

---

# **PvProtect**

## **Software Design Specification**

**Version 1.4 approved**  
**Jiaying Liu, Pablo Perez, Ilia Lecha, Lluís Giménez**  
**UPF - ESCI. Bioinformatics**  
**19-5-2023**

## Revision History

Name	Date	Reason For Changes	Version
Introduction	16/04/2023	Filling the introduction section.	1.0
Overall description	20/04/2023	Filling the overall description section.	1.1
Functional Requirements	19/05/2023	Adding a list of functional and non-functional requirements.	1.2
Requirements relationships	30/05/2023	Adding all the relationships between requirements	1.3
General formatting	02/06/2023	Setting format to the whole document.	1.4
Adding diagrams	06/06/2023	Adding all activity, sequence, class, use case diagrams.	1.5
Diagram description	09/06/2023	Describing the diagrams	

# Table of Contents

<b>1. Introduction</b>	<b>1</b>
<b>1.1 Purpose</b>	<b>1</b>
<b>1.2 Document Organization</b>	<b>1</b>
<b>1.3 Intended Audience and Reading Suggestions</b>	<b>1</b>
<b>1.4 Project Scope</b>	<b>2</b>
<b>1.5 References</b>	<b>2</b>
<b>2. Overall Description</b>	<b>2</b>
<b>2.1 Product Perspective</b>	<b>2</b>
<b>2.2 Product Features</b>	<b>3</b>
<b>2.3 User Classes and Characteristics</b>	<b>3</b>
<b>2.4 Operating Environment</b>	<b>4</b>
<b>2.5 Design and Implementation Constraints</b>	<b>5</b>
<b>2.6 User Documentation</b>	<b>5</b>
<b>2.7 Assumptions and Dependencies</b>	<b>6</b>
<b>3. System Features</b>	<b>6</b>
<b>3.1 Table of requirements</b>	<b>6</b>
<b>3.2 Requirement details</b>	<b>7</b>
<b>4. External Interface Requirements</b>	<b>12</b>
<b>4.1 User Interfaces</b>	<b>12</b>
<b>4.2 Hardware Interfaces</b>	<b>13</b>
<b>4.3 Software Interfaces</b>	<b>15</b>
<b>4.4 Communications Interfaces</b>	<b>17</b>
<b>5. Other Nonfunctional Requirements</b>	<b>18</b>
<b>5.1 Performance Requirements</b>	<b>18</b>
<b>5.2 Safety Requirements</b>	<b>18</b>
<b>5.3 Security Requirements</b>	<b>19</b>
<b>5.4 Software Quality Attributes</b>	<b>19</b>
<b>6. Software diagrams</b>	<b>20</b>
<b>6.1 Architecture diagram</b>	<b>20</b>
<b>6.2 Use case diagrams</b>	<b>21</b>
<b>6.3 Activity and Sequence diagrams</b>	<b>22</b>
<b>6.4 Class diagram</b>	<b>32</b>

# **1. Introduction**

The PVProtect software, an innovative solution in the domain of renewable energy technology, is specifically designed to optimize the usage of solar panel systems. Solar panels are widely recognized as devices that convert sunlight into electricity, offering a sustainable source of energy for various applications. However, over time, solar panels experience a significant decline in efficiency. To address this challenge and maximize their lifespan, we have developed PVProtect.

PVProtect employs an intelligent algorithm that dynamically activates or deactivates the solar panels based on real-time factors, including battery level (if applicable), energy consumption, and energy production. The primary objective of the software is to minimize the duration of solar panel operation while maximizing self-consumption of the generated energy.

This Software Requirements Specification (SRS) document outlines the detailed requirements and functionalities of PVProtect. It serves as a comprehensive guide for the development team, stakeholders, and other relevant parties involved in the project. The document presents a clear understanding of the software's purpose, its intended users, and the features it offers.

## **1.1 Purpose**

The purpose of PVProtect is to provide an intelligent and automated solution that optimizes the operation of solar panel systems. By strategically activating or deactivating the panels, PVProtect aims to extend their lifespan, enhance their efficiency, and increase self-consumption of the generated energy. This software offers an intuitive application interface through which users can monitor energy production, consumption, and savings, while also receiving alerts regarding potential efficiency drops or maintenance needs.

## **1.2 Document Organization**

This SRS document is structured to provide a comprehensive understanding of PVProtect. It includes sections covering system overview, functional requirements, non-functional requirements, user interfaces, system constraints, and other pertinent aspects. Additionally, this document aims to ensure effective communication between the development team, stakeholders, and users throughout the software development lifecycle.

## **1.3 Intended Audience and Reading Suggestions**

PVProtect is designed to cater to a diverse range of users, including:

- Citizens: Individuals who have solar panel systems installed in their residential properties.
- Companies: Businesses of varying sizes and sectors that rely on solar energy for their operations.
- Factories: Industrial facilities that utilize solar panels to meet their energy requirements. The software empowers these users to optimize their energy consumption, reduce dependence on the electric grid, and contribute to a more sustainable future.

## 1.4 Project Scope

The scope of the PVProtect project encompasses the development of a software solution that optimizes the usage of solar panel systems to prolong their lifespan and maximize self-consumption. PVProtect will provide automated control over solar panel activation and deactivation based on real-time factors such as battery level, energy consumption, and energy production.

- **Optimized Solar Panel Operation:** PVProtect will intelligently activate or deactivate solar panels to minimize their usage time while maximizing self-consumption. It will consider factors such as battery level (if applicable), energy consumption, and energy production to make informed decisions about panel activation.
- **Energy Monitoring and Analysis:** The software will provide users with real-time data regarding energy production and consumption. Users will be able to track the amount of energy generated by the solar panels, the amount consumed, and the proportion of energy provided by the solar panel system compared to the electric network.
- **Alerts and Notifications:** PVProtect will include an alert system that notifies users of potential efficiency drops in the solar panel system. This will help identify situations where cleaning or maintenance may be required due to obstructions or reduced panel performance.
- **User Interaction and Control:** PVProtect will offer an intuitive application interface through which users can monitor and interact with the software. The application will provide features such as a button to manually connect or disconnect the solar panel system, the ability to view the status of panel activation, and access to historical data on panel usage.

The PVProtect software will be primarily designed for use by citizens, companies, and factories that have solar panel systems installed. It aims to empower users to optimize their energy consumption, reduce reliance on the electric grid, and contribute to sustainable energy practices.

## 1.5 References

URL to GitHub project repository: <https://github.com/jiayiingliu/PVProtect>

# 2. Overall Description

## 2.1 Product Perspective

PVProtect is a standalone software system that operates independently and does not require integration with any existing solar panel management or control systems. It acts as an intelligent layer on top of the solar panel system, providing optimized control and monitoring functionalities.

PVProtect interacts with the solar panel system through compatible hardware interfaces or communication protocols. It does not modify the underlying hardware components but instead utilizes real-time data from the system to make informed decisions about solar panel activation and deactivation.

## 2.2 Product Features

The PVProtect software offers a range of features designed to optimize solar panel usage and enhance user experience. The key features include:

1. **Automated Panel Activation/Deactivation:** PVProtect utilizes an intelligent algorithm to automatically activate or deactivate the solar panel system based on real-time factors, including battery level (if applicable), energy consumption, and energy production. This feature ensures efficient energy utilization and maximizes self-consumption.
2. **Energy Monitoring and Analysis:** PVProtect provides users with real-time data on energy production and consumption. Users can track the amount of energy generated by the solar panels, the amount consumed, and the percentage of energy provided by the solar panel system compared to the electric network.
3. **Efficiency Drop Warnings:** The software includes an alert system that notifies users when a decrease in solar panel efficiency is detected. This feature helps identify potential obstructions or maintenance needs that may be impacting panel performance.
4. **User Interaction and Control:** PVProtect offers an intuitive application interface through which users can monitor and interact with the software. Users can view the status of panel activation, access historical data on panel usage, and manually connect or disconnect the solar panel system if necessary (though automated control is recommended for optimal results).
5. **Cost Savings Tracking:** PVProtect calculates and displays the amount of money saved by utilizing solar panels instead of relying solely on the electric network. This feature provides users with a tangible measure of the economic benefits of their solar panel investment.

## 2.3 User Classes and Characteristics

PVProtect is designed to cater to the following user classes, each with their own unique characteristics and requirements:

- **Citizens:** Individuals who have solar panel systems installed in their residential properties. They may have varying levels of technical knowledge and may prioritize sustainability, energy savings, and environmental consciousness. They may also be interested in monitoring their energy consumption patterns and the cost savings achieved through solar panel usage.
- **Companies:** Businesses of varying sizes and sectors that rely on solar energy for their operations. These users may have more sophisticated energy management requirements and may seek to optimize their energy usage and reduce dependence on the electric grid. They may also have specific reporting needs and the desire to track the performance of their solar panel systems.
- **Factories:** Industrial facilities that utilize solar panels to meet their energy requirements. These users may have large-scale solar panel installations and complex energy consumption patterns. They may require advanced monitoring and control capabilities to manage energy usage efficiently and track the performance of multiple solar panel systems.

The users of PVProtect may possess the following characteristics:

- **Technical Proficiency:** Users may have varying levels of technical expertise, ranging from novice to advanced. The software should provide an intuitive user interface and clear instructions to accommodate users with different levels of technical knowledge.

- **Monitoring and Control Needs:** Users may have a strong desire to monitor and control their solar panel systems. They may want to track energy production, consumption, and savings, and have the ability to manually connect or disconnect the panels when necessary.
- **Sustainability Focus:** Users may prioritize sustainability and environmental consciousness. They may appreciate the eco-friendly benefits of solar panel systems and seek to maximize their self-consumption of clean energy.
- **Cost-consciousness:** Users may have a strong interest in cost savings achieved through solar panel usage. They may value features that provide insights into the financial benefits of using solar energy and the return on their investment.
- **Data Analysis and Reporting:** Some users, particularly companies and factories, may require advanced data analysis and reporting capabilities. They may want to generate reports, analyze energy consumption patterns, and make informed decisions about energy management strategies.

The PVProtect software aims to accommodate the needs and characteristics of these user classes by providing a user-friendly interface, customizable monitoring features, and relevant data analysis options.

## 2.4 Operating Environment

PVProtect operates in an environment with the following characteristics:

- **Hardware Requirements:** PVProtect requires compatible hardware interfaces or communication protocols to interact with the solar panel system. The hardware should support the necessary data exchange and control capabilities.
- **Software Requirements:** PVProtect is designed to run on compatible operating systems, including but not limited to Windows, macOS, and Linux. The software may require specific system libraries or dependencies, which will be documented separately.
- **Network Connectivity:** PVProtect may require network connectivity to access real-time data and send notifications. It is assumed that the software will operate in an environment with stable network connectivity.
- **User Interface:** PVProtect provides a user-friendly application interface that can be accessed through compatible devices such as smartphones, tablets, or computers. The application may be available as a standalone installation or as a web-based application, depending on the specific implementation.

## 2.5 Design and Implementation Constraints

PVProtect must adhere to the following design and implementation constraints:

- **Hardware Compatibility:** The software must be compatible with the hardware interfaces or communication protocols used by the solar panel system. Compatibility constraints should be considered during the development and integration phases to ensure seamless interaction with the system.
- **Real-Time Data Processing:** PVProtect requires the ability to process and analyze real-time data from the solar panel system efficiently. The software should be designed to handle data streams in a timely manner and make optimal decisions based on the received information.
- **Scalability:** PVProtect should be designed to accommodate a range of solar panel system sizes, from residential installations to large-scale industrial setups. The software should be scalable to handle varying numbers of solar panels and efficiently manage data for different user scenarios.
- **Reliability and Robustness:** PVProtect should be reliable and robust, ensuring consistent and accurate control over solar panel activation and deactivation. It should be able to handle unexpected situations, such as intermittent network connectivity or power fluctuations, and gracefully recover without compromising functionality.

## 2.6 User Documentation

PVProtect requires comprehensive user documentation to aid users in understanding the software and effectively utilizing its features. The user documentation will include:

- **Installation Guide:** Step-by-step instructions for installing and configuring the PVProtect software on the user's device.
- **User Manual:** A detailed guide on how to use the PVProtect application interface, including explanations of various features, options, and controls available to the user.
- **Troubleshooting Guide:** Documentation that provides troubleshooting steps for common issues users may encounter while using PVProtect. It will offer solutions or workarounds to help users resolve problems effectively.
- **FAQs and Knowledge Base:** A collection of frequently asked questions (FAQs) and a knowledge base that addresses common inquiries and provides additional information about PVProtect and its functionalities.



## 2.7 Assumptions and Dependencies

The development and successful operation of PVProtect are based on the following assumptions and dependencies:

- **Assumptions:**

The solar panel system is properly installed, functional, and capable of providing real-time data on energy production and consumption.

The underlying hardware and communication interfaces necessary for PVProtect's interaction with the solar panel system are available and compatible.

Sufficient network connectivity is available to facilitate data exchange and notifications between PVProtect and the solar panel system.

- **Dependencies:**

PVProtect is dependent on the availability and compatibility of the necessary hardware interfaces or communication protocols provided by the solar panel system.

The software relies on the availability of a stable network connection to access real-time data from the solar panel system and send notifications to users.

## 3. System Features

### 3.1 Table of requirements

Functional Requirements		Nonfunctional Requirements	
<b>F1</b>	Connect/disconnect panel.	<b>NF1</b>	User-friendly interface
<b>F2</b>	Energy production and consumption display.	<b>NF2</b>	High reliability and availability
<b>F3</b>	Electric network energy calculation.	<b>NF3</b>	Efficient algorithm
<b>F4</b>	Warning generation	<b>NF4</b>	Solar panel compatibility
<b>F5</b>	Emergency system disconnect button	<b>NF5</b>	Scalability
<b>F6</b>	Calculate savings	<b>NF6</b>	Data security
<b>F7</b>	On/Off racking	<b>NF7</b>	Multiple language support
<b>F8</b>	Current activation status	<b>NF8</b>	Device compatibility
<b>F9</b>	Solar panel energy consumption percentage calculation.	<b>NF9</b>	Data monitoring efficiency
<b>F10</b>	Energy sent to electric network percentage calculation	<b>NF10</b>	Minimal resource consumption
<b>F11</b>	Performance tracking	<b>NF11</b>	Scalability
		<b>NF12</b>	Energetic efficiency
		<b>NF13</b>	Robust error handling
		<b>NF14</b>	Fast and reliable time connection
		<b>NF15</b>	Data retrieval efficiency

### 3.2 Functional requirement details

<b>F1</b>	<b>Connect/Disconnect panel automatically</b>	<b>Version: 1.0</b>
This functionality allows the system to make the correct decision to activate or deactivate their solar panels according to the energetic needs of the user. To take this decision, it will use the information provided by the different sensors.		
<b>Relation:</b> F8, NF13		

<b>F2</b>	<b>Energy production and consumption display</b>	<b>Version: 1.0</b>
The PVProtect application provides users with real-time data on energy production and consumption. Users can view the current energy production from their solar panels as well as the energy consumption within their household or facility, enabling them to monitor and track their energy usage in real-time.		
<b>Relation:</b> F6, F9, F10, NF2, NF9, NF14		

<b>F3</b>	<b>Electric network energy calculation</b>	<b>Version: 1.0</b>
The PVProtect application calculates and displays the percentage of energy sourced from the electric network. By presenting this information, users can understand the extent to which they rely on the grid for their energy needs, helping them assess their energy independence and the effectiveness of their solar panel system.		
<b>Relation:</b> F9, NF2, NF3		

<b>F4</b>	<b>Warning generation</b>	<b>Version: 1.0</b>
PVProtect monitors the efficiency of the solar panels and generates warnings if a significant drop in efficiency is detected. These warnings alert users to potential obstructions, such as dust or debris on the panels, which may be impacting their energy production. This allows users to take appropriate action, such as cleaning or maintenance, to optimize panel performance.		
<b>Relation:</b> F11, NF2, NF13		

<b>F5</b>	<b>Connect/Disconnect panel manually</b>	<b>Version: 1.0</b>
This functionality allows users to remotely connect or disconnect their solar panels using the PVProtect application. By clicking a button within the application, users can control the activation status of their panels, enabling or disabling their energy production as needed.		
<b>Relation:</b> F8, NF13		

<b>F6</b>	<b>Calculate savings</b>	<b>Version: 1.0</b>
PVProtect calculates and displays the cost savings achieved by utilizing solar panels instead of relying solely on the electric network. By presenting this information, users can track and quantify the financial benefits of their solar panel investment, promoting awareness of the cost-effectiveness of renewable energy.		
<b>Relation:</b> F2, F3, F9, NF2, NF6		

<b>F7</b>	<b>On/Off tracking</b>	<b>Version: 1.0</b>
The PVProtect application tracks and records the specific times when the solar panels are turned on or off. This feature provides users with a historical log of panel activity, allowing them to analyze usage patterns and make informed decisions regarding panel activation based on their energy consumption patterns.		
<b>Relation:</b> F1, F5, F8		

<b>F8</b>	<b>Current activation status</b>	<b>Version: 1.0</b>
PVProtect clearly indicates the current activation status of the solar panels in the application. Users can easily see whether the panels are currently activated or deactivated, providing them with instant visibility into the operational state of their panels and their energy generation status.		
<b>Relation:</b> F7		

<b>F9</b>	<b>Solar panel energy consumption calculation</b>	<b>Version: 1.0</b>
This functionality calculates and displays the percentage of the energy generated by the solar panels that is consumed internally within the user's household or facility. Users can gauge the level of self-consumption and optimize their energy usage to maximize the benefits of their solar panel system.		
<b>Relation:</b> F2, F10, NF2		

<b>F10</b>	<b>Energy sent to electric network calculation</b>	<b>Version: 1.0</b>
PVProtect calculates and presents the percentage of excess energy generated by the solar panels that is sent back to the electric network. This information allows users to understand the amount of energy exported to the grid, aiding in evaluating the system's contribution to renewable energy generation.		
<b>Relation:</b> F2, NF2		

<b>F11</b>	<b>Performance tracking</b>	<b>Version: 1.0</b>
PVProtect measures and tracks the performance evolution of the solar panel system over time. It collects data on energy production, efficiency, and other relevant metrics to provide users with insights into the long-term performance of their panels. This feature facilitates the assessment of panel health, degradation, and overall system effectiveness.		
<b>Relation:</b> F2, NF2, NF9, NF14		

### 3.3 Non-functional requirements details

<b>NF1</b>	<b>User-friendly interface</b>	<b>Version: 1.0</b>
The system should have a user-friendly interface that is intuitive and easy to navigate, allowing users to interact with the system without any difficulties. The interface should be designed in a way that promotes efficient and effective user interactions, minimizing the learning curve and ensuring a positive user experience.		
<b>Relation:</b> NF7, NF13, NF14		

<b>NF2</b>	<b>High reliability and availability</b>	<b>Version: 1.0</b>
The system should exhibit high reliability and availability, ensuring that it remains operational and accessible to users with minimal downtime. It should be designed to handle unexpected failures gracefully, recover quickly from errors, and maintain a high level of system uptime to meet user demands and expectations.		
<b>Relation:</b> NF3, NF12, NF13		

<b>NF3</b>	<b>Efficient algorithm</b>	<b>Version: 1.0</b>
The system should employ efficient algorithms and data processing techniques to perform tasks and computations in a timely manner. The algorithms should be designed to optimize resource utilization, minimize computational complexity, and maximize the system's overall performance and responsiveness.		
<b>Relation:</b> NF12		

<b>NF4</b>	<b>Solar panel compatibility</b>	<b>Version: 1.0</b>
The system should be compatible with solar panel technology, allowing it to operate using solar power as a renewable energy source. It should be designed to efficiently utilize solar energy, provide compatibility with solar panel systems, and optimize energy consumption to reduce reliance on traditional power sources.		
<b>Relation:</b> NF8		

<b>NF5</b>	<b>Scalability</b>	<b>Version: 1.0</b>
The system should be scalable, capable of accommodating increased workload and user demands as the system usage grows over time. It should be designed to handle additional users, data, and processing requirements by effectively distributing resources, scaling up or down, and maintaining optimal performance without sacrificing functionality or user experience.		
<b>Relation:</b> None		

<b>NF6</b>	<b>Data security</b>	<b>Version: 1.0</b>
The system should prioritize data security, implementing robust measures to protect sensitive information from unauthorized access, alteration, or disclosure. It should incorporate encryption, access controls, authentication mechanisms, and other security measures to ensure the confidentiality, integrity, and availability of user data and system resources.		
<b>Relation:</b> None		

<b>NF7</b>	<b>Multiple language support</b>	<b>Version: 1.0</b>
------------	----------------------------------	---------------------

The system should support multiple languages, allowing users to interact with the system in their preferred language. It should provide language localization capabilities, enabling the user interface, error messages, and system responses to be presented in different languages based on user preferences or system configurations.

**Relation:** None

**NF8**

**Device compatibility**

**Version: 1.0**

The system should be compatible with a wide range of devices, including desktop computers, laptops, tablets, and smartphones. It should be responsive and adaptable, ensuring that the user interface and system functionality are accessible and optimized for different device types, operating systems, and screen sizes.

**Relation:** NF4

**NF9**

**Data monitoring efficiency**

**Version: 1.0**

The system should provide efficient data monitoring capabilities, allowing real-time or near-real-time monitoring and analysis of data generated by the system. It should enable timely detection of anomalies, performance issues, or critical events, and provide relevant notifications or alerts to users or system administrators for prompt action or decision-making.

**Relation:** NF13

**NF10**

**Minimal resource consumption**

**Version: 1.0**

The system should strive for minimal resource consumption, optimizing the utilization of computational resources, memory, storage, and network bandwidth. It should be designed to minimize the system's footprint, reduce energy consumption, and optimize resource allocation to enhance system performance, scalability, and cost-effectiveness.

**Relation:** NF3

**NF11**

**Energetic efficiency**

**Version: 1.0**

The system should prioritize energetic efficiency, optimizing the use of energy resources to reduce energy consumption and minimize environmental impact. It should incorporate energy-efficient design principles, power management techniques, and intelligent resource allocation strategies to achieve optimal energy efficiency without compromising system performance or functionality.

**Relation:** NF3

**NF12 Robust error handling**

**Version: 1.0**

The system should implement robust error handling mechanisms, capable of detecting, reporting, and recovering from errors or exceptions gracefully. It should provide clear and meaningful error messages, log error details for debugging and troubleshooting purposes, and ensure that the system remains stable and operational even in the presence of unexpected errors or failures.

**Relation:** None

**NF13 Fast and reliable time connection**

**Version: 1.0**

The system should establish a fast and reliable time connection, allowing accurate time synchronization with reliable time sources. It should ensure that time-sensitive operations, such as time-stamping, scheduling, or event synchronization, are performed with precision, reliability, and minimal time discrepancies across the system.

**Relation:** None

**NF14 Data retrieval efficiency**

**Version: 1.0**

The system should provide efficient data retrieval capabilities, allowing users to access and retrieve relevant data quickly and accurately. It should employ optimized data retrieval techniques, indexing mechanisms, caching strategies, or data organization methods to minimize latency, enhance search performance, and enable rapid access to requested information or records.

**Relation:** NF3

## 4. External Interface Requirements

### 4.1 User Interfaces

PVProtect requires user interfaces to interact with the software and manage the solar panel system. The following user interfaces are needed:

## 2. PVProtect Dashboard:

- Logical Characteristics: The dashboard provides an overview of the solar panel system's performance and allows users to access various features and information.

- Sample Screen Image: [Provide a sample image of the dashboard screen, if available.]

- GUI Standards: Follow a clean and intuitive design with clear visual representations of data.

- Screen Layout Constraints: Use a responsive layout that adapts to different screen sizes.

- Standard Buttons and Functions: Include a navigation menu, home button, and settings button for easy access to different sections and configurations.

- Keyboard Shortcuts: Not applicable.

- Error Message Display Standards: Display error messages in a prominent and understandable manner, with clear instructions on how to resolve issues.

## 3. Energy Monitoring:

- Logical Characteristics: This interface provides real-time data on energy production and consumption.

- Sample Screen Image: [Provide a sample image of the energy monitoring screen, if available.]

- GUI Standards: Use clear visualizations, such as graphs or meters, to display energy production and consumption information.

- Screen Layout Constraints: Organize the data in a visually appealing and easy-to-understand format.

- Standard Buttons and Functions: Include options to switch between different time periods (e.g., daily, weekly, monthly) for energy data analysis.

- Keyboard Shortcuts: Not applicable.

- Error Message Display Standards: Not applicable.

## 4. Threshold Settings:

- Logical Characteristics: This interface allows users to set thresholds for production and consumption that determine the activation or deactivation of the solar panels.

- Sample Screen Image: [Provide a sample image of the threshold settings screen, if available.]

- GUI Standards: Use intuitive input fields or sliders to set threshold values.

- Screen Layout Constraints: Organize the threshold settings in a clear and structured manner.

- Standard Buttons and Functions: Include buttons to save and apply the threshold settings.

- Keyboard Shortcuts: Not applicable.

- Error Message Display Standards: Display error messages if invalid threshold values are entered or if the settings cannot be saved.

## 5. Efficiency Warnings:

- Logical Characteristics: This interface alerts users when there is a drop in panel efficiency due to potential obstructions or issues.

- Sample Screen Image: [Provide a sample image of the efficiency warnings screen, if available.]

- GUI Standards: Use clear visual indicators or notifications to highlight efficiency warnings.

- Screen Layout Constraints: Display efficiency warnings prominently to ensure they catch the user's attention.



- Standard Buttons and Functions: Include options to acknowledge or dismiss efficiency warnings.
  - Keyboard Shortcuts: Not applicable.
  - Error Message Display Standards: Not applicable.
6. Emergency Disconnect:
- Logical Characteristics: This interface allows users to immediately disconnect the solar panels in case of a fire emergency.
  - Sample Screen Image: [Provide a sample image of the emergency disconnect screen, if available.]
  - GUI Standards: Use a clear and prominent emergency disconnect button or confirmation dialog.
  - Screen Layout Constraints: Ensure that the emergency disconnect option is easily accessible.
  - Standard Buttons and Functions: Include a confirmation dialog or additional steps to prevent accidental disconnections.
  - Keyboard Shortcuts: Not applicable.
  - Error Message Display Standards: Not applicable.

## 4.2 Hardware Interfaces

PVProtect software interacts with hardware components of the solar panel system. The following hardware interfaces are required:

1. Solar Panels:
  - Logical Characteristics: PVProtect software communicates with the solar panels to receive data on energy production and send control signals to activate or deactivate them.
  - Physical Characteristics: Solar panels typically consist of photovoltaic cells and are connected to the system through electrical connections.
  - Supported Device Types: Solar panels compatible with the system's communication protocols.
  - Data and Control Interactions: PVProtect software receives energy production data from the solar panels and sends control signals to activate or deactivate them based on the optimization algorithm.
  - Communication Protocols: Communication protocols such as Modbus, CAN bus, or other industry-standard protocols may be used for data exchange between the software and solar panels.
2. Batteries (if applicable):
  - Logical Characteristics: If the system includes batteries for energy storage, PVProtect software interacts with the battery system to monitor battery level and optimize panel usage accordingly.
  - Physical Characteristics: Batteries used for energy storage within the system.
  - Supported Device Types: Battery systems compatible with the system's communication protocols.
  - Data and Control Interactions: PVProtect software receives battery level data from the battery system and adjusts the panel activation or deactivation based on the optimization algorithm.
  - Communication Protocols: Communication protocols such as Modbus, CAN bus, or other industry-standard protocols may be used for data exchange between the software and battery systems.
3. Energy Consumption Meter:

- Logical Characteristics: PVProtect software communicates with the energy consumption meter to receive data on energy consumed by the house or building.
- Physical Characteristics: Energy consumption meter installed within the electrical system of the house or building.
- Supported Device Types: Energy consumption meters compatible with the system's communication protocols.
- Data and Control Interactions: PVProtect software receives energy consumption data from the meter to inform the panel activation or deactivation decisions.
- Communication Protocols: Communication protocols such as Modbus, CAN bus, or other industry-standard protocols may be used for data exchange between the software and energy consumption meter.

#### 4. Fire Emergency System:

- Logical Characteristics: PVProtect software interfaces with the fire emergency system to receive signals for immediate panel deactivation in case of a fire emergency.
- Physical Characteristics: Fire emergency system components such as smoke detectors or fire alarm systems.
- Supported Device Types: Fire emergency systems compatible with the system's communication protocols or signal inputs.
- Data and Control Interactions: PVProtect software receives signals or commands from the fire emergency system to initiate an immediate panel deactivation.
- Communication Protocols: Communication protocols or standard signal inputs as per the requirements of the fire emergency system.

### 4.3 Software Interfaces

PVProtect software interfaces with other software components and systems. The following software interfaces are required:

#### 1. Application Programming Interface (API):

- Connections: PVProtect software may need to integrate with external software components, databases, or services via APIs.
- Purpose: API connections allow data exchange and integration between PVProtect and other software systems.
- Data Items/Messages: Data items/messages exchanged through the API will depend on the specific integration requirements and functionalities.
- Services Needed: PVProtect may require API services for weather data, energy pricing, or data logging/storage, among others.
- Nature of Communications: API communications can involve data retrieval, data submission, or real-time data streaming, depending on the integration needs.

- Documentation: Detailed API protocols should be referenced for each specific integration.
2. Database System:
- Connections: PVProtect may interact with a database system for data storage and retrieval.
  - Purpose: The database system stores relevant data, such as energy production/consumption logs, system settings, and performance metrics.
  - Data Items/Messages: PVProtect will interact with the database to store and retrieve data such as energy production, consumption, system configurations, and user settings.
  - Services Needed: Database services for data storage, retrieval, and querying.
  - Nature of Communications: PVProtect will communicate with the database system using database query languages (e.g., SQL) or database-specific APIs.
  - Documentation: Reference documentation for the specific database system should be consulted for implementation details.
3. Operating System:
- Connections: PVProtect software runs on a specific operating system.
  - Purpose: The operating system provides the underlying platform for PVProtect to execute and manage system resources.
  - Data Items/Messages: Data items/messages related to system resource management, process scheduling, and file operations are handled by the operating system.
  - Services Needed: Services provided by the operating system include process management, memory management, file system access, and interprocess communication.
  - Nature of Communications: PVProtect interacts with the operating system through system calls and APIs provided by the operating system.
  - Documentation: Reference the documentation of the specific operating system for detailed information on system interfaces and APIs.
4. Integrated Commercial Components:
- Connections: PVProtect may utilize integrated commercial components such as libraries or software modules.
  - Purpose: Integrated commercial components enhance the functionality or performance of PVProtect through pre-built or third-party software.
  - Data Items/Messages: Data items/messages exchanged with integrated commercial components depend on the specific functionalities provided by those components.
  - Services Needed: Services provided by integrated commercial components, such as advanced optimization algorithms, data analysis, or graphical visualization.
  - Nature of Communications: Communications with integrated commercial components can involve function calls, data passing, or configuration exchanges.
  - Documentation: Refer to the documentation or specifications provided by the integrated commercial components for details on their interfaces and usage.

## 4.4 Communications Interfaces

PVProtect software requires specific communication interfaces for various functions. The following communication interfaces and requirements are associated with the product:

### 1. Internet Connectivity:

- Requirements: PVProtect software requires an internet connection to access external services, retrieve weather data, fetch energy pricing information, or perform remote monitoring and control.

- Protocols: The software may utilize standard communication protocols such as HTTP or HTTPS for web-based interactions.

- Message Formatting: Data exchanged through web-based communication interfaces can be in various formats, such as JSON or XML, depending on the specific APIs or services used.

- Security and Encryption: Secure communication protocols like HTTPS should be implemented to ensure data privacy and integrity during transmission.

- Data Transfer Rates: Data transfer rates depend on the network connection and the specific services or data being exchanged.

### 2. Local Network Communication:

- Requirements: PVProtect software may communicate with devices or systems within the local network, such as energy consumption meters, battery management systems, or monitoring devices.

- Protocols: Communication protocols like Modbus, CAN bus, or other industry-standard protocols can be used for local network communication.

- Message Formatting: Data formats and message structures will depend on the specific protocols and devices being used for communication.

- Security and Encryption: Depending on the requirements and sensitivity of the data being exchanged, appropriate security measures, such as encryption or access control, may be implemented at the network level.

### 3. Email Notifications:

- Requirements: PVProtect software may send email notifications to users or administrators for system alerts, warnings, or performance reports.

- Protocols: Email notifications can be sent using standard email protocols such as SMTP (Simple Mail Transfer Protocol).

- Message Formatting: Emails can be formatted in HTML or plain text, depending on the content and formatting requirements.

- Security and Encryption: Standard email protocols generally support encryption options (e.g., TLS/SSL) for secure transmission of email messages.

### 4. System Integration Interfaces:

- Requirements: PVProtect software may need to integrate with other systems or software components through specific integration interfaces or APIs.

- Protocols: The communication protocols and interfaces will depend on the specific systems being integrated.

- Message Formatting: Data formatting and message structures will be defined by the integration interfaces or APIs being used.

- Security and Encryption: Security measures will depend on the security requirements of the integrated systems and the protocols being utilized.

## 5. Synchronization Mechanisms:

- Requirements: PVProtect software may require synchronization mechanisms for data consistency and real-time updates between different software components or systems.

- Mechanisms: Synchronization can be achieved through event-driven notifications, polling mechanisms, or real-time data streaming, depending on the specific requirements and capabilities of the integrated components.

- Data Transfer Rates: The data transfer rates will depend on the synchronization frequency and the volume of data being synchronized.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

- Response Time:

- PVProtect should provide a responsive user interface with a maximum response time of 1 second for user interactions, such as button clicks or data retrieval.

- The response time should be measured from the user's action to the system's acknowledgment or completion of the requested operation.

- Energy Production/Consumption Update Frequency:

- PVProtect should update energy production and consumption data in real-time or with a frequency of at least once every 5 seconds.

- Real-time or near real-time updates enable users to monitor and analyze their energy usage effectively.

- Scalability:

- PVProtect should be able to handle a growing number of connected solar panels and users without significant degradation in performance.

- The system should scale horizontally or vertically to accommodate increased user demand and data processing requirements.

### 5.2 Safety Requirements

- Electrical Safety:

- PVProtect should comply with relevant electrical safety standards and regulations to ensure the safe operation of the solar panel system.

- The software should not interfere with the proper functioning of safety mechanisms and protocols built into the hardware components.

- Emergency Shutdown:

- PVProtect should provide a mechanism for emergency shutdown of the solar panels in case of fire or other hazardous situations.
- The emergency shutdown feature should be easily accessible and clearly labeled to facilitate quick response in critical situations.

### 5.3 Security Requirements

- Data Privacy:
  - PVProtect should protect user data, including energy consumption patterns and user settings, from unauthorized access or disclosure.
  - User authentication and secure data transmission (e.g., HTTPS) should be implemented to ensure data privacy and integrity.
- Access Control:
  - PVProtect should enforce appropriate access controls to prevent unauthorized system configuration changes or tampering with critical settings.
  - User roles and permissions should be defined to restrict access to sensitive functionalities and data.

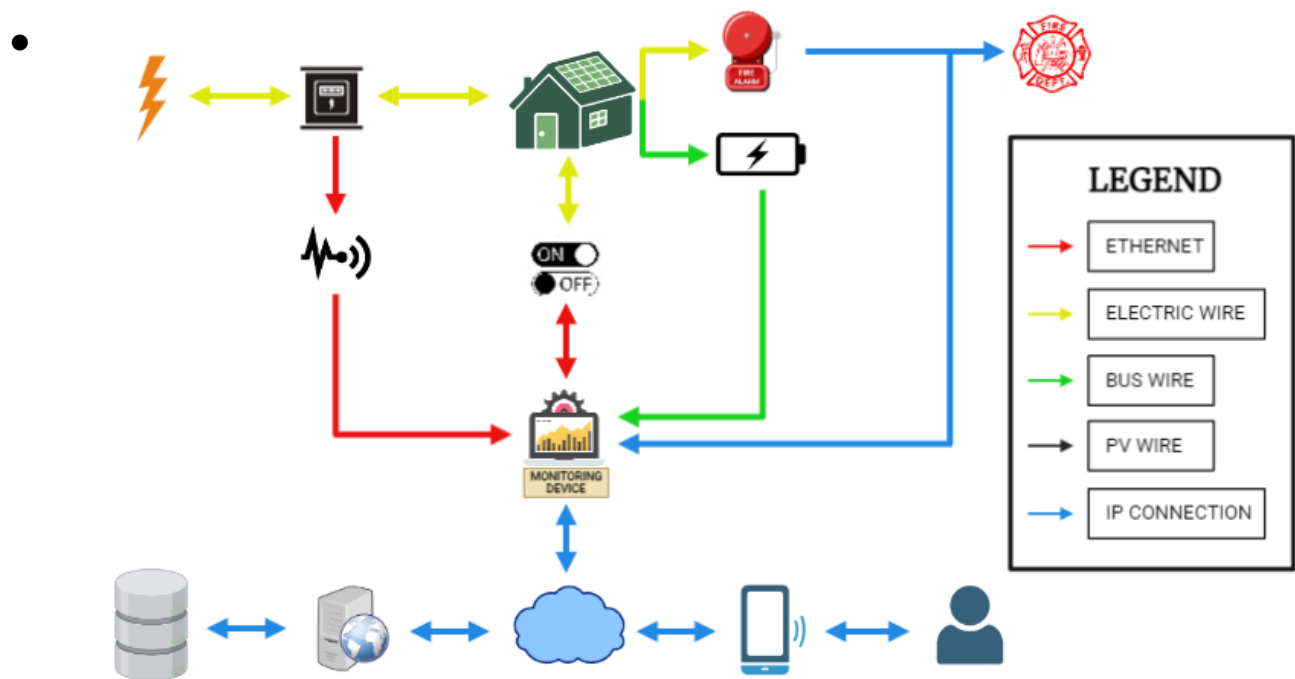
### 5.4 Software Quality Attributes

- Reliability:
  - PVProtect should operate reliably, ensuring minimal system downtime and data loss.
  - The software should handle exceptions and errors gracefully, providing informative error messages and appropriate recovery mechanisms.
- Maintainability:
  - PVProtect should be designed and implemented following good software engineering practices to facilitate future maintenance and updates.
  - The codebase should be well-structured, modular, and thoroughly documented to enable easy understanding and modification by developers.
- Usability:
  - PVProtect should have a user-friendly interface, allowing users to interact with the system intuitively and efficiently.
  - The system should provide clear and concise instructions, helpful error messages, and informative feedback to guide users in operating the software.
- Performance Efficiency:
  - PVProtect should optimize resource utilization and system performance to ensure efficient energy monitoring and control.
  - The software should be optimized to minimize computational overhead and memory usage, allowing for smooth and efficient operation.

- Interoperability:
  - PVProtect should support integration with a variety of solar panel systems, hardware devices, and software components commonly used in the renewable energy industry.
  - Compatibility with industry-standard communication protocols and data formats should be ensured to facilitate seamless integration.
- Security:
  - PVProtect should implement robust security measures to protect against unauthorized access, data breaches, and cyber-attacks.
  - Regular security audits, vulnerability assessments, and adherence to security best practices should be part of the software development and maintenance process.

## 6. Software diagrams

### 6.1 Architecture diagram



The provided Architecture Diagram shows the different physical elements that PVProtect software will monitorize. Let's see how these elements smoothly come together within the software's architecture.

We can divide the architecture in 2 parts:

- On the top we have all the elements that will be sending their information to the Monitoring Device
- On the bottom we have the other elements that display the information to the user and make the computations and storing of the data.

Now we will walk through each one of the elements and explain how they are connected to each other. So, at the top we have the following elements:

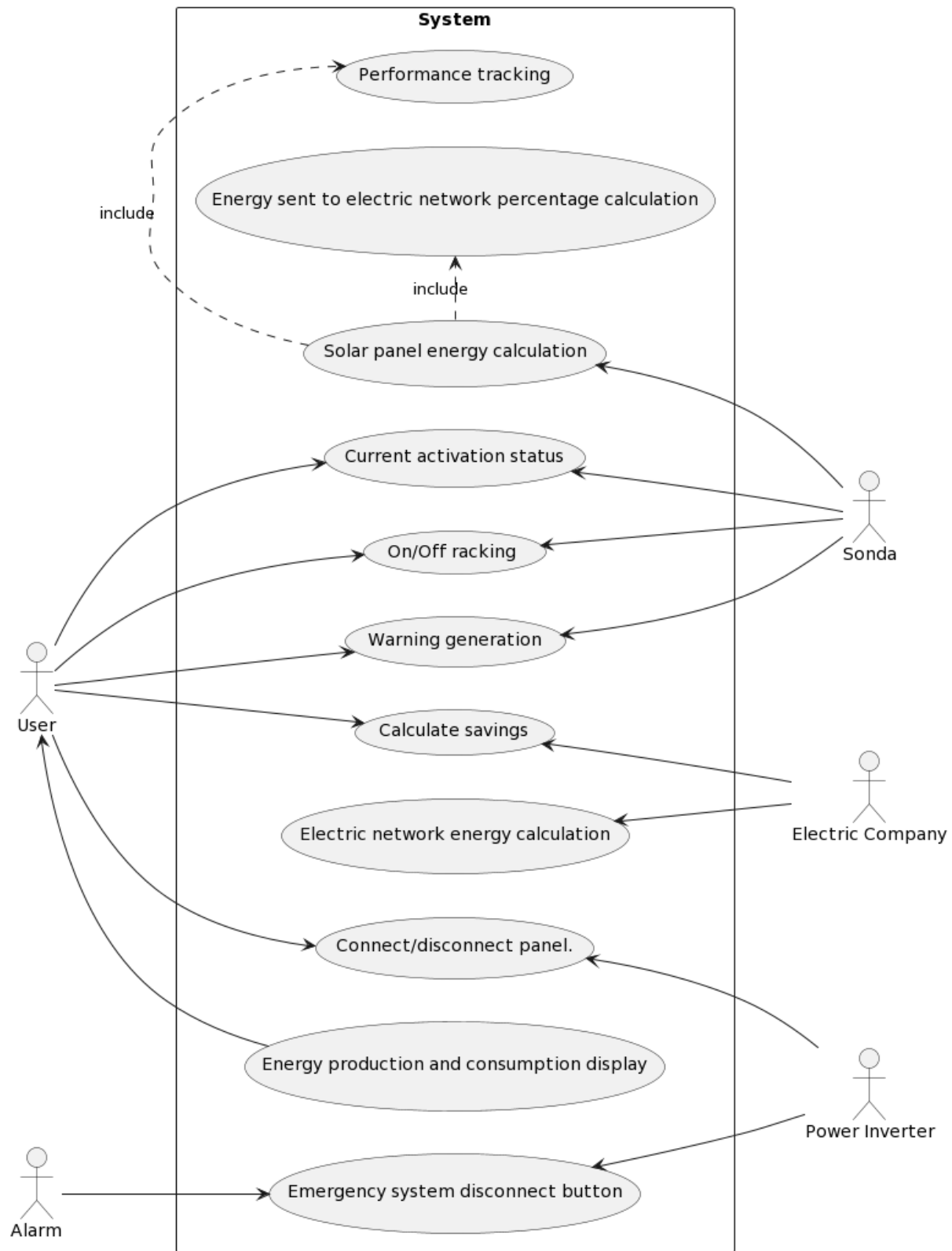
- ALARM, to detect if there is a fire emergency and if it is the case immediately call the firefighters and send the information to the monitoring service and switch off the solar panel. The alarm will be connected to the house via electric wire and via IP connection to the firefighters and monitoring device.
  
- BATTERY, the % of energy capacity in the battery bank. It will be connected to the house and to the monitoring service via bus wire, a special type of wire used to connect a battery with different elements that allows the high transmission rate and data flow since standard data cables are not enough.
  
- SENSOR, computes the flow of energy between the house and the electric network. We do not want to send energy to the electrical network. It will be connected to the monitoring device and electricity box via ethernet. At the same time the electricity box will be connected to the network and the house via electric wire.  
Note that the information provided by the sensor is directly related to the energy generated by the solar panel and the energy consumed by the house. So, we are not only monitoring the energy sent to the electric grid but also the other two key aspects mentioned.
  
- POWER INVERTER, it is responsible for turning on or off the solar panel depending on the data retrieved or a decision of the user or a fire emergency. It will be connected to the monitoring service via ethernet and via electric wire to the house.

At the bottom, we can see that is connected to the monitoring device through IP connection:

- The user will be able to access the web app with his smartphone, for example. Here he will find all the information mentioned before and he will also receive warnings to prevent a loss of efficiency due to the lack of maintenance of the PV panel (there is dust).
  
- Of course, we have a WEBSERVER that is connected to both monitoring device and database. This element will be in charge of making all the necessary computations that will be sent to the monitoring device, so that it has the correct inputs to decide whether to activate or not the solar panel
  
- Finally, we will use the DATABASE to keep track of the evolution of the efficiency of the panels, to know the amount of energy generated by the solar panels, at which time the panels were turned ON or OFF, the amount of saved money due to solar panels...



## 6.2 Use case diagrams



The provided UML diagram illustrates the interactions and relationships among various actors and use cases in a system. The diagram follows a left to right direction, and it represents a system with several actors and corresponding use cases.

The actors in the diagram include "User", "Sonda", "Alarm", "Electric Company" and "Power Inverter". These actors interact with the system to perform different tasks and access various functionalities.

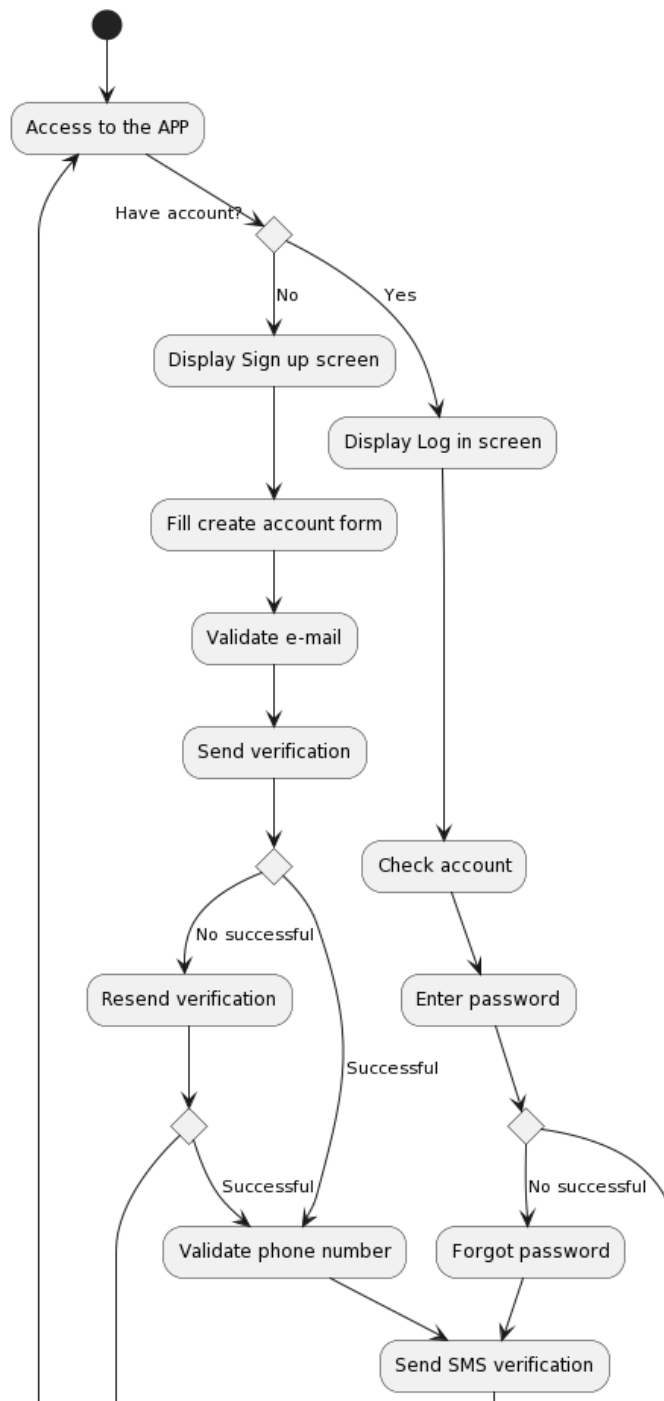
The main focus of the system is represented by the rectangle labeled "System." Inside the system, there are several use cases identified by their respective labels such as "Energy production and consumption display", "Connect/disconnect panel", "Solar panel energy calculation", "Electric network energy calculation", "Warning generation", "Emergency system disconnect button", "Calculate savings", "On/Off racking", "Current activation status", "Energy sent to electric network percentage calculation", and "Performance tracking".

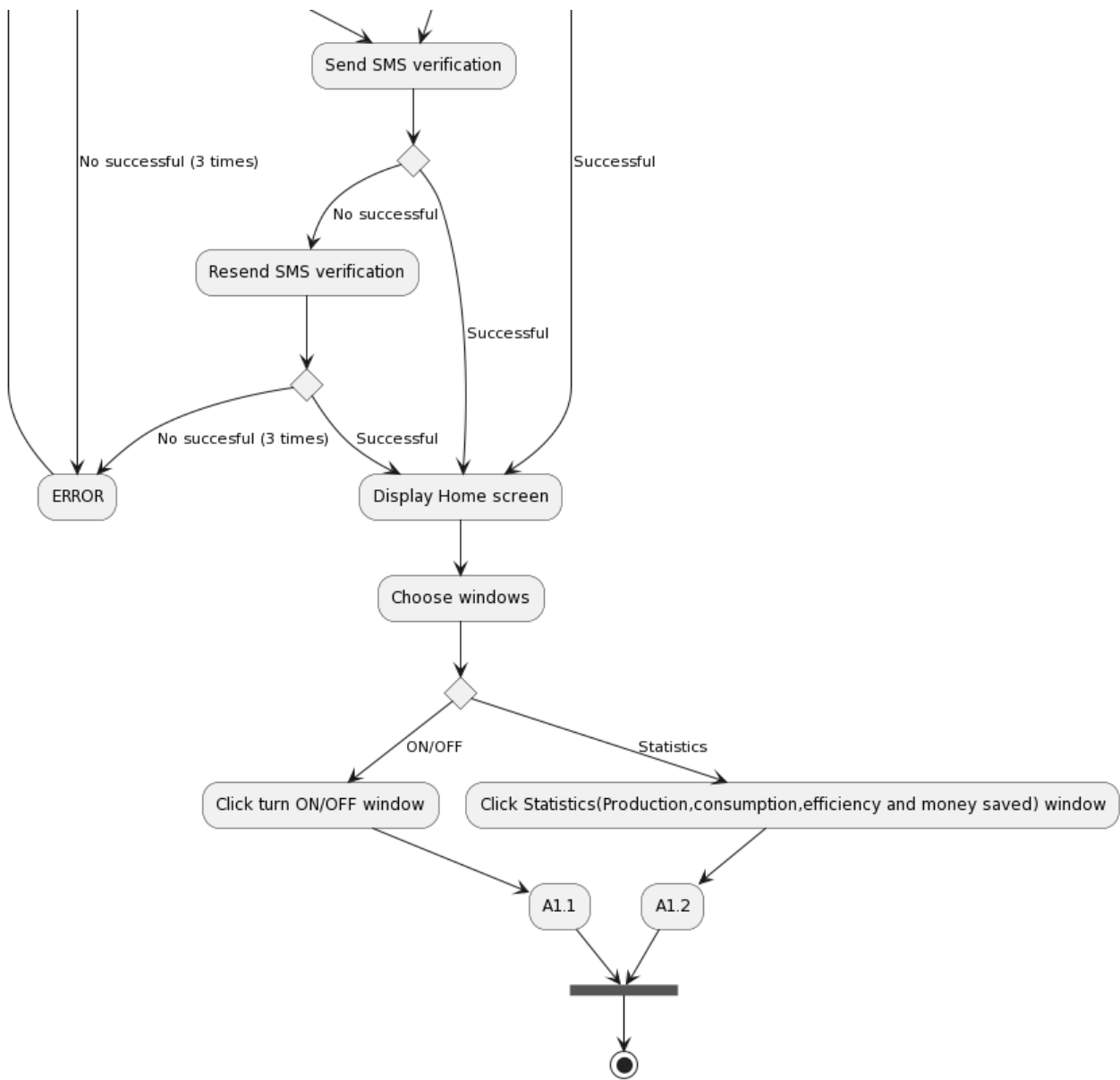
The arrows in the diagram depict the relationships and interactions between the actors and use cases. For example, "User" interacts with "Connect/disconnect panel", "Energy production and consumption display", "Warning generation", "Calculate savings", "On/Off racking", and "Current activation status". Similarly, other actors have their corresponding interactions and dependencies with the use cases in the system.

Overall, this UML code provides a visual representation of the actors involved and the use cases they interact with in a system. It helps to understand the functional requirements and interactions within the system, facilitating communication and analysis during the software development process.

## 6.3 Activity and Sequence diagrams

### A1.Interactions between users-application





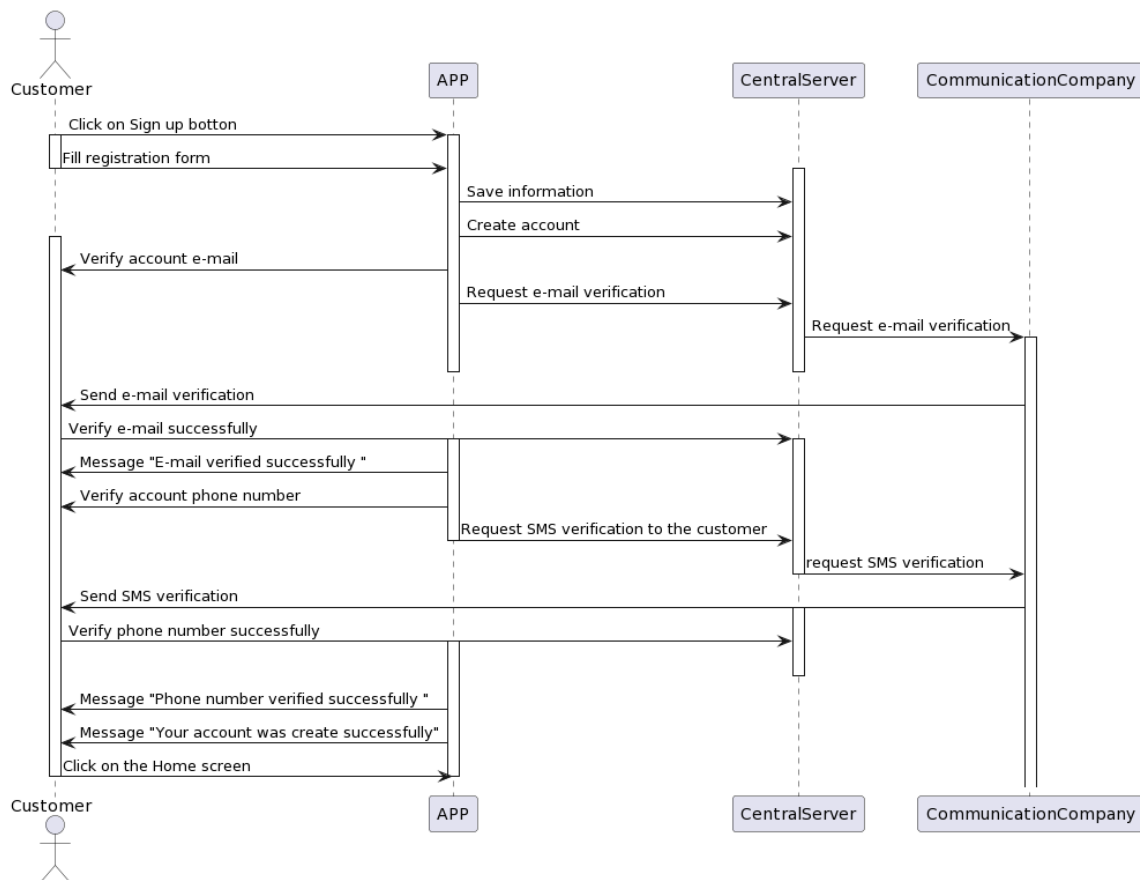
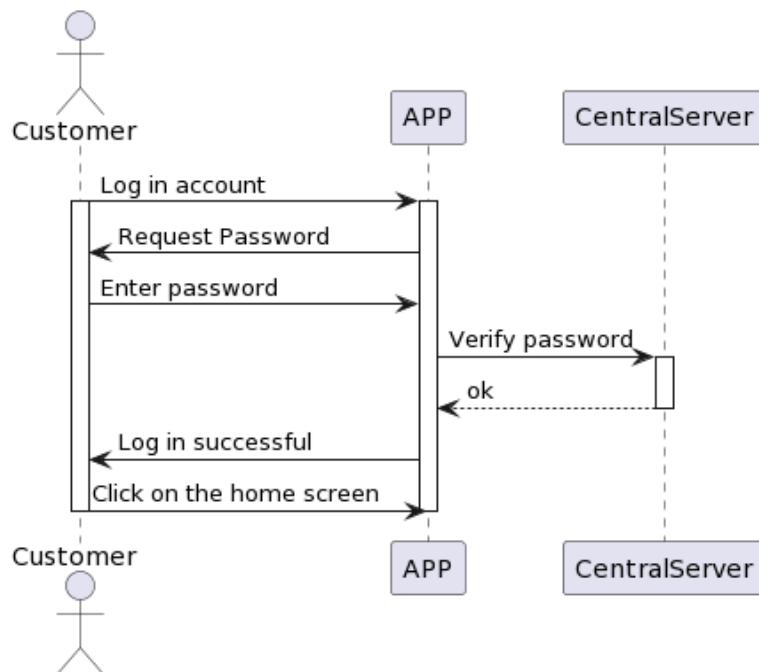
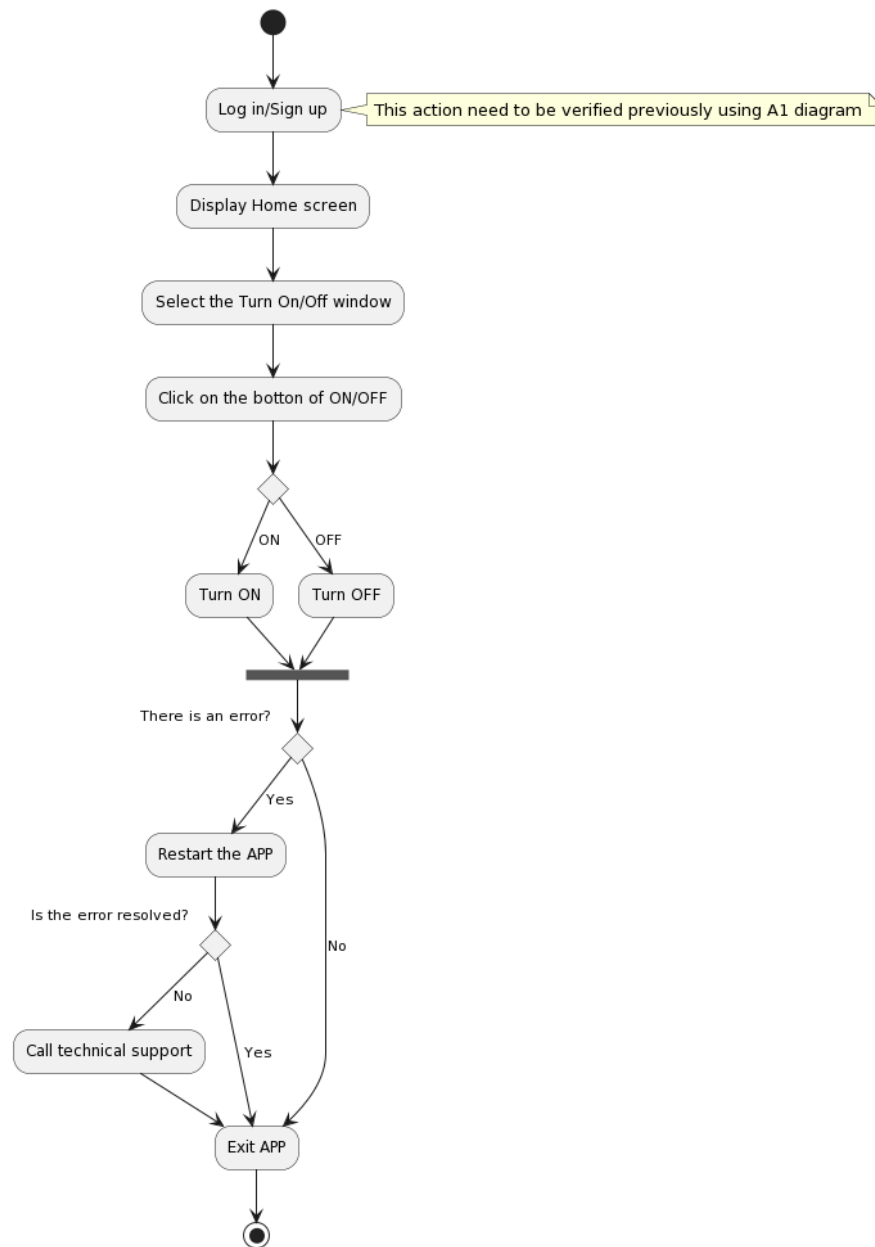
**S1: Sign up to the APP****S2: Log in APP**

Diagram depicting the interactions between users and an application. The diagram starts with an initial point and branches into a decision point regarding access to the app. If the user doesn't have an account, they are directed to the sign-up screen, where they fill out an account form, validate their email, and initiate a verification process. Depending on the success of the verification, the flow either proceeds to resend the verification (if unsuccessful) or moves to validate the phone number and send an SMS verification. If the SMS verification fails, there is an option to resend it, and if unsuccessful three times, an error state is reached. Successful SMS verification leads to the display of the home screen.

From the home screen, there is another decision point representing user actions. Clicking "turn ON/OFF window" leads to an additional step (A1.1) and a connection to another part of the diagram. Choosing "Statistics" opens a window displaying production, consumption, efficiency, and money saved (A1.2), followed by a connection and returning to the starting point.

If the user already has an account, they are directed to the login screen, where they check their account and enter the password. If the password is unsuccessful, they can opt for password recovery and proceed to a verification step. Successful password entry leads directly to the display of the home screen.

**A1.1 Turn ON/OFF**

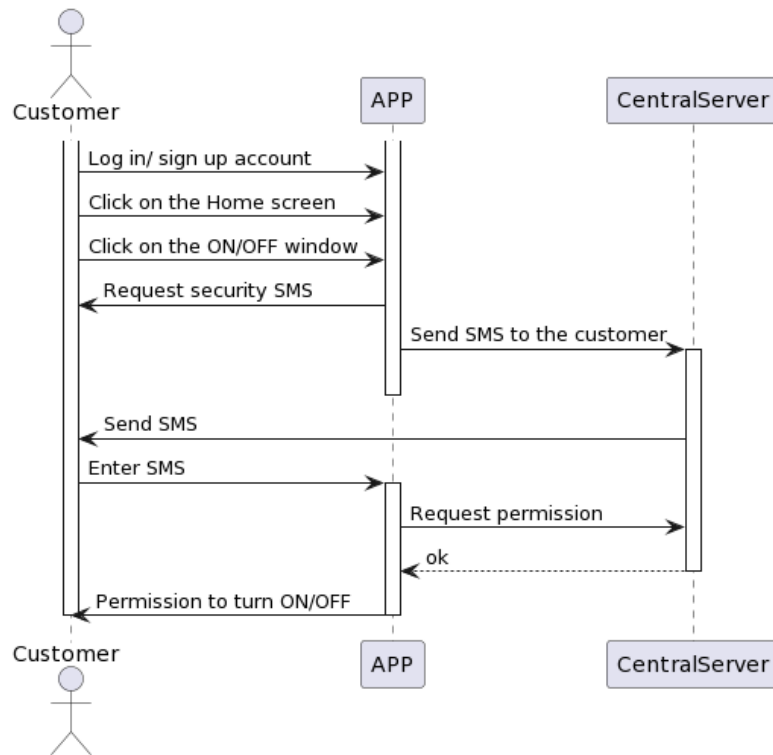
**S3: Turn ON/OFF through APP**

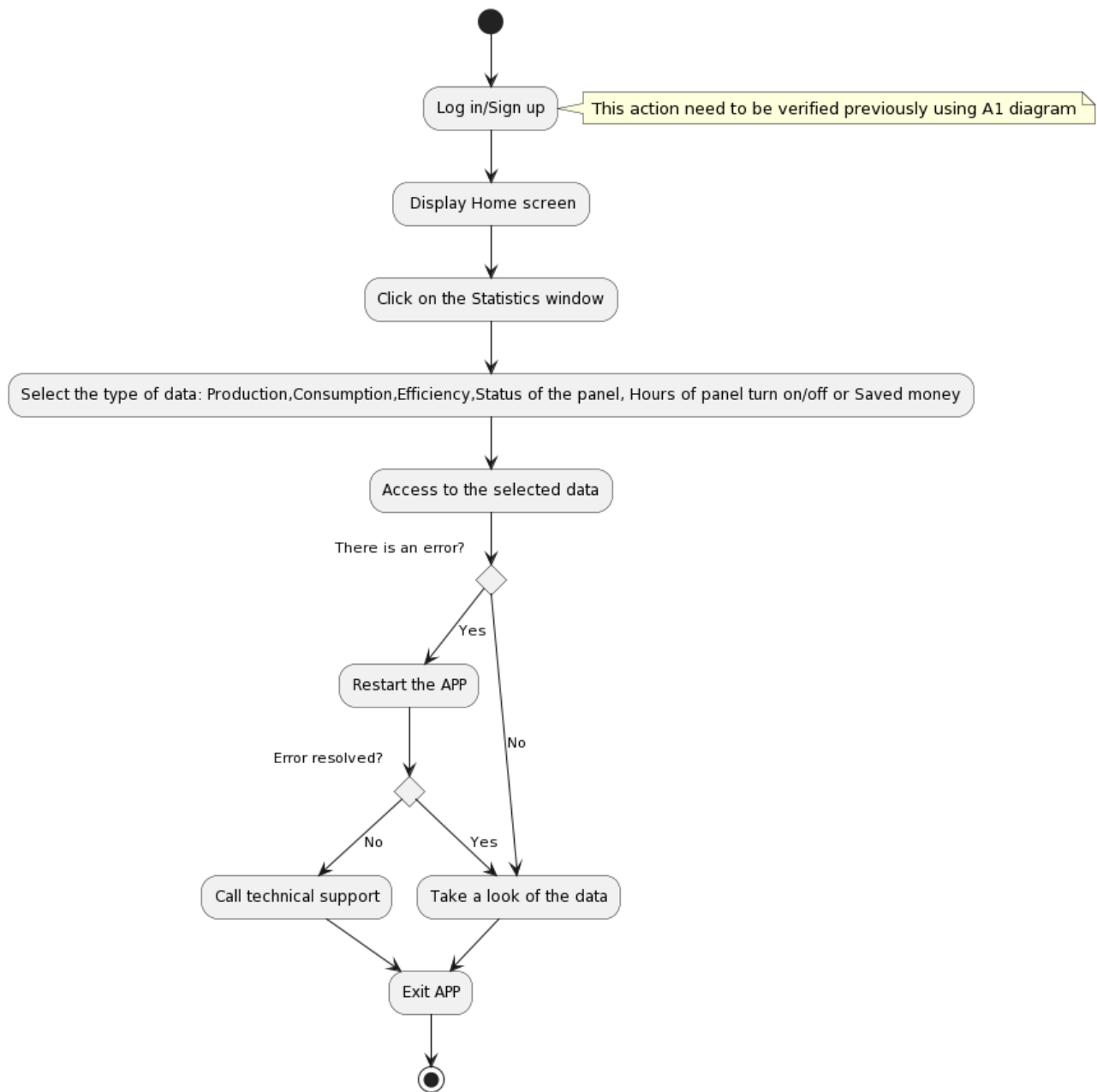
Diagram illustrating the process of turning ON/OFF a feature within an application. The diagram begins with an initial point and flows into a step labeled "Log in/Sign up." A note on the right indicates that this action must be verified beforehand using the A1 diagram, implying a prerequisite step. After logging in or signing up, the flow proceeds to the "Display Home screen" and prompts the user to select the specific "Turn On/Off" window. The user is then expected to click the ON/OFF button.

Once the button is clicked, a decision point is encountered, denoted by an empty condition. This missing condition represents whether the button was turned ON or OFF, requiring additional information to fully understand the diagram.

Within the ON branch, another decision point arises, asking if there is an error. If an error occurs (Yes branch), the flow leads to the "Restart the APP" step. If the error is resolved (No branch), the user can choose to either call technical support or exit the application. Whether the error is resolved or not, the diagram eventually returns to the initial point to potentially repeat the process.

Therefore, the diagram provides an overview of the process for turning a feature ON/OFF within the application, taking into account potential errors and the option to seek technical support.



**A1.2 Statistical data query**

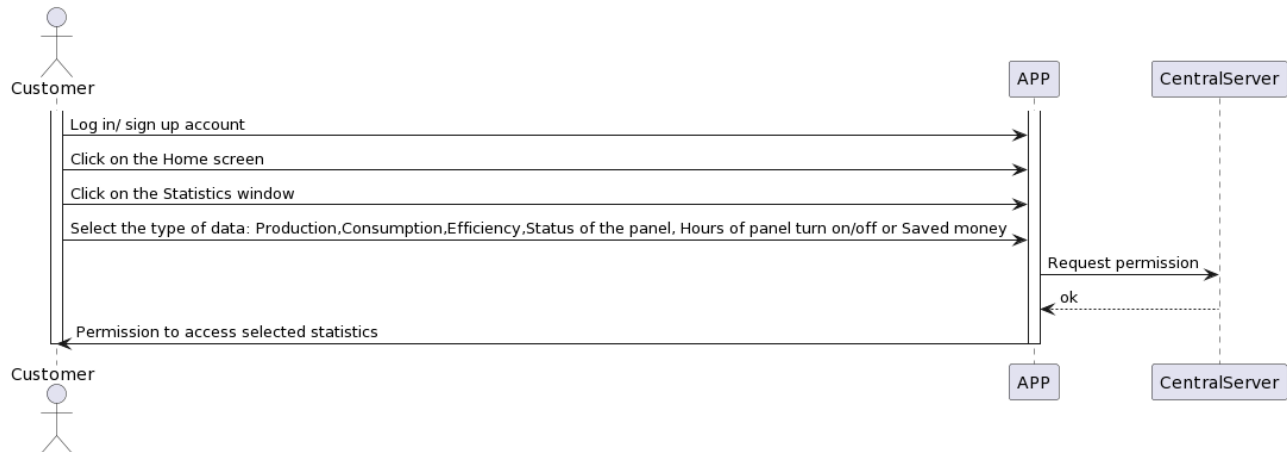
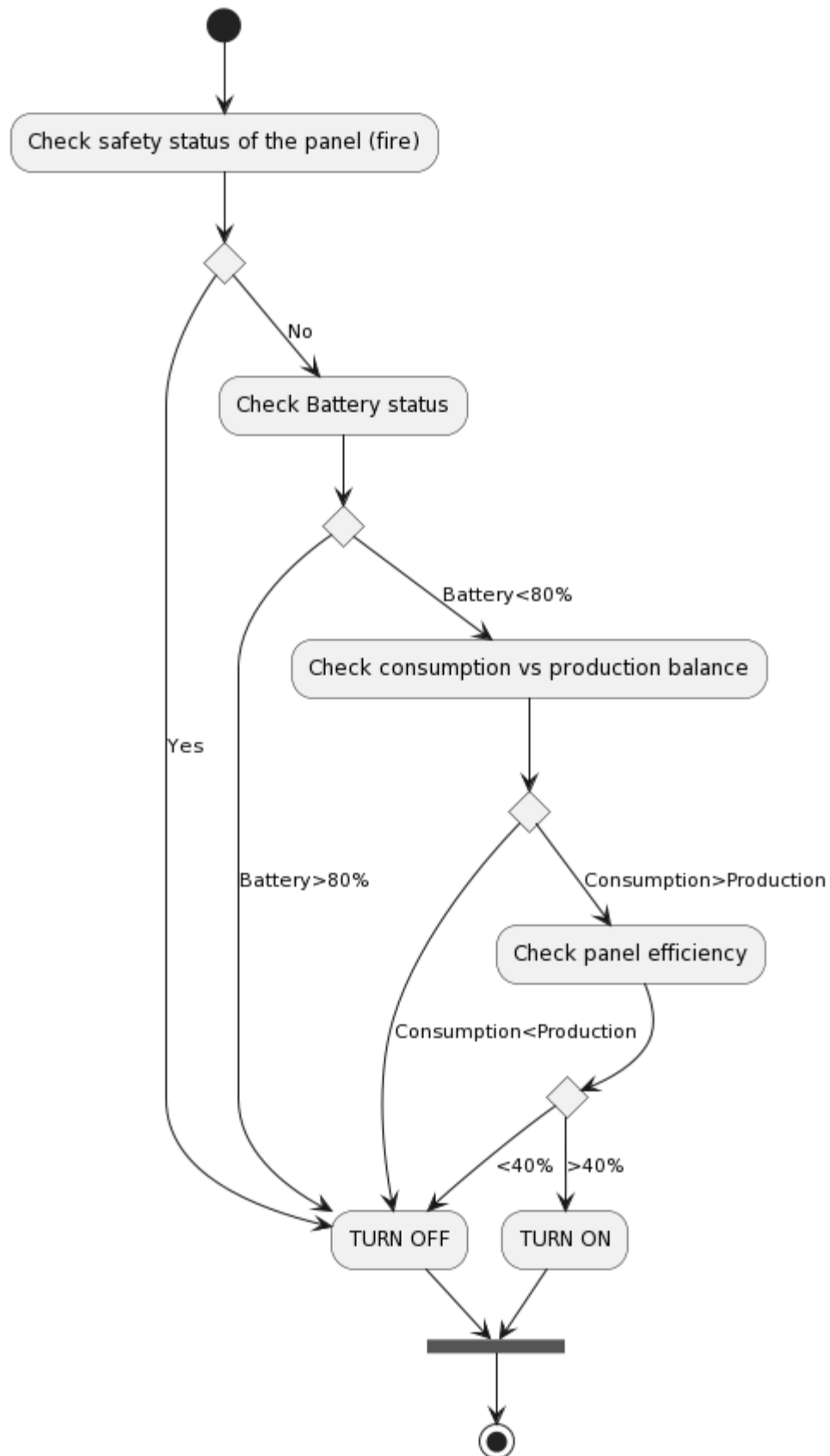
**S4: Access to the Statistics of the panel through APP**

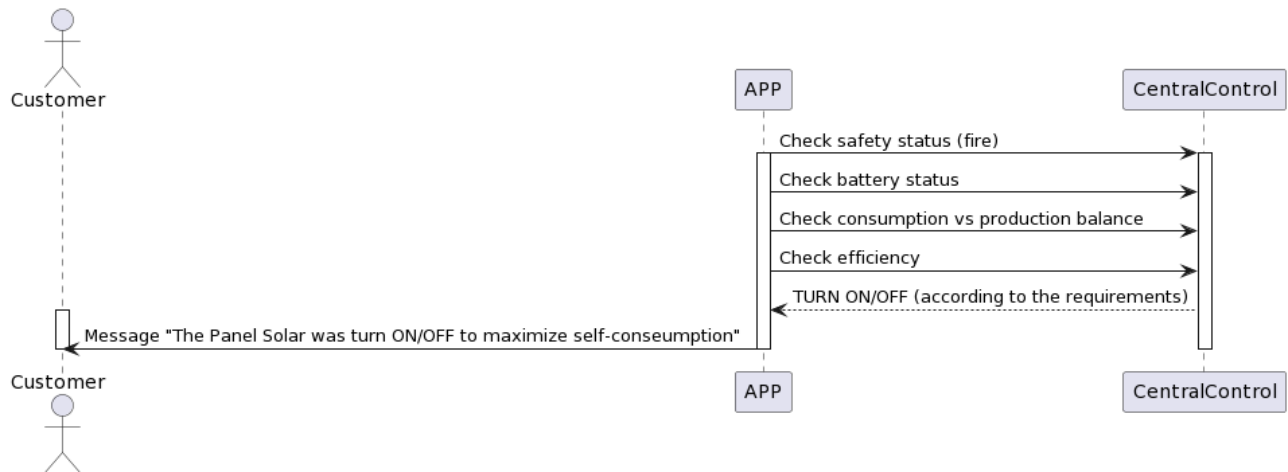
Diagram illustrating the process of querying statistical data within an application. The diagram starts with an initial point (\*) and proceeds with the "Log in/Sign up" step, which requires verification as stated in the note. Upon successful authentication, the flow leads to the "Display Home screen." From there, the user is prompted to click on the "Statistics" window, initiating the data querying process. The user is then expected to select the specific type of data they want to access, such as Production, Consumption, Efficiency, Status of the panel, Hours of panel turn on/off, or Saved money.

After selecting the desired data type, the flow continues to the "Access to the selected data" step. At this point, a decision point arises, represented by the question "There is an error?" This decision point determines whether an error occurred during the data query process. If an error is present, the flow branches to the "Restart the APP" step, indicating a potential need to resolve the error. Subsequently, another decision point appears, asking whether the error has been resolved. If the error persists, the user is advised to call technical support and exit the application (as depicted by the "Call technical support" and "Exit APP" steps). However, if the error is resolved, the flow proceeds to the "Take a look at the data" step, allowing the user to examine the queried statistical data.

If no error occurs during the data query, the flow bypasses the "Restart the APP" step and proceeds directly to the "Take a look at the data" step. From there, the flow continues to an unlabeled step called "data," suggesting that the user can view and analyze the queried statistical data. Finally, the diagram concludes with the "exit" step, representing the potential end of the process or the return to the initial point (\*) for subsequent interactions.

The diagram illustrates the process of querying statistical data within the application, with considerations for error handling and user verification. It provides a visual representation of the steps involved in accessing and examining different types of data.

**A2.Turn ON/OFF by the system**

**S5: Turn ON/OFF by the system**

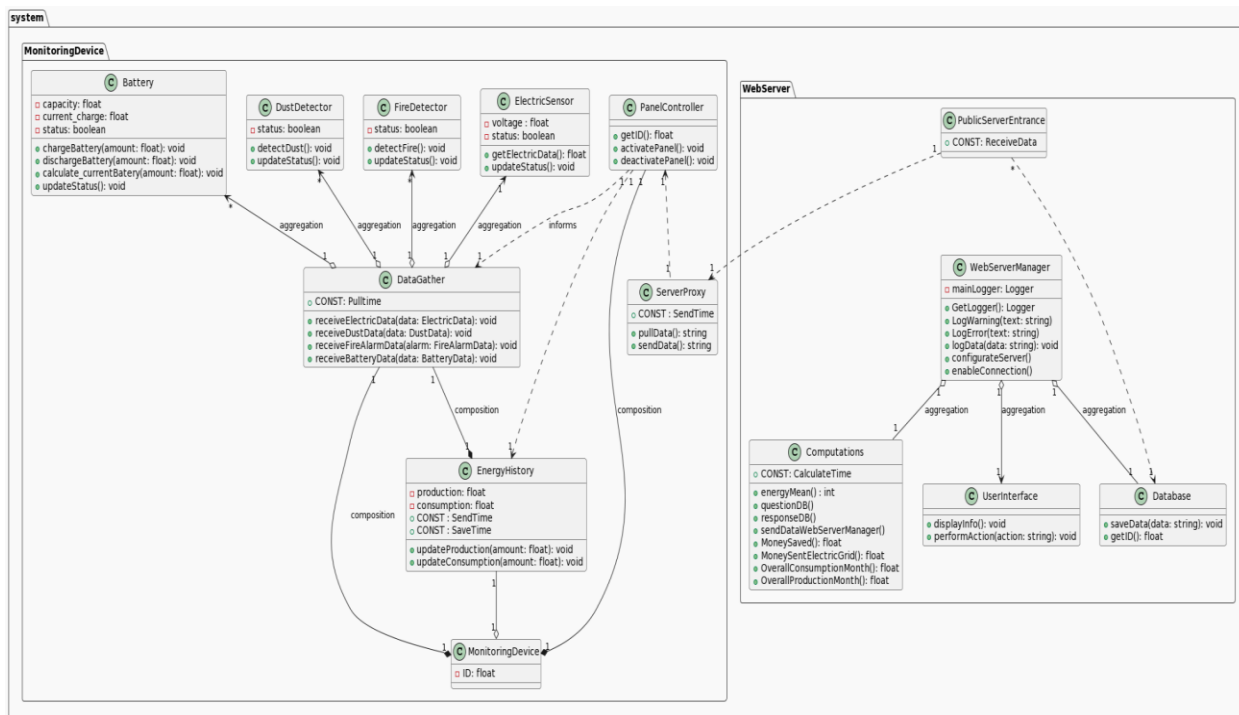
The diagrams from above represent a system's process for turning ON/OFF a panel based on various conditions. The system starts at the initial state represented by (\*), and the first step is to check the safety status of the panel, specifically for the presence of fire. If there is a fire, indicated by the "Yes" branch, the system proceeds to turn off the panel and transitions to the `==B1==` state. On the other hand, if there is no fire, indicated by the "No" branch, the system moves to check the battery status.

If the battery level is above 80%, the system turns off the panel and remains in the off state. However, if the battery level is below 80%, the system proceeds to check the balance between consumption and production. If the consumption is lower than the production, the system turns off the panel. Conversely, if the consumption exceeds the production, the system evaluates the panel efficiency.

If the panel efficiency is less than 40%, the system turns off the panel. If the efficiency is greater than or equal to 40%, the system proceeds to turn on the panel and transitions to the `==B1==` state. Finally, after completing the turning ON/OFF process, the system returns to the initial state (\*).

Overall, this UML code represents a decision-making process in a system that determines whether to turn ON or OFF a panel based on factors such as safety, battery level, consumption versus production balance, and panel efficiency.

## 6.4 Class diagram



We have designed a series of packages inside the whole class called system.

As we can see we have MONITORING DEVICE, and WEB SERVER, both packages are connected.

Now let's see in detail each of the packages and the different classes that compose them.

Regarding the MONITORING DEVICE:

- MonitoringDevice is the main class and it is composed of both PanelController (which will decide to activate or not the solar panel) and DataGather (which aggregates the information from all the different sensors we have seen in the architecture).
- Note that the energyHistory class will receive the gathered data every 30 minutes (for example) since it does not make sense to send the data every second to the PanelController. The only input that really needs to be taken into account in a matter of seconds is the fire alarm (for this reason, panelController receives reports from DataGather).
- Finally, Server Proxy will be retrieving data with its Monitoring Device ID constantly from the Panel Controller that will be sent to the PublicServerEntrance, a class of the package Web Server.

From the point of view of the Web Server Package:

- In this case we have 2 main classes, Public Server Entrance which is the bridge between both packages as well as Server Proxy from the Monitoring Device. And the Web Server Manager which organizes every action that must be performed.
- Public Server Entrance, will send all the information received to the Database where it will be saved and sent directly to the Web Server Manager.
- And Computations will perform all computations that must be done in order to show them via User Interface to the user in our Web App.