTTP protocol and JSON data exchange. This ensures a lightweight and standardized method for receiving high-level strategies.

The core of the ADCE and Smart DER Optimizer components is being developed in Python, utilizing frameworks suitable for building robust API services (e.g., FastAPI, Flask).

For the execution layer (API Gateway), which must interact with diverse hardware, the architecture is being designed to be protocol-agnostic. In the next versions, it will incorporate connectors or drivers for common smart home and DER protocols (e.g., MQTT, REST adapters) to facilitate communication with physical assets at the pilot sites.

2.4.4 Assumptions and Constraints

This first version of the application has been developed with several practical limitations and starting conditions:

1. **Connectivity:** Development has been conducted internally. The links to the VPP Configuration Component, Energy Savings Application, and Health and Security Application are not yet live.
2. **Hardware Integration:** There is no connectivity to physical devices in v1. The "Execution of Control Events" is simulated. The primary focus has been on defining the data flow and command structure, not on hardware-specific implementation.

Funded by the European Union

3. **Functionality:** The strategy translation logic is partially complete. E.g., a "demand response" event might trigger a generic "turn off" command but more fine-grained control strategies may be defined as the final version of VPP application is ready.

2.4.5 Development plan for next version

In the next version, the focus will be on fully integrating the application within the DIGITISE framework and enabling real-world control capabilities. This includes establishing live API connections, developing sophisticated translation logic, and integrating with physical pilot-site hardware. The full list of features to be reported in the next version are presented in Table 8.

TABLE 8 – FEATURES OF SMART HOME AND (DERS) AUTOMATED CONTROL FOR 2ND VERSION

Feature ID	Feature Name
SHC-1	Modeling for Energy Efficiency Strategy Execution
SHC-2	Modeling for Demand Response Strategy Implementation
SHC-3	Handling of Control Events Triggered by Business Applications
SHC-4	Execution of Control Events

2.5 Consumer Application for Personalized Empowerment and Capacity Building

2.5.1 Overview

The aim of task T4.4 is to develop a unified DIGITISE Ecosystem, the DIGITISE Consumer Dashboard, on which all applications will be integrated. It provides historic and real-time data, AI-driven analytics, energy and non-energy management capabilities and empowers consumers and prosumers. Moreover, it supports sustainability, energy efficiency, and advances active participation in energy markets.

The DIGITISE Consumer Dashboard, can collect and integrate not only real-time and historic data, but aggregates information about consumption and generation from smart meters, EV, CPs, local RES, etc. and enables users to monitor performance, make data-driven decisions and to visualize all the data. Advanced analytics are also supported, like energy demand forecasting, renewable production predictions, and leveraging machine learning models.

The DIGITISE Consumer Dashboard also emphasizes user management and behavioral empowerment. Integrating personalized insights, notifications, and energy tools through the Energy Savings and Self-Consumption Optimization application, recommendations to improve efficiency and better understand their energy behavior. Based on these features, sustainable consumption informed decision-making features are encouraged.

Furthermore, the DIGITISE Consumer Dashboard is integrated with the DIGITISE Health and Security Management application functionalities and provides real-time indoor environment monitoring, such as air quality, humidity, thermal comfort, etc. User-oriented notifications are sent to notify the users about potential health risks, like poor air quality, abnormal energy usage and can view automated or manual recommendations maintaining comfort and safety. Thus, a safer and more comfortable household is a result of the security features there are, such as anomaly detection, device malfunction alerts, and monitoring of unauthorized consumption.

Asset Sizing and Investment Optimization Application is also integrated with the DIGITISE Consumer Dashboard, allowing users to search for various energy investment scenarios. The simulation tools and digital twin modelling can be used to assess the technical and financial impacts of investments in solar PVs, EV infrastructure, or battery storage. ROI, IRR, Payback Period and other key financial indicators are presented and support decision-making and long-term sustainability planning.

Finally, the Flexibility Marketplace is directly integrated with the DIGITISE Consumer Dashboard, to provide and engage in demand responses and flexibility programs. Users can visualize their flexibility potential, enroll in relevant programs, and gain insights into financial opportunities arising from participation in local or regional energy markets.

Overall, the DIGITISE Consumer Dashboard acts as the central control and visualization hub of the DIGITISE ecosystem.

a) In terms of visualization, it delivers interactive and user-friendly graphical representations of consumption data, simulation results, household performance indicators, and scenario outcomes.
b) Regarding configuration, it allows users to define and modify input parameters—such as building profiles, device characteristics, or operational constraints—through intuitive forms and dynamic fields, ensuring smooth preparation for simulations or other DIGITISE functionalities.

2.5.2 Delivered Functionality

In the first version (v1) of the DIGITISE Consumer Dashboard, the first version of the applications will be integrated, providing and supporting communication with the DIGITISE Data Space and visualization of selected KPIs. Users can log in through the web interface, access their household information and data, display energy consumption KPIs, and receive alerts and notifications. These include insights from Energy Savings and Self-

Funded by
the European Union

consumption Optimization, Health and Security Management, Behavioral Profiling, Asset Sizing, and Flexibility Marketplace applications.

Real-time updates, AI-derived insights are presented through widgets, charts, and personalized summary indicators. This provides a comprehensive overview of the household's energy and comfort performance while maintaining the system's modularity and scalability.

TABLE 9 – FEATURES OF CONSUMER APP FOR PERSONALIZED EMPOWERMENT AND CAPACITY BUILDING FOR 1ST VERSION

Feature ID	Feature Name	Description	Development status for v1
APP-01	Integration with DIGITISE Applications	Integration with Energy Savings, Self Consumption Optimization, Health and Security, Behavioral Profiling, Asset Sizing, and Flexibility Marketplace applications.	In progress
APP-02	Data Management (Real-time and historic data retrieval and processing)	Aggregate and process real-time and historical data from devices, smart meters, and electric vehicles for energy consumption and generation trend analysis.	In progress
APP-04	User-Friendly Visualization and Control (User Portal / Dashboard UI)	Interactive dashboards with dynamic graphs, widgets, and real-time alerts to visualize energy usage, savings opportunities, and performance insights.	In progress
APP-05	Personalized Recommendations and Comparisons	Deliver tailored insights comparing user energy behavior with similar households and forecasting future energy needs. Include behavioral profiling visibility.	In progress
APP-06	Notifications	Offer browser- and user-oriented notification center	In progress

2.5.2.1 Technology Stack

The DIGITISE Consumer Dashboard is developed using a modular and scalable web-based architecture ensuring interoperability with all DIGITISE applications and the central Data Space. The selected technologies enable historic and real-time data handling and support flexibility, responsiveness (laptop, tablet, mobile) and secure data handling.

Funded by
the European Union

Frontend: For the frontend of the DIGITISE Consumer Dashboard, Angular technology was used. Angular is a modern TypeScript framework designed to build dynamic, responsive, and component-drive applications. It provides also efficient data binding and strong modularization to enable smooth user interaction and real-time updates. The graphs and widgets are created using Angular's libraries and open-source components, ensuring intuitive and consistent user interface across different devices.

Backend: The backend functions are built using the Spring framework, applying Spring framework component for the RESTful APIs implementation that manages the communication between the frontend and the DIGITISE Applications - Data Space. Moreover, WebSocket technology is laboring to support real-time updated and event-driven interactions. Thus, bidirectional communication between client and server is allowed. Additionally, the DIGITISE Consumer dashboard integrates Spring WebFlux, ensuring non-blocking, asynchronous data processing and scalability under high-load conditions. Maven is used for dependency management, build automation, and version control, supporting continuous integration of workflows across the DIGITISE ecosystem.

Rest API Security: In order to protect the RESR APIs, the DIGITISE Consumer Dashboard uses Spring Security by implementing a Jason Web Token (JWT) based authentication. Different role-based access controls are defined, to ensure only authenticated and authorized users can access the DIGITISE ecosystem. By leveraging Spring Security's filters and configuration capabilities, the system provides a robust, customizable security layer while remaining stateless and scalable.

Data Management and Integration: For database operations, the Spring Data MongoDB is used to allow efficient management of semi-structured and time-series data, such as energy consumption, environmental measurements, and behavioral metrics. The data is securely pulled from the DIGITISE Data Space, processed and persisted to support both historic and real-time visualizations.

Testing and Quality Assurance: For unit and integration testing, Mockito is used to ensure reliability, consistency, robustness of multiple individual modules and API endpoints. Full testing will be performed in all versions including end-to-end validation across all DIGITISE applications.

Overall, the selected technologies for the DIGITISE Consumer Dashboard were selected to provide a solid and flexible foundation. Based on these, high performance, scalability, and secure communication are ensured by supporting the delivery of an interactive and user-centric experience aligned with the objectives of the DIGITISE ecosystem.

Funded by
the European Union

2.5.2.2 User Interface screenshots and detailed explanations

To provide the reader with a better understanding of the DIGITISE Consumer Dashboard, below mockups of different functionalities are provided, detailing the functionalities currently supported. Importantly, based on the output of the corresponding DIGITISE applications that will be integrated with the DIGITISE Consumer Dashboard, the latter will support browser and user-oriented notifications. Such notifications will be available to users after logging in.

To log in, users input their credentials and gain access to visualized information pertaining not only to their building-household as a whole, but also to individual devices and sensors. Currently, a one-to-one correlation has already been implemented with a demonstration partner and a building in which there are multiple and different devices and sensors. Figure 14 depicts the initial screen observed by a demonstration partner, after signing in.

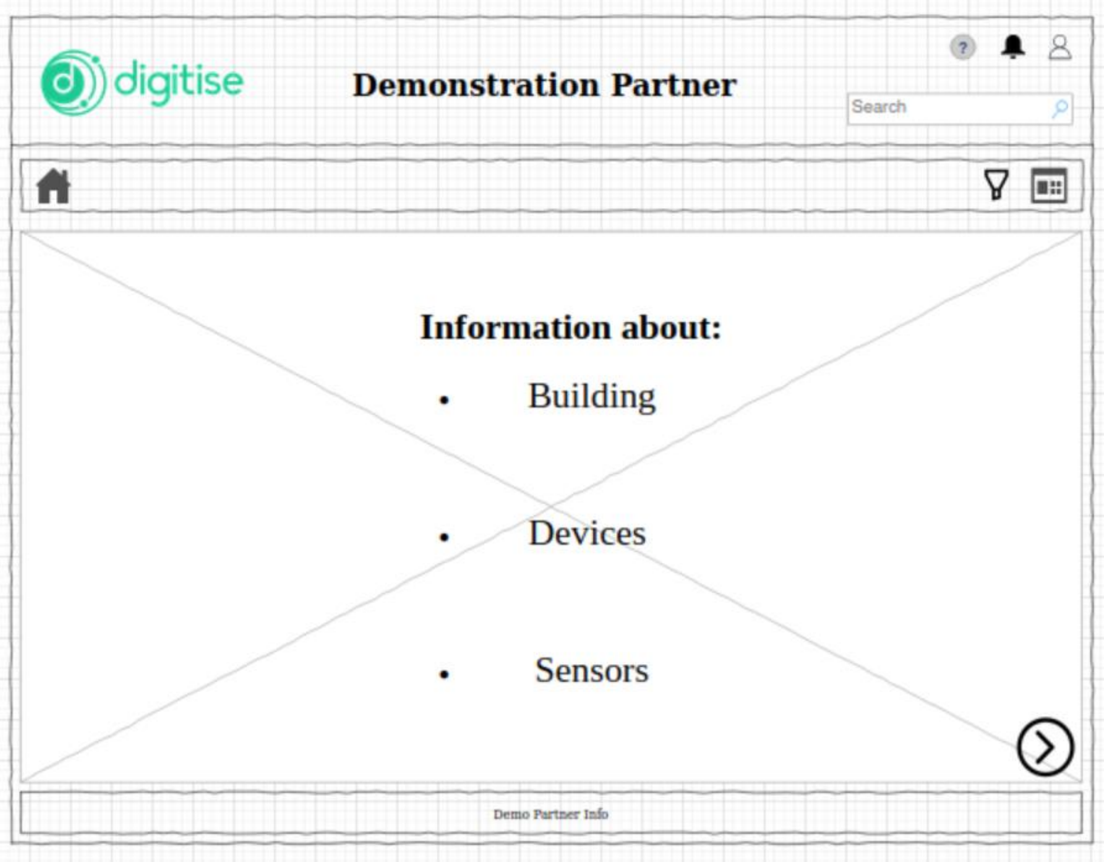

FIGURE 14. HOME PAGE OF THE DIGITISE CONSUMER DASHBOARD DISPLAYING BASIC INFORMATION ABOUT THE BUILDING, DEVICES AND SENSORS OF THE USER (AFTER SIGNING IN).

Now, by using the arrow button on the bottom right part of the tab, the user will be redirected to the next page. As Figure 15 showcases, the user now has access to the set of DIGITISE applications that have been integrated with the DIGITISE Consumer Dashboard (left column) and is able to select real-time or historic data to visualize from their devices and sensors (main window).

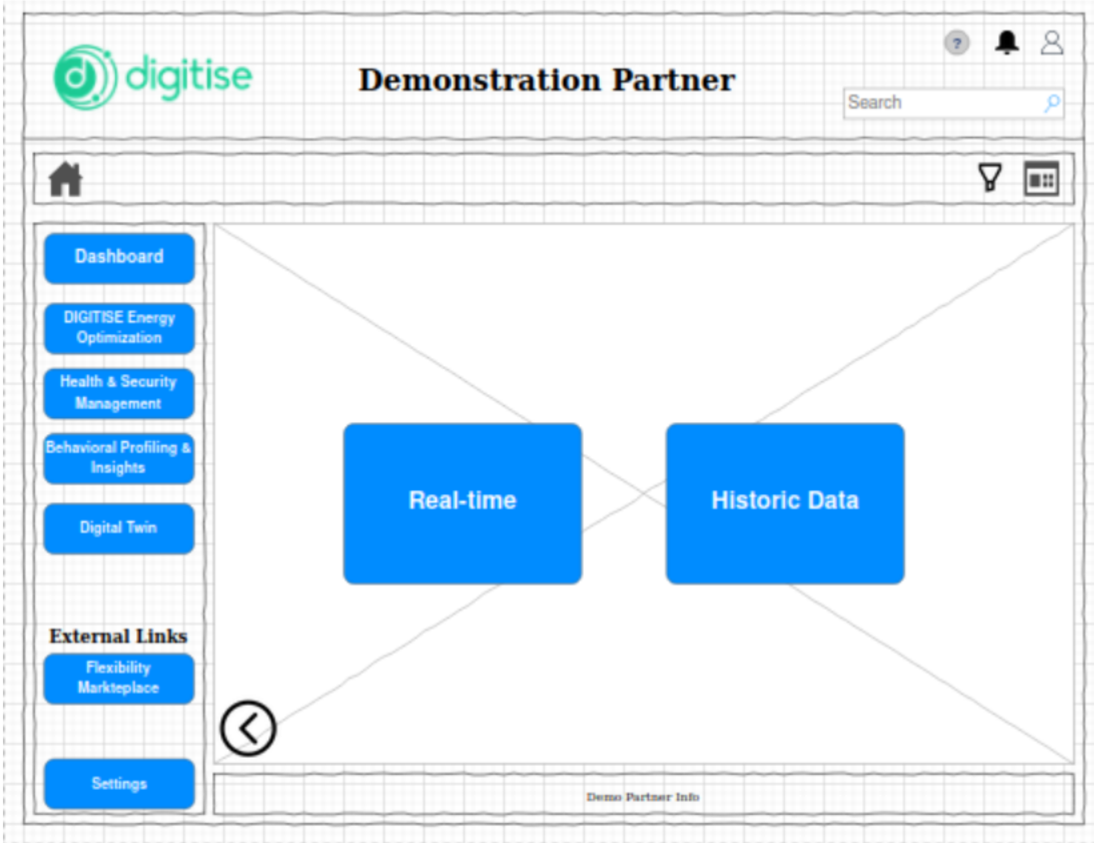

FIGURE 15. THE SECOND PAGE OF THE DIGITISE CONSUMER DASHBOARD, IN WHICH THE USER WOULD BE ABLE TO SELECT AND DISPLAY REAL–TIME OR HISTORIC DATA.

By selecting the "real-time" option, the user can see real-time data visualization for their devices and sensors inside the building-household. Depending on the KPIs, the user will be able to display time series graphs, single point measurement, as well as line and bar charts, depending also on the constraints of the available granularity of the measurements. Figure 16, Figure 17, and Figure 13 provide example views for such visualizations.

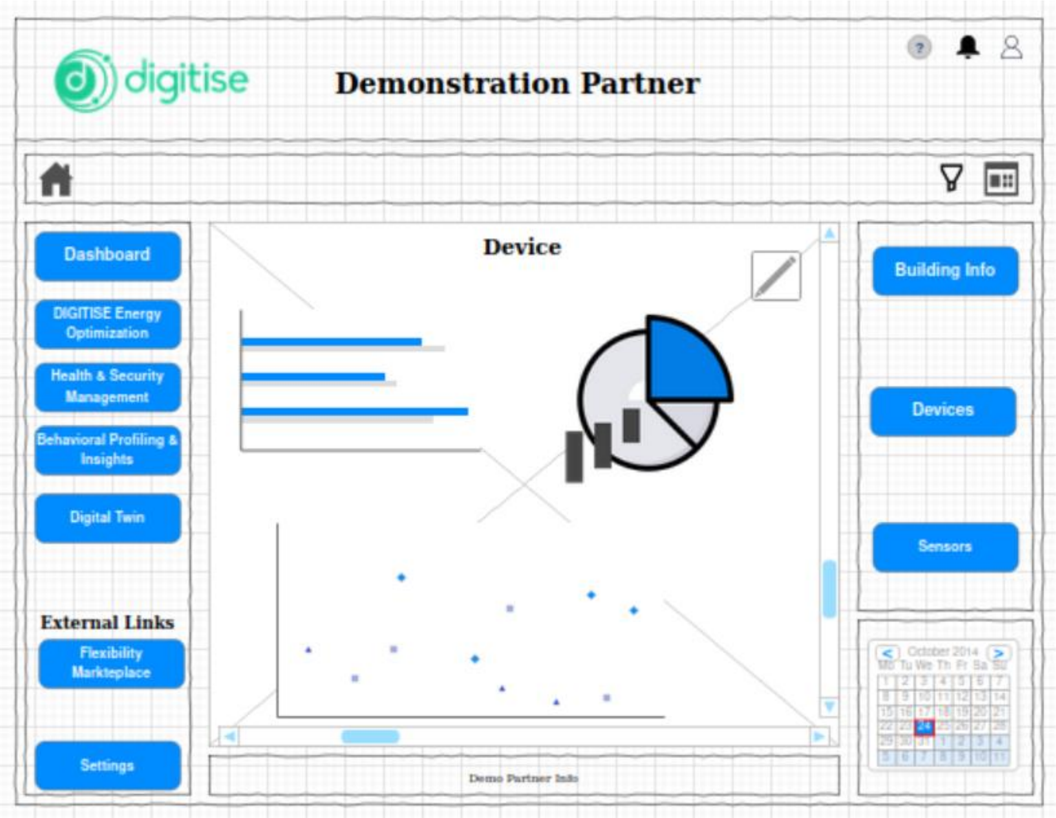

FIGURE 16 – THE VISUALIZATION OF THE REAL-TIME DATA FOR A SPECIFIC USER.

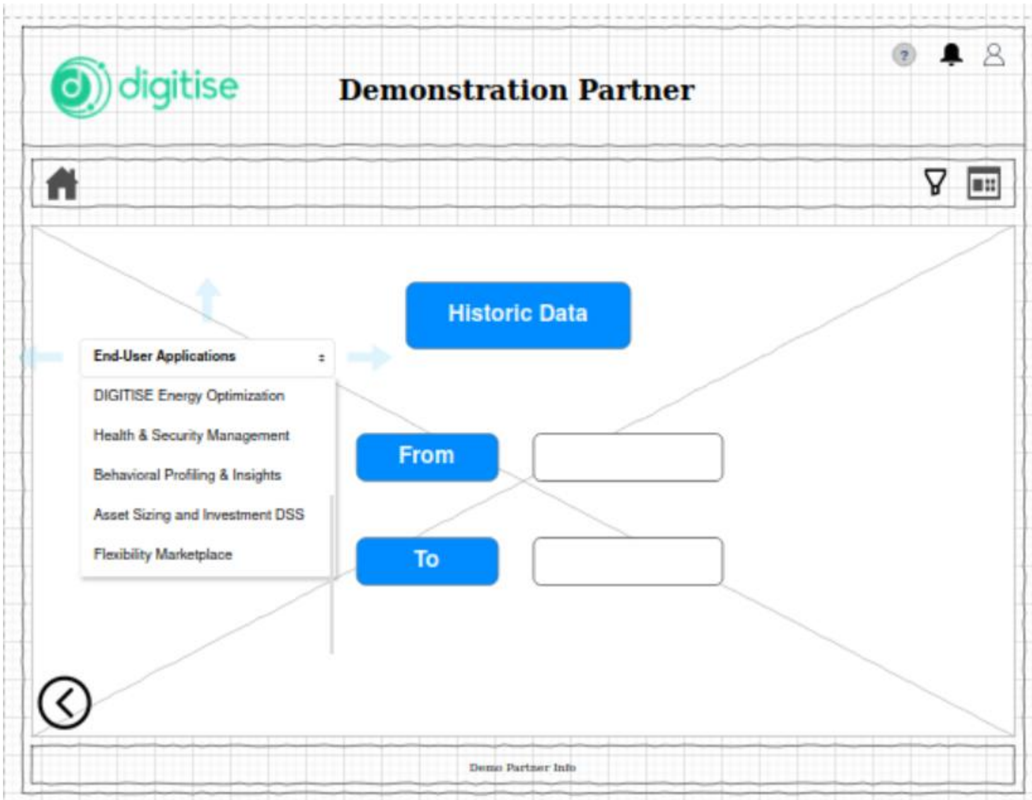

FIGURE 17 – THE USER COULD SELECT TO VISUALIZE HISTORIC DATA BY SELECTING APPLICATION FROM A DATE RANGE (FROM-TO).

Funded by
the European Union

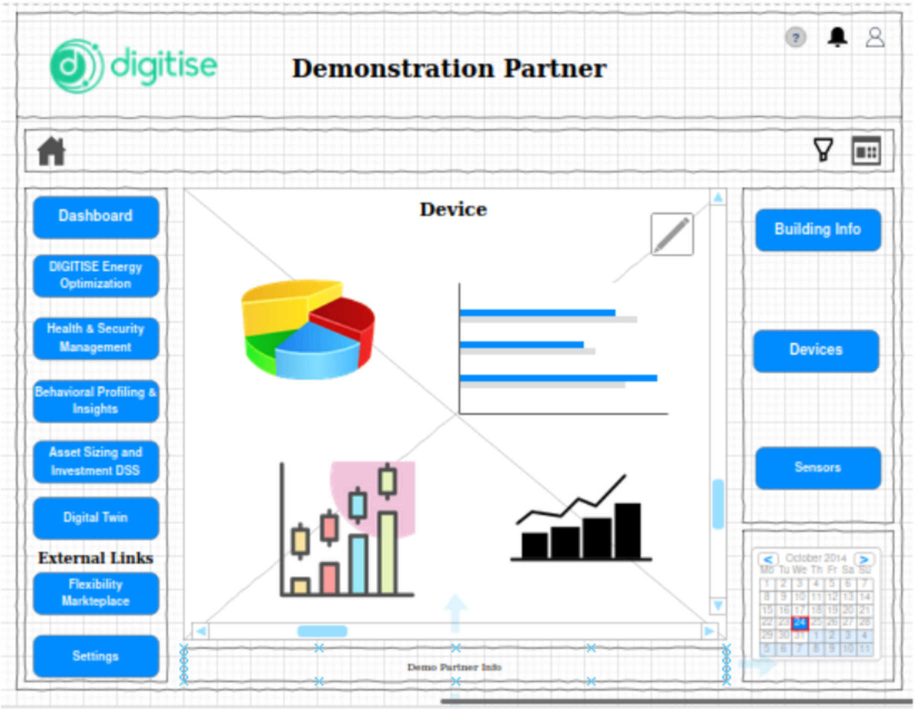

FIGURE 18 – HISTORIC DATA VISUALIZATION PER USER, MEANING PER BUILDING, DEVICES AND SENSORS.

Last, an example of data visualization from individual applications is provided. In this case, once users select the Health and Security Management application from the left menu, the image depicted in Figure 19 is presented. In general, upon integration between the DIGITISE applications and the Dashboard gets finalized, the corresponding application will retrieve the relevant data from the DIGITISE Data Space, will produce its outputs which will then be forwarded to the DIGITISE Consumer Dashboard for visualization purposes.

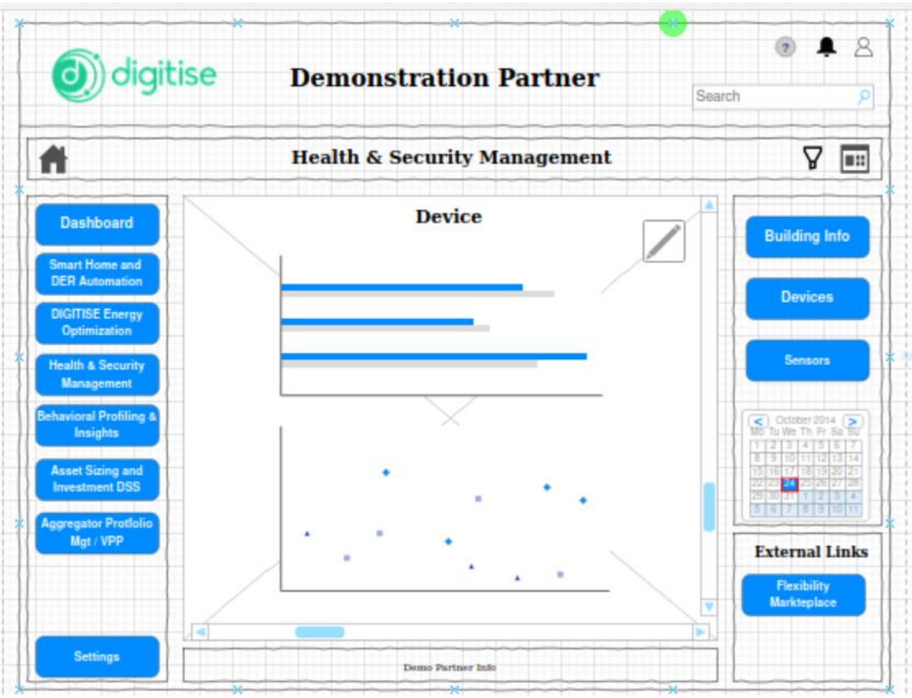

FIGURE 19 – AN EXAMPLE, BASED ON HEALTH AND SECURITY MANAGEMENT APPLICATION, OF DATA VISUALIZATION RETRIEVED AND ELABORATED FROM THE DIGITISE APPLICATION.

2.5.3 Assumptions and Constraints

Below are presented the assumptions and constraints of the first version of the DIGITISE Consumer Dashboard:

- **Integration Dependencies:** Features and functionalities rely on API endpoints from the DIGITISE applications and Data Space. In v2, will be completed the full integration once these API endpoints will be finalized.
- **Data Availability:** In v1 only synthetic data were used for preliminary testing. The real-time date will be pulled and used in v2.
- **Visualization Scope:** In v1 dashboard mockups establish key visualizations features on each application. The interactive analytics and graphs will be added and used in v2.
- **User Interface:** An initial version of UI was built for the v1 of the DIGITISE Consumer Dashboard. Enhanced features will be introduced in v2 after the demonstration and technical partners' feedback.
- **Performance Testing:** Tests, like scalability, etc., are pending and will be conducted when the integration with the final version of applications of the DIGITISE ecosystem is done.

2.5.4 Development plan for next version

The second version (v2) of the DIGITISE Consumer Dashboard will include full integration with all DIGITISE applications and advanced analytics capabilities as outlined in D2.2. The main objectives for the next iteration are as follows:

- **Full System Integration:** Establish bi-directional data exchange with the DIGITISE Data Space, enabling seamless access to all real-time and historical datasets.
- **Enhanced Visualization:** Develop interactive charts and dashboards displaying energy usage, indoor conditions, KPIs, forecasts, and optimization results. Incorporate filtering, comparison, and scenario simulation capabilities.
- **AI-Driven Personalization:** Integrate behavioral insights and forecasting modules to provide personalized recommendations for energy efficiency, investment planning, and comfort optimization.
- **Cross-Domain Intelligence:** Combine outputs from energy, health, security, and flexibility applications to present holistic household performance indicators.
- **User Engagement and Gamification:** Introduce educational and gamified features to encourage energy literacy, self-consumption, and participation in flexibility markets.
- **Security and Privacy:** Implement secure authentication, user consent management, and GDPR-compliant data handling mechanisms.

- **Final Technical Integration and Testing:** Conduct performance, usability, and integration of tests across all modules before deployment in demonstration sites.

Through these advancements, the DIGITISE Consumer Dashboard will evolve into a robust, user-centric platform delivering comprehensive monitoring, intelligent control, and active participation in the digital energy transition.

TABLE 10 – FEATURES OF CONSUMER APP FOR PERSONALIZED EMPOWERMENT AND CAPACITY BUILDING FOR 2ND VERSION

Feature ID	Feature Name
APP-01	Integration with DIGITISE Applications
APP-02	Data Management (Real-time and historic data retrieval and processing)
APP-03	User-Friendly Visualization and Control (User Portal / Dashboard UI)
APP-04	AI-Driven Energy Forecasting Analytics
APP-05	Personalized Recommendations and Comparisons
APP-06	Automated Energy Optimization and Insights
APP-07	Personalized Non-Energy Data and Tailored Outputs
APP-08	Asset Sizing and Investment Insights
APP-09	Flexibility and Demand Response Features
APP-10	Notifications

Funded by
the European Union

Conclusions

This document shows the current state of the suite of end-user applications that DIGITISE offers aims to increase the wellbeing and energy literacy at a household level. As far as we can tell, after the work carried out for this work package, each described version of these human-centric services for prosumers offers essential functionality and lays the ground for the next and final, enhanced, versions of the applications. The exploration of various technological components alongside a fusion of techniques has been utilized at a ``standalone'' level to develop the first version of the applications, in accordance to previously provided information i.e., in D2.2. Apart from the core functionalities that are currently supported, constraints have been identified, and a development plan has been proposed. In the following months of the project, the remaining features will be added to the supported functionality of the applications.

Funded by
the European Union

References

[1] Y. Dong, Z. Dong, T. Zhao, and Z. Ding, "A Strategic Day-ahead bidding strategy and operation for battery energy storage system by reinforcement learning," *Electric Power Systems Research*, vol. 196, p. 107229, Jul. 2021, doi: 10.1016/j.epsr.2021.107229.

[2] J. Cardo-Miota, H. Beltran, E. Pérez, S. Khadem, and M. Bahloul, "Deep reinforcement learning-based strategy for maximizing returns from renewable energy and energy storage systems in multi-electricity markets," *Appl Energy*, vol. 388, p. 125561, Jun. 2025, doi: 10.1016/j.apenergy.2025.125561.

[3] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, Aug. 2014, doi: 10.1109/TEVC.2013.2281535.

[4] L. M. Naranjo-Mora, A. C. Ochoa-Siavichay, L. F. Guerrero-Vásquez, J. P. Bermeo-Moyano, J. O. Ordoñez-Ordoñez, and P. A. Chasi-Pesantez, "Marine Predator Algorithm: A Nature-Inspired Metaheuristic Algorithm Applied in Rectangular Patch Antennas Optimization," 2024, pp. 308–316. doi: 10.1007/978-3-031-52090-7_29.

[5] J. Cadena-Albuja, C. Barrera-Singaña, H. Arcos, and J. Muñoz, "Economic Dispatch in Electrical Systems with Hybrid Generation Using the Differential Evolution Algorithm: A Comparative Analysis with Other Optimization Techniques Under Energy Limitation Scenarios," *Energies (Basel)*, vol. 18, no. 13, p. 3414, Jun. 2025, doi: 10.3390/en18133414.

[6] L. Argiolas, M. Stecca, L. M. Ramirez-Elizondo, T. B. Soeiro, and P. Bauer, "Optimal Battery Energy Storage Dispatch in Energy and Frequency Regulation Markets While Peak Shaving an EV Fast Charging Station," *IEEE Open Access Journal of Power and Energy*, vol. 9, pp. 374–385, 2022, doi: 10.1109/OAJPE.2022.3198553.

[7] C. Zhou *et al.*, "Two-stage robust optimization for space heating loads of buildings in integrated community energy systems," *Appl Energy*, vol. 331, p. 120451, Feb. 2023, doi: 10.1016/j.apenergy.2022.120451.

[8] M. S. H. Nizami, M. J. Hossain, B. M. R. Amin, and E. Fernandez, "A residential energy management system with bi-level optimization-based bidding strategy for day-ahead bi-directional electricity trading," *Appl Energy*, vol. 261, p. 114322, Mar. 2020, doi: 10.1016/j.apenergy.2019.114322.

[9] A. Esmaeel Nezhad, P. H. J. Nardelli, S. Sahoo, and F. Ghanavati, "Scheduling of Energy Hub Resources Using Robust Chance-Constrained Optimization," *IEEE Access*, vol. 10, pp. 129738–129753, 2022, doi: 10.1109/ACCESS.2022.3228388.

[10] A. Sorour, M. Fazeli, M. Monfared, A. A. Fahmy, J. R. Searle, and R. P. Lewis, "MILP Optimized Management of Domestic PV-Battery Using Two Days-Ahead

Forecasts," *IEEE Access*, vol. 10, pp. 29357–29366, 2022, doi: 10.1109/ACCESS.2022.3158303.

Funded by
the European Union

Annex I

State of art on EMS optimization algorithms

Given the fact that the key focus of the Energy Savings and Self Consumption Optimization Application is the optimization engine, an in-depth literature review was carried out, assessing potential optimization methodologies. According to the open literature, indeed several optimization methodologies can be applied to short-term energy management, each offering different advantages. Common optimization methodologies include, Reinforcement Learning (RL), heuristic and metaheuristic methodologies, and deterministic optimization methodologies. RL has gained attention for its ability to adapt to uncertainty and make rapid decisions once trained. By learning control policies from interactions with the environment, RL avoids the need for explicit plant models. However, it typically demands large training datasets or detailed simulation environments, and its actions do not always guarantee strict adherence to operational constraints, an important limitation in safety-critical or highly regulated systems [1], [2].

Heuristic and metaheuristic approaches are other options, excelling in problems with nonlinear objectives, multiple conflicting goals, or black-box components. Techniques such as evolutionary algorithms or swarm-based methods can efficiently explore complex solution spaces and are particularly useful for multi-objective trade-off analysis. Despite their flexibility, these methods often lack guarantees of optimality and can be computationally expensive when applied to real-time scheduling [3][4][5].

Given these considerations, the characteristics of the pilots and availability of data, deterministic optimization emerges as a more reliable and structured alternative [6], [7]. In more specific terms, we deploy Mixed-Integer Linear Programming (MILP), which combines discrete decision-making with linear formulations, enabling transparent, reproducible, and optimally solvable models [8], [9]. MILP offers a strong balance between modelling capability and computational efficiency, supported by robust commercial and open-source solvers. While linear approximations may simplify some nonlinear dynamics, convex representations typically provide sufficient accuracy for operational use. More advanced deterministic methods like MINLP can capture nonlinearities more precisely but are often too computationally demanding for real-time applications [10]. For this deployment, MILP was chosen as the primary optimization approach due to its maturity, interpretability, and ability to deliver high-quality solutions within operational time frames.

Details on developing Health and Security Application

To realize the application the PLEIADA dataset was utilized, which is a suitable candidate and is publicly available. This dataset contains various measurements from different devices inside three building blocks (A, B, and C). For this 1st version only data from Block

A was considered in the development of the AI engine and the integration with the rest of the components. Consumption data was considered, and data preprocessing was needed and performed. Since most of the data had outliers and missing points, these were filled using a forward-fill method. The cleaned data was then resampled into hourly averages. The HVAC measurements include user temperature setpoints, the operating mode of the system, and on/off status of devices. Similar resampling and cleaning were performed. Additional features were created, such as the time of day (morning, afternoon, and night) and the season of the year (summer, autumn, and winter). The other type of measurements used for this version of the application is the CO_2 levels (measured in ppm). All three measurement types were resampled to one-hour intervals and merged into a single dataset for modelling.

The implementation of the application follows a two-layer pipeline architecture designed to integrate forecasting and control in its design. The first layer consists of a multi-output forecaster for indoor environmental-related predictions. The second layer implements a multi-objective controller that takes the previously-generated forecasts as input and determines actions based on user-predefined health, comfort, and energy-related criteria. Via combining these two layers together, users can opt for a proactive rather than a reactive management approach for their households. To train the forecasting layer, the processed dataset above was used. However, to capture temporal dependencies, the dataset was enriched with lagged variables such as previous indoor temperature (indoor_temp_lag1), previous CO_2 concentration (co2_lag1), previous setpoint values (setpoint_lag1), and HVAC operational state (hvac_on_lag1). Additionally, averages over three-hour windows were generated for short-term rolling to smooth out local fluctuations. Notably, this does not affect the pipeline's ability to extract meaningful patterns from the data. Categorical context features, including hour of the day, season, operational mode, and device type, are also encoded to capture cyclical and situational effects. Thus, the forecasting model can learn complex relationships between system dynamics and environmental conditions.

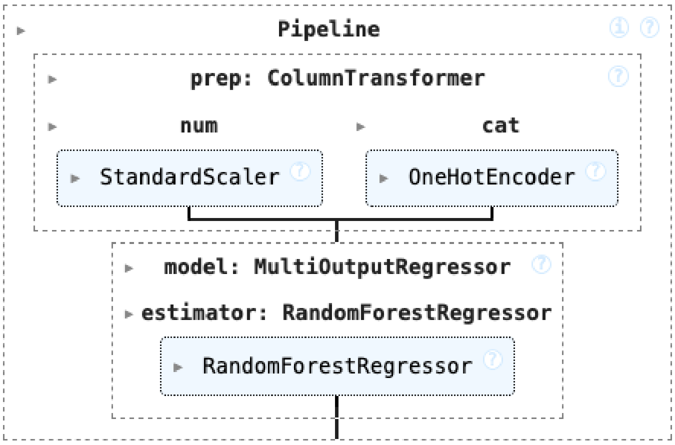

FIGURE 20. THE TWO-LAYER PIPELINE ARCHITECTURE OF THE HEALTH AND SECURITY APPLICATION

Funded by
the European Union

For feature selection, Pearson correlation analysis was employed to reduce redundancy and highlight the most predictive variables. Once the feature set was finalized, the forecasting task was addressed using a multi-output regression strategy. Specifically, XGBoost was embedded within a MultiOutputRegressor framework to jointly predict next-hour indoor temperature and CO_2 concentration. Training was conducted with an 80/20 time-ordered split to preserve the temporal structure of the data, while preprocessing steps included numeric standardization and one-hot encoding of categorical variables. Figure 11 and Figure 12 provide two examples of feature correlations.

Model performance was evaluated using mean absolute error and root mean square error, which capture average deviation and penalize large errors, respectively. For benchmarking purposes, a RandomForest-based multi-output model was also trained and tested. Both models demonstrated strong predictive capabilities, with XGBoost exhibiting slightly superior performance for CO_2 forecasting, while RandomForest showed marginally better accuracy for indoor temperature predictions. The results showcase that both approaches are suitable for short-term forecasting for comfort and air quality. Utilizing a combination of the two in future iterations could yield potential benefits in the process.

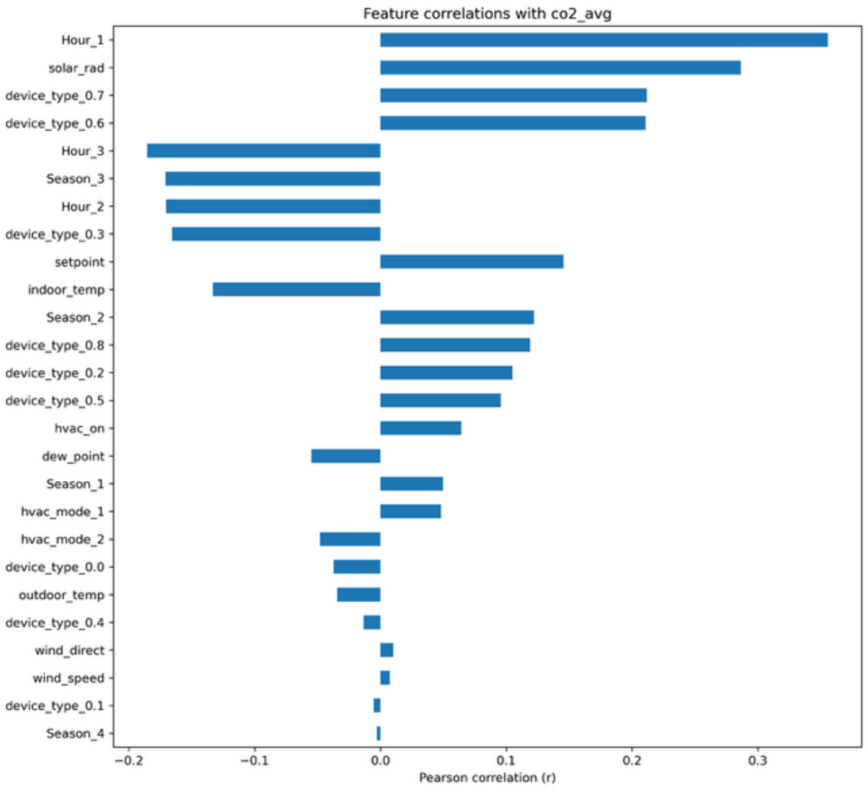

FIGURE 21. FEATURE CORRELATION OF DIFFERENT ATTRIBUTES FOR CO_2

Funded by
the European Union

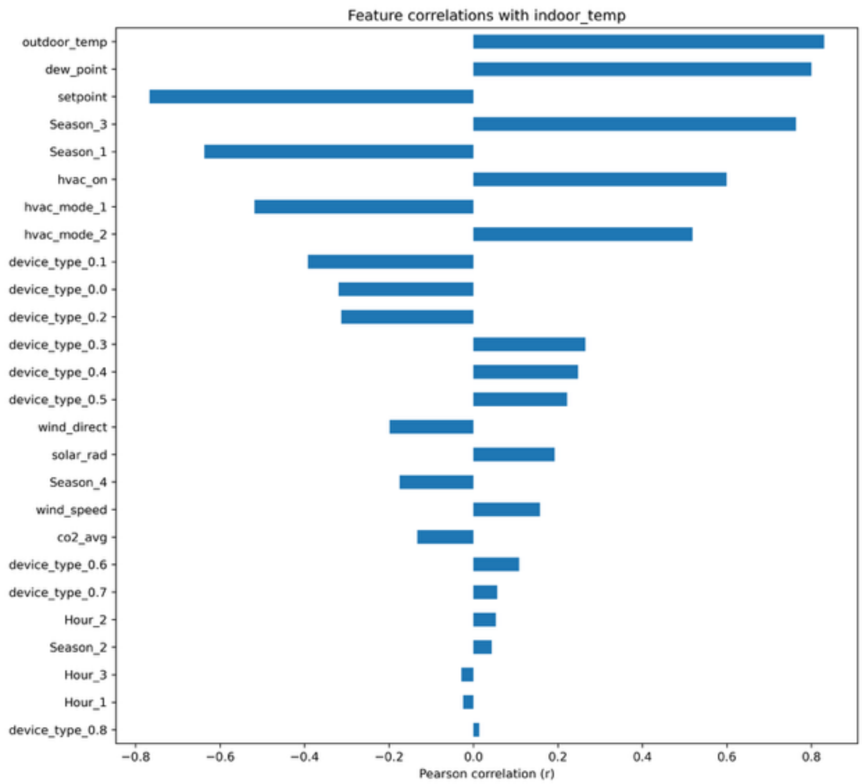

FIGURE 22. FEATURE CORRELATION OF DIFFERENT ATTRIBUTES FOR INDOOR TEMPERATURE

The deployment loop of the system integrates this forecasting functionality with the control layer (see Figure 13). Predictions of indoor temperature and CO_2 concentrations are generated for the next hour, adjusted under hypothetical control actions, and passed to the controller. The controller evaluates these outcomes against predefined objectives and selects the action that minimizes the overall cost. This loop enables continuous, adaptive operation, assisting the household environment to remain within healthy and comfortable ranges.

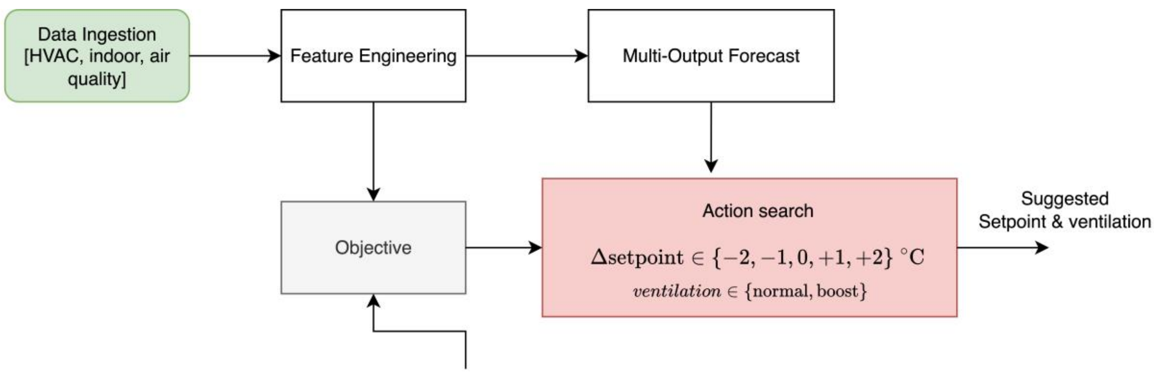

FIGURE 23. THE DEPLOYMENT LOOP OF THE HEALTH AND SECURITY APPLICATION

Funded by the European Union

The controller is designed to minimize a weighted cost function that captures three critical dimensions: thermal comfort, indoor air quality (IAQ), and energy efficiency. Thermal discomfort is quantified as the deviation of indoor temperature from the acceptable range of 21–24 °C. Specifically, the discomfort penalty grows in proportion to the extent that temperatures fall below 21 °C or exceed 24 °C, reflecting the well-documented impacts of such conditions on health, concentration, and perceived comfort. Additionally, IAQ penalties are imposed whenever CO_2 concentrations exceed the threshold of 1000 ppm, which is widely accepted in the literature. Importantly, the magnitude of the penalty is proportional to the degree of exceedance. The energy term serves as a proxy cost, calculated as the magnitude of changes to setpoints and the activation of ventilation boosts. By structuring the cost function in this way, the weights can be adapted in different contexts, for example, prioritizing health during periods of high respiratory illness risk, or energy during demand-response events. This adaptability makes the framework suitable for diverse use cases within the households.

The control actions available to the system are discrete and deliberately simple to ensure feasibility in real-world deployments. Candidate temperature setpoint adjustments are defined as changes of -2, -1, 0, +1, or +2 °C relative to the current setpoint. Ventilation control is modelled as a binary choice between normal operation and a temporary boost mode. For each candidate action, the system applies lightweight "what-if" response models calibrated from historical data to approximate the expected effect on future indoor conditions. Adjustments to temperature forecasts are modelled as:

$$T' = T - \gamma_C * \max(0, \Delta) + \gamma_h * \max(0, \Delta)$$

where cooling and heating responses are separately parameterized by coefficients γ_C and γ_h. This simple linear formulation allows the system to capture the first-order impacts of thermostat adjustments on indoor climate. Ventilation effects are modelled by reducing the predicted CO_2 concentration according to:

$$CO_2' = CO_2 - \eta * 1_{\{boost\}}$$

where η quantifies the expected reduction from a ventilation boost event, and the indicator function ensures the adjustment is only applied under that condition.

These response models allow the controller to simulate the consequences of all possible actions. Each hypothetical scenario yields adjusted forecasts of indoor temperature and CO_2 concentration, which are then passed to the cost function. The action associated with the lowest weighted cost is selected and implemented, ensuring that decisions are both data-driven and forward-looking. This design emphasizes computational efficiency while maintaining alignment with the physical behaviour of HVAC and ventilation systems, making it well-suited for real-time operation. Finally, the functionality of the application in this version can be summarized in the following screenshot. It contains all the information available to the user in terms of different supported features.