TED UNIVERSITY
HIGH LEVEL DESIGN PROJECT

CMPE 491
SENIOR DESIGN PROJECT 1

Begüm Akdeniz 16669037806
Umut Yıldırım 10822781650
Ceren Büyükgüllü 27142510204
İrem Tamay 59701366816

TABLE OF CONTENT

1.Introduction

Faceguard, an advanced facial recognition technology, will be implemented at campus entrances of Ted University to enhance security and streamline access control. The system utilizes biometric analysis to identify and verify individuals based on their unique facial features, including the shape of the nose and the distance between the eyes. By capturing and analyzing facial images or videos, Faceguard authenticates authorized persons such as students, staff, and authorized guests by matching their facial traits against a database of recognized faces.

The primary goal of adopting facial recognition technology at campus entrances is to bolster campus security, prevent unauthorized access, and ensure the safety of students and staff. The system eliminates the need for physical identification cards, simplifying access control procedures. As employees and students enter the building, their faces are scanned and verified against the database, reducing the likelihood of access credentials being forgotten or stolen and ensuring that only authorized individuals can enter the school grounds.

To prioritize privacy and data protection, explicit policies will be established. Safeguarding the acquired data and complying with relevant laws and regulations are essential to maintain trust and transparency within the university community.

This report provides a brief description of the Faceguard project, outlining its purpose, functionality, and the measures taken to protect privacy and data security. The architectural design patterns of the FaceGuard project are mentioned. Apart from this, there will be subsystems and hardware/software mapping diagrams to be found in the architecture. How to manage data, how to ensure security and access control system will be mentioned. The restrictions in the FaceGuard project (KVKK etc.) The measures taken against these restrictions will be mentioned. Subsystems will be shown in detail through diagrams. The presentation must include an initial demonstration of the intended product of the project.

1.1 Purpose of the System

Securing university entrances is FaceGuard's primary objective. It will improve security because it will scan each person's face as they enter the campus. The mechanism makes sure that admittance to the university is quick and precise due to the significant amount of students. Although human power will decrease, reliability will rise. Instead of using cards, Face Guard will identify the faces of students and personnel, prohibiting unwanted foreigners from entering the campus. This will speed up the login procedure and lowers the possibility of access credentials being lost or stolen.

1.2 Design Goals

The design goals for face recognition system are efficiency, maintainability, privacy and usability.

Efficiency: The face recognition system must to be effective and rapid at performing recognition tasks. Since using face recognition would produce a quicker result than using a card, we can deduce that the face recognition system is effective.

Maintainability: The ability of the system to be easily maintained, updated, and repaired is referred to as maintainability. It includes creating the face recognition system to make routine maintenance duties simpler and to demand less work to fix problems or make enhancements.
Privacy: Building trust between the user and the system is crucial since the face recognition system needs to protect users' private information. Face recognition systems should employ suitable data protection mechanisms to address privacy issues.

Usability: The face recognition system ought to be easy to use, with an intuitive interface that makes it simple to add people's faces to the database and provide transparent feedback throughout recognition operations. Additionally, it should seamlessly interact with current infrastructure and be interoperable with a variety of devices, including cameras.

1.3 Definitions, Acronyms, and Abbreviations

Face Recognition System: FaceGuard is defined as a technology that can compare human faces in an image from a digital photo or video frame to an identity kept in a database of faces.
Biometrics: The measurement and analysis of specific physical characteristics such as facial features.

SQL: The term stands for Structured Query Language (SQL), a language used for database queries. This language demonstrates how to query such tables as well as other objects, allowing you to interact with table data.

Acronyms and Abbreviations:
FR: Face Recognition
AI: Artificial Intelligence
ML: Machine Learning

1.4 Overview

Face Guard is an innovative technology aimed at enhancing security standards through the use of artificial intelligence and machine learning. Its primary objective is to improve identity verification by recognizing faces and analyzing facial features. By capturing and analyzing pictures or videos of individuals' faces, Face Guard facilitates easier, quicker, and safer access to institutions.

Key Features:

Facial Recognition: FaceGuard utilizes advanced algorithms to identify and authenticate individuals based on facial features such as the shape of the nose and the distance between the eyes. This technology replaces the need for identification cards, minimizing the risk of unauthorized entry into the campus.

Functionality: The system performs four main tasks: scanning, recognizing, identifying, and authenticating faces. Firstly, it scans the face of a subject when they look into the camera, capturing a photograph or video stream. Secondly, it searches a database to recognize the face. Thirdly, it identifies the user based on the recognized face. Lastly, it authenticates the face by comparing it with the faces stored in the database, allowing access only to recognized individuals.

Restrictions and Considerations:

Financial Restrictions: The project faces budgetary constraints, as the purchase of a system kit may impose financial burdens on the team members. However, as the system is intended for use within the university, IT support has been sought as a means to overcome this challenge.

Scheduling Restrictions: The project may need to adhere to specific timelines or deadlines due to operational or organizational considerations. It is essential to address any time-related restrictions to ensure successful implementation.

Legal Obligations: Compliance with legal requirements and data protection regulations must be a priority when implementing the Face Guard system. This includes ensuring appropriate consent and safeguarding the privacy of individuals whose data is captured and processed.

3 Proposed Software Architecture:

The FaceGuard project follows the Model-View-Controller (MVC) architectural design pattern to provide advanced face recognition capabilities for enhanced security. The subsystems in this architecture consist of the Model, View, and Controller.

1) The Model subsystem is responsible for performing face detection, feature extraction, and recognition. It maintains a face database sourced from the university IT department's database and integrates with access control systems like campus doors and tourniquets.

2) The View subsystem represents the client-side application, allowing users to capture facial images, manage profiles, and control access settings. It provides a user-friendly interface for interacting with the system.

3) The Controller subsystem acts as the intermediary between the Model and View. It handles user input, triggers appropriate actions in the Model, and updates the View accordingly. The Controller ensures the separation of concerns and facilitates the flow of data and logic between the Model and View.

The project focuses on performance, security, and user-friendliness. It aims to achieve efficient identification and access control while reducing reliance on traditional access cards in the university environment. Thanks to the FaceGuard, no one from outside the university can enter the campus.
This concise overview highlights the utilization of the MVC architecture, the presence of the Model, View, and Controller subsystems, key functionalities, and the goals of the FaceGuard project.

3.1 Overview:

The FaceGuard project follows the Model-View-Controller (MVC) architectural design pattern to provide advanced face recognition capabilities for enhanced security. The subsystems in this architecture consist of the Model, View, and Controller.

The Model subsystem is responsible for performing face detection, feature extraction, and recognition. It maintains a face database sourced from the university database and integrates with access control systems like school doors and tourniquets.

The View subsystem represents the client-side application, allowing users to capture facial images, manage profiles, and control access settings. It provides a user-friendly interface for interacting with the system.

The Controller subsystem acts as the intermediary between the Model and View. It handles user input, triggers appropriate actions in the Model, and updates the View accordingly. The Controller ensures the separation of concerns and facilitates the flow of data and logic between the Model and View.

The project focuses on performance, security, and user-friendliness. It aims to achieve efficient identification and access control while reducing reliance on traditional access cards in the university environment.

This concise overview highlights the utilization of the MVC architecture, the presence of the Model, View, and Controller subsystems, key functionalities, and the goals of the FaceGuard project.

3.2 Subsystem Decomposition:

Our FaceGuard project is designed following a MVC Architecture model, consisting of two subsystems: Client Application and Server Application.

1) Model Subsystem:

The Model subsystem is responsible for maintaining the domain knowledge and encapsulating the business logic of the FaceGuard system. It includes functionalities such as face detection, feature extraction, recognition algorithms, and user profile management. The Model subsystem is independent of the user interface and handles the core processing of facial data.

2) View Subsystem:

The View subsystem focuses on the presentation layer and user interface components of the FaceGuard system. It includes functionalities related to capturing facial images or videos, displaying recognition results, managing user profiles, and access settings. The View subsystem communicates with the Model subsystem to retrieve data and update the user interface accordingly.

3) Controller Subsystem:

The Controller subsystem acts as an intermediary between the Model and View subsystems. It handles user interactions, interprets user input, and triggers appropriate actions in the Model or View. The Controller subsystem receives input from the View, processes it, and updates the Model or View accordingly. It ensures the separation of concerns by decoupling the user interface from the core logic.

In this subsystem decomposition, the View subsystem handles the user interface and presentation of the FaceGuard system. The Controller subsystem manages user interactions and controls the flow of data between the View and Model subsystems. The Model subsystem encapsulates the core face recognition logic and data management.

The use of the MVC design pattern promotes separation of concerns, modularity, and maintainability in the FaceGuard project. It allows for easier updates or modifications in each subsystem independently without affecting the others, facilitating code reuse and enhancing the overall system design.

3.3 Hardware/Software Mapping

The project has many hardware requirements. These can be specified as cameras, gateways, processing unit, storage, display screen and network infrastructure.

Cameras: Cameras are required to read the faces of people who want to pass through the door.

Gates: The existing school entrance gates will be used.

Processing Unit: A powerful computer or server is required to process image processing and face recognition algorithms.

Storage: Sufficient storage capacity is required to store facial data and related information of persons for identification purposes.
Network Infrastructure: It is the connection established to provide communication between cameras, processing unit and other components of the system.
Display Screens: Screens where people who want to pass through the door can see their data.
NVIDIA Jetson: It can be used to increase performance in face recognition systems and to perform real-time operations. Thanks to its high processing power, it can detect faces quickly and effectively. It can be used to detect faces on images from the camera. It can perform recognition by comparing it with the previously saved face data and match the face with the identification information. Thanks to its high-speed image processing and analysis capabilities, you can get instant results and make quick decisions.

It has low power consumption.

Face identification algorithms, Image/Video Processing Software, Database Management System, User-Interfaces, Security Protocols will be used as software components of the project.

Facial Recognition Algorithms: Machine learning and artificial intelligence algorithms are used to analyze facial features and identify individuals.
Image/Video Processing Software: They are software tools to be used to process images or videos captured by cameras and extract facial features.
Database Management System: A system for storing and managing facial data and information held in the school system for quick access during identification.
User Interface: As previously mentioned in mock-ups, there will be user-friendly user interfaces with the colors of the school, where the data of the users will appear on the screen.
Security Protocols: Encryption and secure communication protocols will be in place to protect the confidentiality and integrity of Face data and system functionality, which is also legally important and protected within the framework of KVKK.

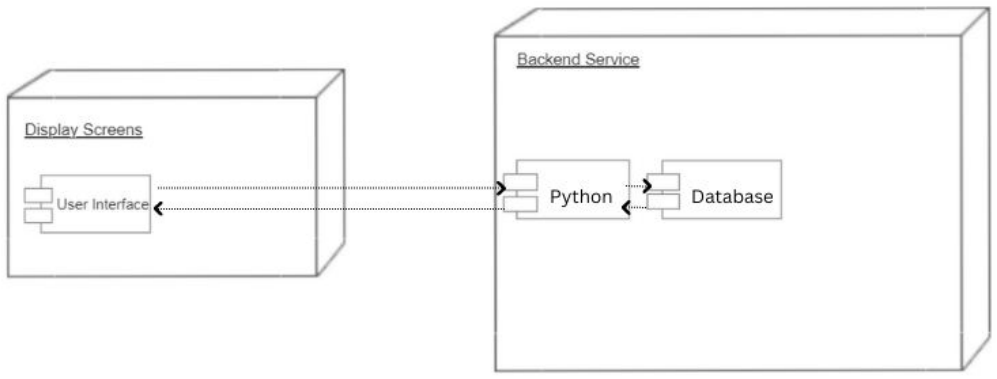


Figure 1: Mapping of Application

3.4 Persistent Data Management

Persistent data management is essential for the Face Guard project to store and use face data and related information.
Here are some considerations for managing persistent data in this project:
Database Management System (DBMS): mySQL will be used in the project as a suitable DBMS that can efficiently process large amounts of data and provide fast access.
Data Storage: A storage is required to keep the information of the users when they enter and exit the system and to record the entry/exit data of the school. At the same time, it is important for the safety of the school that the faces and videos of the people entering and exiting are recorded.

3.5 Access Control And Security

A strong authentication mechanism is implemented in the FaceGuard project to authenticate users before granting access. The identity information of the users in the school and their biometric visuals are used.
Since the data to be used in the project includes biometric data, the use and storage of users' information include KVKK clauses. VKK has provisions regulating the processing and

protection of biometric data. Because biometric data includes unique data based on people's physical or behavioral characteristics. In Article 3 of the KVKK, it is stated that biometric data is personal data of special nature. Therefore, stricter regulations and safeguards are required for the processing of biometric data. KVKK foresees a processing obligation based on legal grounds such as express consent, legitimate interest or legal obligation in the processing of such data. There are obligations such as taking the necessary technical and organizational measures for the security of personal data, reporting data breaches and protecting the rights of data subjects.
There will be a Role Based Access Control mechanism to define different user roles and assign appropriate permissions to the roles according to their specific responsibilities and access needs. For example, a school janitor must have a larger reach than a school student, who must have permission to specific areas they need to access.
The school's security teams and departments such as IT will be able to access the information of people entering and leaving the school. other than that, there will be no open source. When necessary, certain teams will follow these entry-exit activities in accordance with security rules and authorized persons will be able to control and change the information.

3.6 Global Software Control:

Our project is an event-driven system project, because in event-drvien systems, it is necessary to trigger the action and continue the flow of the system. Components and modules must respond to events and act accordingly. In our project, the camera receives data and triggers events for students to log in. Triggered events are received, status is updated after checks are provided. If the faces of the users are recognized, the turnstiles are opened, if not, they are not opened and a notification is provided. Event-driven systems provide flexibility to add new events or modify existing behaviors. For example, new authentication methods or additional states and transitions may be added depending on future requirements.

3.7 Boundary conditions:

1. Hardware Limitations:

The FaceGuard System must have the appropriate hardware components. This includes factors such as using a camera with sufficient resolution and meeting processing power and memory requirements.

Data Storage: There is a need for a storage area that holds user data, information and transition information from turnstiles. Data storage is important to ensure security and access fast data.

Parallel Processing Capabilities: Ted University has 4 turnstiles, it should be possible to pass at the same time. Therefore, it must be capable of parallel processing. This increases system performances and makes intensive calculations easier.

Processor Power: The FaceGuard system contains complex mathematical operations and algorithms. It needs a processor that provides high performance to do these operations.

Memory: Recorded images need to be stored for a while. In addition, sufficient memory is required for processing data. Especially processing high resolution data increases memory usage. Considering this, the memory should be calculated according to the number of users.

2. Photo conditions:

FaceGuard System needs suitable photo conditions for the face recognition algorithm to produce accurate and reliable results. Adequate and consistent lighting should be provided. Students' photographs should have the necessary lighting and lighting. It is important that their faces are clear and prominent, and their ears are visible. Adequate lighting is important so that the face recognition algorithm can capture the right features and optimize the recognition process. Insufficient light can prevent the algorithm from producing accurate results. In photographs, attention should be paid to shadows and glare. It should ensure that sufficient image quality is obtained. Photos must be of good quality.

3. User Records:

The system must have appropriate user registration processes to accurately record users facial data. The recorded face data will be used in the authentication process and must be stored in accordance with the privacy policies. User registrations are made by recording passport photos and information when students or employees are first registered to the school.

4. Processing Time:

Optimize users' recognition and transition times. User recognition time should be kept to a minimum and the transition process should be swift and seamless.

5. Database Updates:

Database operations such as adding, removing or updating users should be managed with appropriate methods to ensure that the system has access to real-time and up-to-date data.

6. Data Security:

The FaceGuard System should have appropriate data encryption and protection mechanisms to ensure the security of users face data. It is important that data is protected against unauthorized access and processed in accordance with privacy standards.

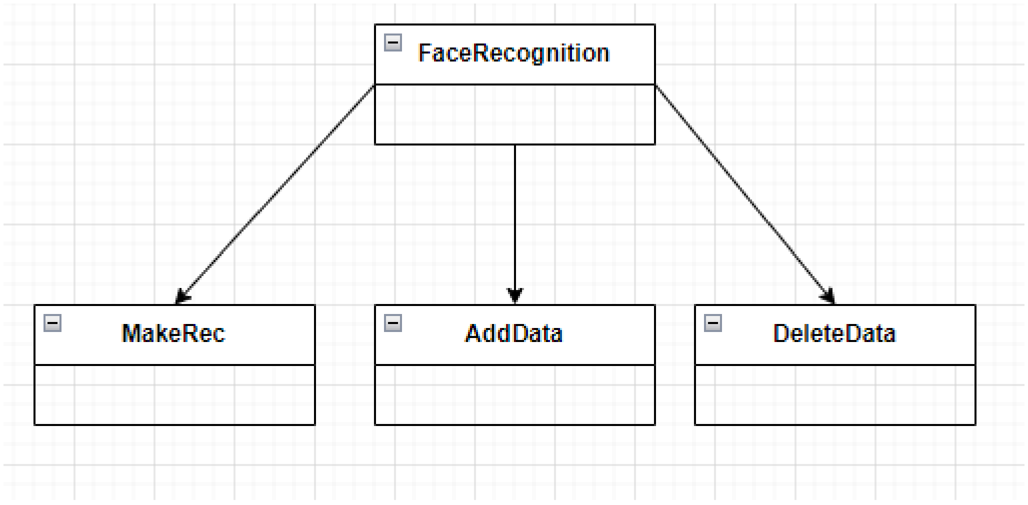
7. Failure:

In the event that one or more of the hardware components fail, the system must still be able to operate safely and reliably, or measures must be taken, such as automatically detecting and replacing faulty components. When software bugs or malfunctioning components are detected, software bugs should be fixed and faulty components should be automatically rebooted or reconfigured. In the event of power outages or failure of power supplies in the system, the system must take appropriate measures to prevent data loss during a power outage and to shut down safely. Transactions interrupted during the outage must be completed and data must be stored securely. During the interruption, students must pass through security with their “Tedu Portal” information.

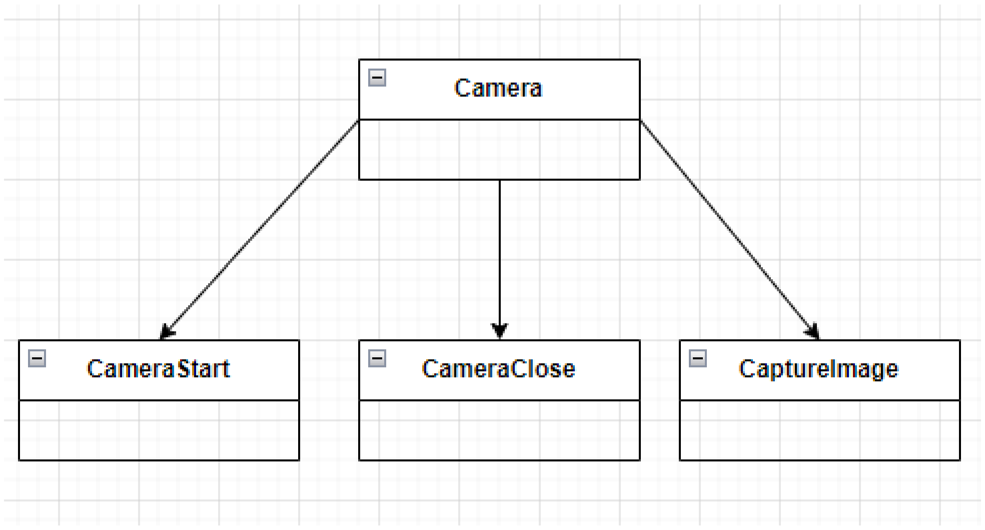
4. Subsystem services:

For software development, we decided to use the MVC design pattern because the MVC design pattern is easier to bug fix in complex applications and provides a nice environment for teamwork. We save time and error.

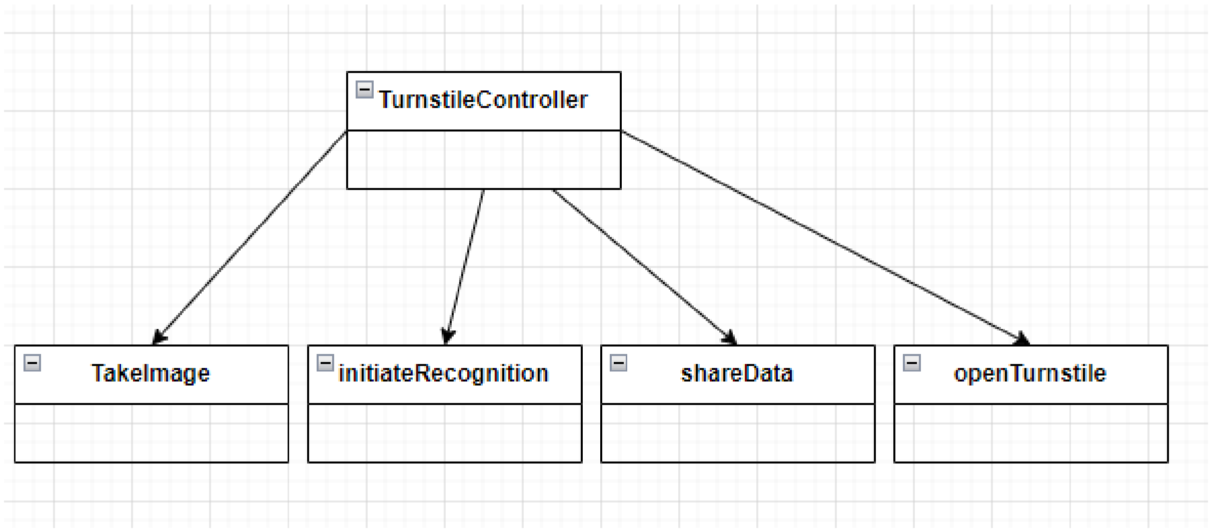
Model: Performs face recognition operations. A class should be created that compares facial recognition data with the database. The generated class processes the images and returns the recognition results.
MakeRec:It is a function that performs face recognition. This function processes the incoming image, compares it with the face data in the database and returns the recognition results.
AddData:It is a function used to add a new face data to the database. This function extracts face data from the incoming image and adds it to the database.
DeleteData: It is a function to delete a specific face data from the database. This function finds and deletes a specified face data in the database.



View: Data on user faces are taken with camera images, components should be created to display the faces of users. They should be able to project the camera image onto the screen and visually report the results.
CameraStart : It is a function that starts the camera. Turns on the camera source and makes it ready for image capture..
CameraClose : It is a function that stops the camera. Turns off the camera source and stops capturing images..
CaptureImage : It is a function that captures an image from the camera. This function takes a snapshot from the camera and can be used to transmit it to rendering or another component.



Controller: A controller class must be created. This class should manage interaction with the camera. It should receive the image from the camera and initiate face recognition. It should share the data it reaches and take actions by transferring information to the necessary places.
TakeImage: It is a method used to get the image from the camera.
initiateRecognition: It is a method used to initiate the face recognition process.
shareData: It is a method used to share the obtained data.
openTurnstile: It is a method used to open the turnstiles.



1. User Interface Subsystem:

- Saves user information.
- Retrieves the passport photo and video data from the camera.
- Suppresses facial recognition results.

2. Camera Subsystem:

- Provides camera images to provide user's facial data.
- Sends face data to store.
- It provides high resolution and clear image.

3. Face Recognition Subsystem:

- Processes data stored in the camera.
- It analyzes people by applying face recognition algorithm.
- It compares the faces of the users whose analysis has been completed with the previously recorded data and makes a match.
- It stores the paired faces and forwards them to the subsystems.

4. Turnstile Control Subsystem:

- The user opens the turnstile with the match notification.

- It does not grant passthrough to matched users, nor does it grant passthrough to unmatched users.
- It provides automatic opening and closing of turnstiles.

5. Database Subsystem:

- It stores users' facial data and information.
- It records recognized face data and unidentified face data.
- It checks for updates and backups.

Glossary:

- *FR:* Face Recognition
- *AI:* Artificial Intelligence
- *ML:* Machine Learning
- *KVKK (Kisisel Verilerin Korunmasi Kanunu):* The Personal Data Protection Law in Turkey, which regulates the processing and protection of personal data, including biometric data.
- *Architectural Design:* Architectural design refers to the process of defining the overall structure and organization of a software system.
- *Database Management System (DBMS):* Software system, such as mySQL, used to efficiently store and manage large amounts of facial data and information.
- *Face Identification Algorithms:* Machine learning and artificial intelligence algorithms used to analyze facial features and identify individuals.
- *NVIDIA Jetson:* A hardware device used to enhance performance in face recognition systems and enable real-time operations. It provides high processing power for quick and effective face detection and recognition.
- *MVC Design Pattern:* The Model-View-Controller (MVC) design pattern is a software architectural pattern commonly used in building user interfaces. It separates the application into three interconnected components.
- *Display Screens:* Screens where individuals can view their data and information when attempting to pass through the door.
- *Subsystem*: In software architecture, a subsystem refers to a self-contained and cohesive part of a larger system. It represents a functional unit within the system that performs specific tasks or provides specific capabilities.

References:

- Dictionary.com. (2021). What’s The Difference Between Acronyms vs. Abbreviations? In Dictionary.com. https://www.dictionary.com/e/acronym-vs-abbreviation/

- Types of Software Architecture Patterns. https://www.geeksforgeeks.org/types-of-software-architecture-patterns/

- NVIDIA Jetson Nano Developer https://openzeka.com/urun/nvidia-jetson-nano-developer-kit/

- Mevzuat Bilgi Sistemi. https://www.mevzuat.gov.tr/mevzuat?MevzuatNo=6698&MevzuatTur=1&MevzuatTertip=5

- draw.io Diagrams Online. https://app.diagrams.net/