Energy Insight Hub

Remote Meter Reading and Data Visualization

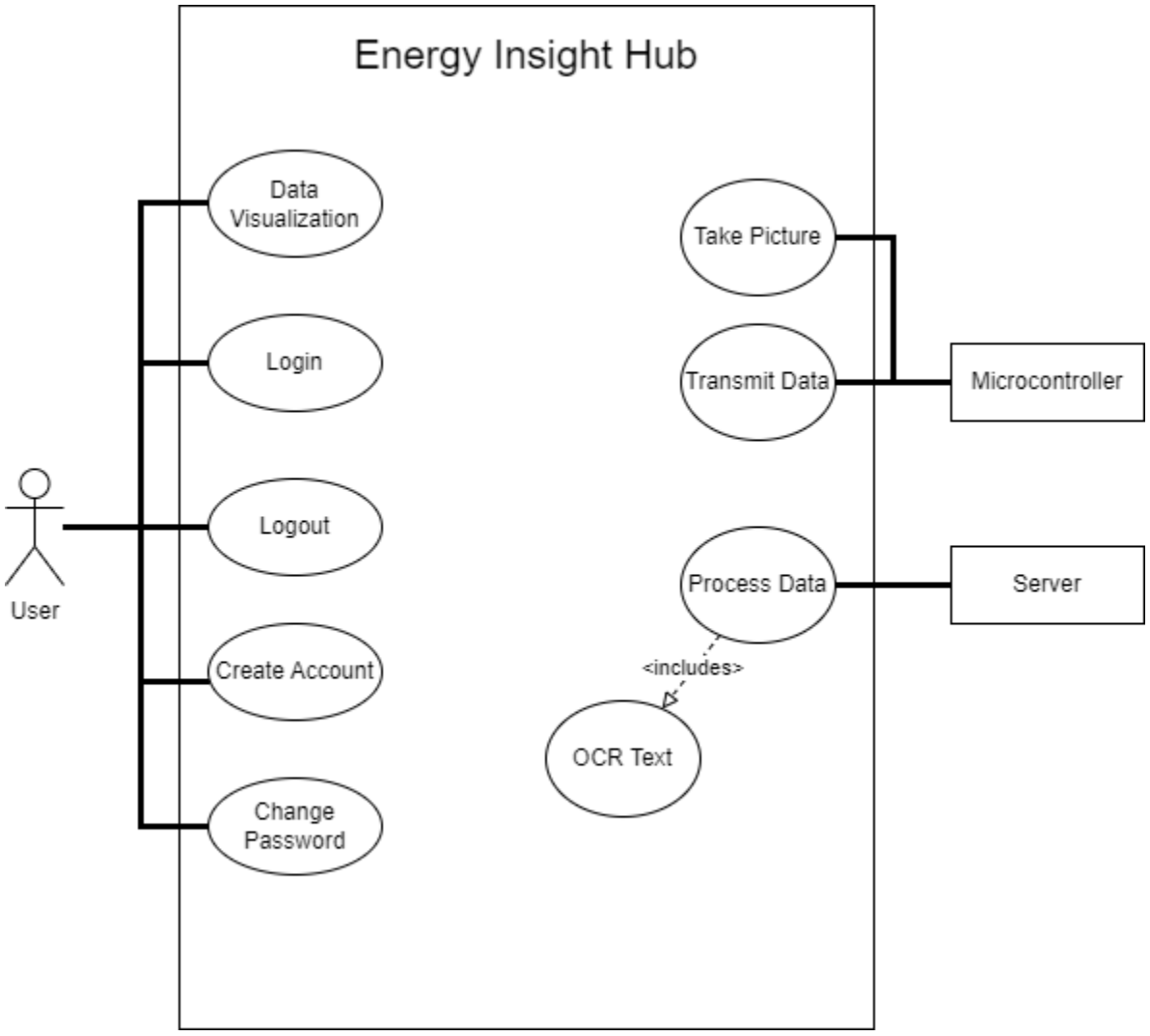
Name: Alexandre Paquette
Student Number: C00302989
Supervisor: Dr. Oisin Cawley
Date: December 22nd, 2023

Table of Contents

Executive Summary

The Energy Insight Hub is a comprehensive system designed to empower users with a convenient and insightful means of tracking their energy usage. The system employs a microcontroller to capture energy meter data at hourly intervals, and transmit the information to a remote server. The server processes the data using OCR technology, and stores it in a database. A web app interface allows the user to access and visualize their energy consumption conveniently.

Use Case Diagram



Microcontroller: The microcontroller facilitates the functionality of taking pictures and transmitting data at hourly intervals.

Server: The server is responsible for processing data, including OCR text extraction. It also hosts the web app the user will interact with.

User: The user interacts with the system through a web app, enabling data visualization, and account management.

Use Case	**Take Picture**
Actors	Microcontroller
Description	This use case describes how the microcontroller takes a picture of the energy meter at regular intervals.
Trigger	Interval timer runs out and triggers taking picture
Normal Flow	1. Hourly event triggers 2. Camera is initialized 3. Camera captures image of meter 4. File is readied for transmission a. Save image in appropriate format b. include timestamp and meterId in filename 5. Transmit image to server 6. Standby for next event
Alternate Flows	2a. Camera initialization fails 1. A log is sent to the server to notify the client 3a. Image capture fails 1. A log is sent to the server to notify the client 5a. Image transmission fails 1. Attempt to reconnect to network

Use Case	**Transmit Data**
Actors	Microcontroller
Description	This use-case describes how the microcontroller will transmit the picture taken from the picture use-case.
Trigger	Invoked by the take picture use-case
Normal Flow	1. Verify that data is ready for transmission 2. Establish connection with the server 3. Package the data for transmission 4. Encrypt data 5. Transmit data using internet protocol 6. Wait for acknowledgment that data was successfully transmitted
Alternate Flows	2a. Connection failure 1. Attempt Connection 2. Loop until successful 5a. Transmission Fails 1. Try transmission again 2. Loop until successful 6a. Acknowledgment Not Received 1. Try transmission again 2. Loop until successful

Use Case	**Process Data**
Actors	Server
Description	This use case describes how the server will process the data received from the microcontroller.
Trigger	Interval timer runs out and triggers data processing check
Normal Flow	1. Check for new data 2. Extract data from file a. Record meterId from filename b. Record DateTime from filename 3. Perform OCR on picture 4. Validate Data 5. Store record in database 6. Standby
Alternate Flows	1a. No Data is found 1. Go to standby 3a. Data extraction failed 1. Log Error 2. Notify User 4a. Data Format Error 1. Log Error 2. Notify User

Use Case	**OCR Text**
Actors	Server
Description	This use case describes how the text is extracted for the picture to be stored in a database and accessed by the user
Trigger	Invoked by the process data use-case
Normal Flow	1. Open Image 2. Perform image preprocessing 3. Extract Values 4. Validate Text 5. Convert to number value 6. Return value
Alternate Flows	NA

Use Case	**Visualize Data**
Actors	User
Description	This use case describes how the data is displayed to the user.
Trigger	User submits a data query request
Normal Flow	1. User initiates request to view data within a date range 2. System retreives data from database based on user request 3. Data is transformed into a suitable format for visualization 4. Generate visualization 5. Display Visualization
Alternate Flows	2a. No data found for the given range 1. Display message informing user 2b. Invalid range provided by user 1. Display message informing user

Use Case	**Login**
Actors	User
Description	This use case describes how the user will log into their account
Trigger	User provides username and passwords and clicks the login button.
Normal Flow	1. User enters login credentials 2. User clicks login button 3. User credentials are validated 4. Redirect to user dashboard
Alternate Flows	3a. User credentials are invalid 1. User is informed of invalid credentials

Use Case	**Logout**
Actors	User
Description	This use case describes how the user logs out of their account.
Trigger	User clicks the logout button
Normal Flow	1. User clicks logout button 2. Session is terminated 3. User redirected to login page
Alternate Flows	2a. If concurrent sessions are found, terminate all other sessions 2b. Session ends due to innactivity timeout

Use Case	**Create Account**
Actors	User
Description	This use case describes how a user would go about creating an account
Trigger	User clicks create account button
Normal Flow	1. User clicks on sign up button 2. System navigates to account creation form 3. User fills account creation form 4. User clicks on create account 5. System validates user data 6. System creates account 7. Send confirmation email
Alternate Flows	5a. User data fails validation 1. Inform user of invalid data 5b. User password is weak 1. Inform user of weak password 6a. Duplicate account is found 1. Inform user of duplicate account 6b. Failed to create account 1. Display error to user

Use Case	**Change Password**
Actors	User
Description	This use case describes how a user would change their password
Trigger	The user clicks “forgotten password” or “change password” button
Normal Flow	1. Navigate to password reset section 2. User provides email address 3. Send email verification 4. User clicks on verification link 5. New password is provided
Alternate Flows	3a. Incorrect email address 1. Display message informing user 3b. Email delivery failure 1. Display message informing user

FURPS

FURPS is an acronym used to define key characteristics and requirements of the system. The acronyms stands for functionality, usability, reliability, performance, and supportability. Each section will be detailed below concerning the energy insight hub system.

Functionality

This category encompasses the core features and capabilities that the energy insight hub system will provide. The core functionalities this system will provide are:

- Collect Energy Data on an hourly basis
- Transmit the data to a remote server
- Store the data collected
- Display the data to a client via a web app

Usability

This category refers to the user interface and overall user experience with the system. The main portion of the system this category pertains to will be the client web portal that allows them to view and interact with their data. However, it is also imperative that the data is displayed in a way that is useful to the user. The web portal will display data usage in a Data Grid where each row represents one day, and each column represents one hour. There will also be a column displaying the total energy usage for that day. The web app will also have options to display energy usage in charts that will be visually easy to understand for the client. The client will be able to query the database by providing a specific time range to view data usage within the range.

Reliability

This category involves the system's ability to perform consistently and accurately over time. With the energy insight hub, this means that the data collection, transmission, and decoding must be exceedingly reliable to deliver the best experience to the user. The system must be able to reliably take a picture of the meter at hourly intervals, transmit the image over the network to a remote server, and reliably extract the reading from the image so that it can be stored in a database. This also means that if there are any issues related to poor network connection, missing transmissions, or issues decoding the images, the user must be well-informed promptly so that they may recalibrate the hardware to minimize the amount of system downtime.

Performance

This category defines requirements of how well the system must perform in terms of speed, response time, and efficiency. With the energy insight hub, this means that the web app must be responsive to user inputs with little load times. The user should not wait more than 10 seconds for large datasets to load into the data grid, and preferably no more than 1 second when loading 1 month's worth of data. The user should also be able to interact with the user interface fluently without any input delays outside of large data processing.

Supportability

This category encompasses the ease with which the software can be maintained, upgraded, and supported over time. For the energy insight hub, the web app will be supported on the biggest browser platform, chromium, and will be designed to be responsive to mobile layout.

Making it a web app means the system will be accessible through any device that supports a web browser, which includes the likes of desktops, laptops, phones, and tablets. The web app will also be built using dotnet 8 Blazor, which is a long-term support platform and will allow for future upgrades in the long term.

Metrics / Success

- The system will be able to take a picture of the energy meter
- The system will be able to transmit the picture to a remote server
- The system will be able to extract energy readings from the picture
- The system will be able to store the readings in a database
- The system will be able to display the readings to a client in a web app