Project Analysis Report

Project: Face Guard

Web Page: https://faceguardtedu.wixsite.com/faceguard

Team Members: Umut Yıldırım, İrem Tamay, Ceren Büyükgüllü, Begüm Akdeniz

Supervisor: Fırat Akba

Jury Members: Venera Adanova and Emin Kuğu

1. Introduction

Modern technology like Face Guard contributes to raising security standards. It makes use of machine learning and artificial intelligence. Its primary duty is to identify faces and confirm someone's identity using their facial characteristics. It takes an image or a video of the person's face, evaluates characteristics such as the nose's shape and the space between the eyes, and as a result, it makes entering the university simpler, faster, and safer. Reliability will rise while human power is reduced. Foreigners won't be allowed to enter the university without authorization since Face Guard will recognize the faces of students and staff instead of requiring a card to access.

2. Current System

Our current door security system is provided by reading the cards to the door. In the condition of switching to this system created by FaceGuard,

1. Situations such as forgetting/lost/stolen the card will be eliminated.

2. The transfer of the card to another person and foreigners will be prevented from entering the school.

3. The number of guards waiting at the door and working hours can be reduced.

3. Proposed System

1) Camera: The system will be equipped with a camera that captures images of individuals' faces who wish to access the campus. The camera will be positioned at the entrance of the campus.

2) Face Recognition Software: The face recognition software will analyze the captured images and match them with stored user profiles in the system. It will be designed to handle a high volume of traffic and provide fast and accurate identification of individuals.

3) Database: The system will store user profiles in a database that will be accessed by the face recognition software. The database will store user information such as name, photograph, and access privileges.

4) User Interface: The system will have a user-friendly interface that allows administrators to add, delete, and update user profiles in the database.

5) Integration with Security Systems: The face recognition software will be integrated with existing security systems on the campus to ensure a seamless experience. This includes integration with door access control systems, security alarms, and security cameras.

6) Scalability: The system will be designed to accommodate future expansion, with the ability to add new users and update profiles as necessary.

By implementing this proposed system architecture, our face recognition software project will provide a reliable and efficient system for campus security and access control.

3.1 Overview

At the beginning, FaceGuard checks if it can distinguish the face features, if it can, then it captures an image in the video stream. Secondly, FaceGuard extracts features of the face, such as the distance between the eyes. The next step is, FaceGuard starts to analyze the face models in the database. If it finds a face in the database, it starts to compare the face features. Finally, if these faces match, the gate will open, otherwise, the entrance will not be allowed.

3.2 Functional Requirements:

- The software must be able to capture and process facial images from the camera on the school campus.
- The software must be able to match the captured facial images with stored user profiles in the system.
- The system must be able to accurately identify individuals, with minimal false positives or false negatives.
- The system must provide access control by allowing authorized users to access specific areas of the campus.
- The software must be able to update user profiles as necessary.

3.3 Non-Functional Requirements:

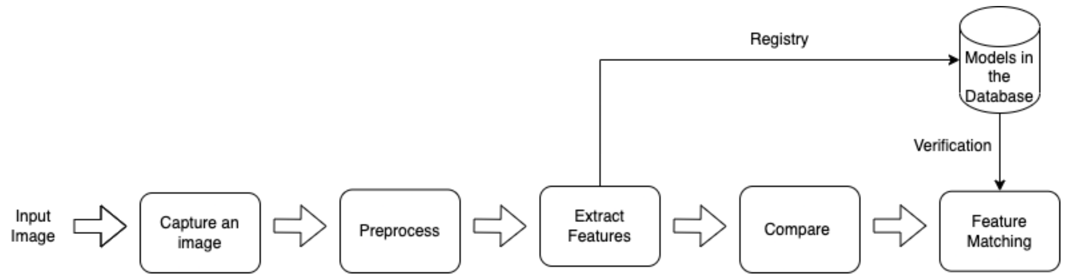
- Accuracy: The system must have a high level of accuracy in identifying individuals, with an accuracy rate of at least 95%.
- Speed: The software must be able to process facial images and match them with stored profiles in under 2 seconds.
- Scalability: The system must be able to handle a high volume of traffic and be scalable to accommodate future expansion.
- Security: The system must be designed with security in mind, with data encryption and protection against unauthorized access.
- Usability: The system must have a user-friendly interface that is easy to use and navigate for both administrators and users.
- Reliability: The system must be highly reliable, with a system uptime of at least 99.9%.

By ensuring that the functional and non-functional requirements are met, the faceguard will provide a reliable, accurate, and efficient system for campus security and access control.

3.4 Pseduo Requirements

1. The system must be able to distinguish facial features accurately.
2. The system must be able to capture and process images or videos of faces.
3. The system must be able to capable of extracting facial characteristics like the size of the eyes and nose's shape.
4. A database of face models must be available for the system to compare against.
5. To identify people, the system must be able to use facial recognition.
6. The system needs to have a gate or access point that can be managed according to the outcomes of the facial recognition.
7. The system must perform consistently and reliably.
8. Less human involvement in the entry process is required by the system.
9. The system needs to be secure and prevent unauthorized access.
10. The system must be able to distinguish between authorized users and students or staff.

3.5 System Models



3.5.1 Scenarios

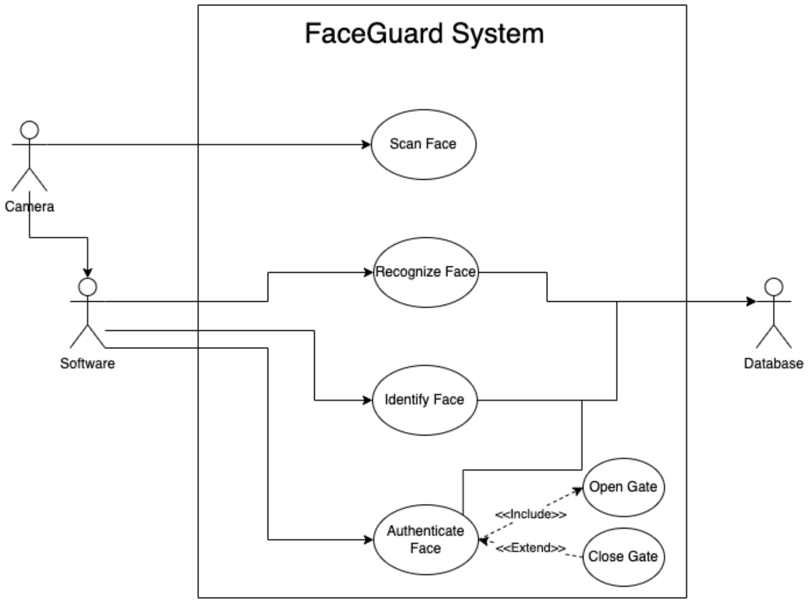
Actors: Student, security guard, school staff.

Equipment: Turnstiles, camera, works on any computer.

- A student has just graduated from high school and won the engineering department of TED University.
- He wanted to come to the school immediately and register.

- When he came to school, he encountered turnstiles at the door. He excitedly turned to security to enter and passed through the turnstiles with the help of security.
- The student who came with a passport photo and information has registered. The student was given a school card.
- This Denizbank card was a card that he would use both as a school card and a bank card. (It will replace the student ID card outside.)
- The student who leaves school with excitement is counting the days to start school.
- The student on the first day of school, saw the text “Please look at the camera” while thinking that he would have a card read at the turnstiles.
- The student was very surprised when he saw the door open automatically when he looked at the camera. He enjoyed this experience, which he had not encountered before.
- The student had an unfortunate accident 2 months later and his nose was broken. When he came to school, he thought that the camera would not detect his face, but he was very happy when the turnstiles opened automatically.
- The student excitedly asked his teacher about this system. His teacher said, “Our students developed this system for us as a graduation project. Our security burden is reduced, and even if the student forgets his card, it is easy to pass. They left us a gift from themselves. You can also do such innovative projects while you graduate.”
- After this speech, the student thought of this project while choosing a department and became ambitious.
- Now he is also a computer engineering student.

3.5.2 Use-Case Model



Use Case	Scan Face
Actor	Camera
Description	The person looks at the camera and the camera detect a face in an image (the camera can take a snap shot) or video stream.

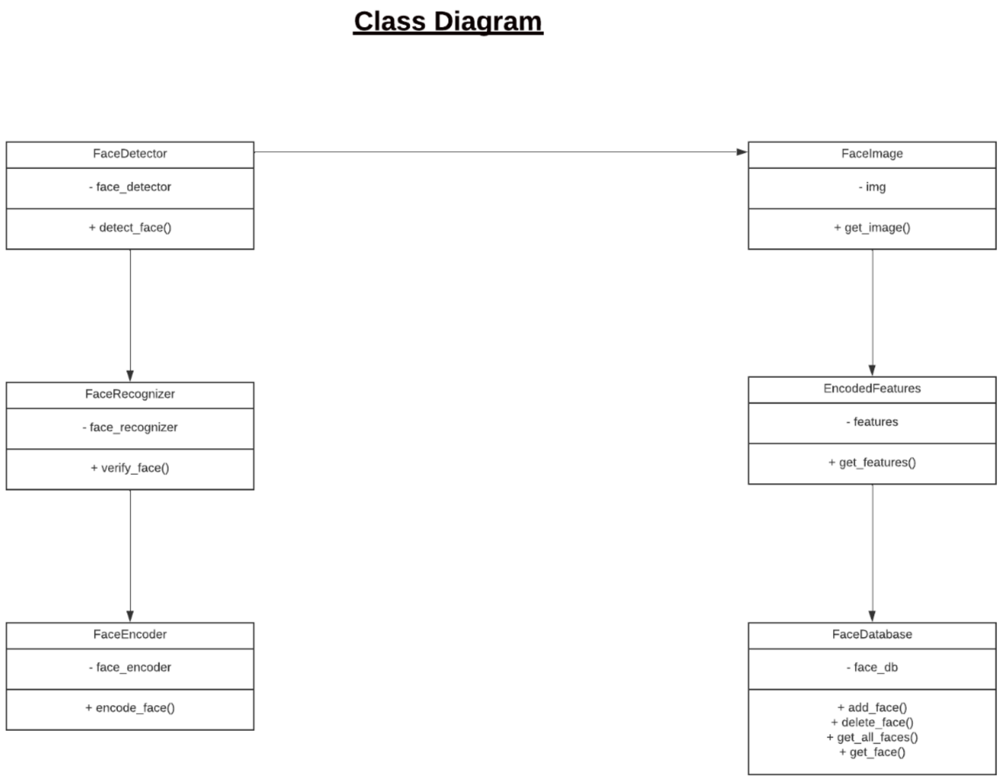
Use Case	Recognize Face
Actor	Software
Description	The software searches the detected face from the database.

Use Case	Identify Face
Actor	Software
Description	The software identifies the person whose face

	has been recognized.

Use Case	Authenticate Face
Actor	Software
Description	The software compares the detected face and the face in the database; if it matches then the gate will open, if the software cannot match faces, then the gate won’t open.

3.5.3 Object and Class Model



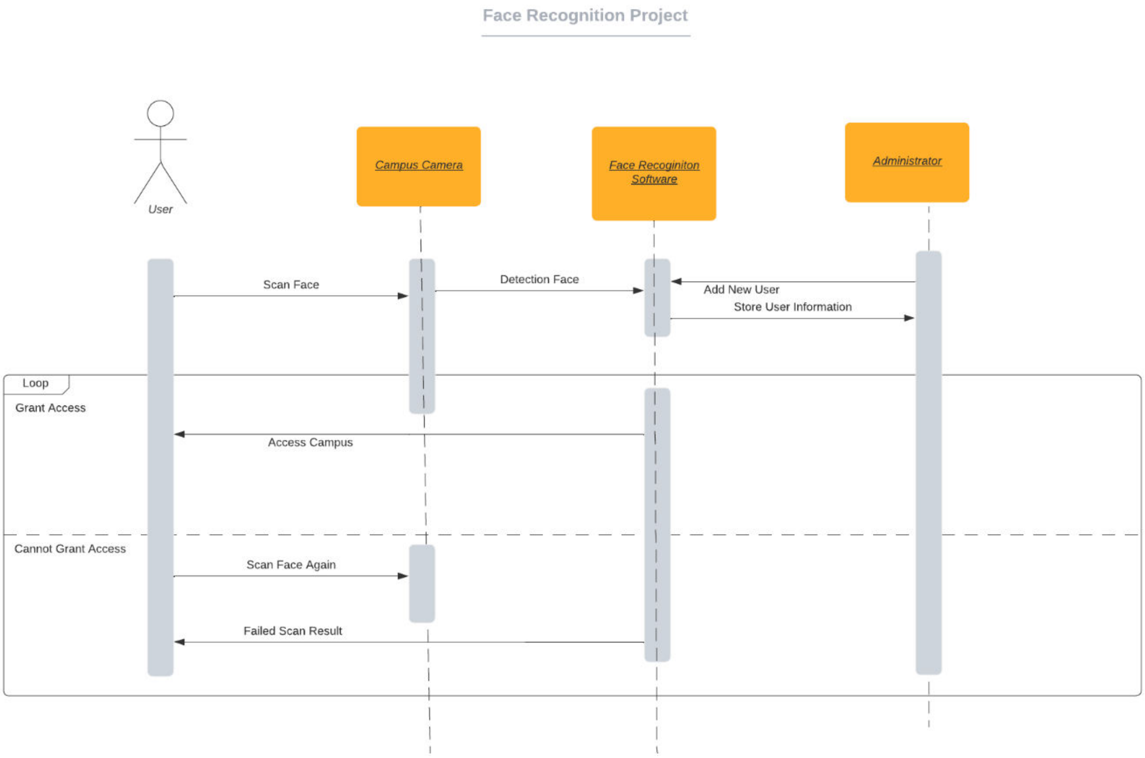
The FaceDetector and FaceEncoder classes use the FaceImage class to represent the face images that they process. The FaceImage class has a private attribute img that stores the image data and a method get_image() that returns the image data.

The FaceEncoder class returns an object of the EncodedFeatures class, which represents the encoded features of a face. The EncodedFeatures class has a private attribute feature that stores the feature vector and a method get_features() that returns the feature vector.

The FaceDatabase and AccessControl classes use the EncodedFeatures class to store and compare the face embeddings. The FaceDatabase class has methods for adding, deleting, and retrieving face embeddings from the database. The AccessControl class has a method verify_face() that takes a face image as input, encodes it using the FaceEncoder class, and compares it to the face embeddings stored in the FaceDatabase.

3.5.4 Dynamic Models

- A sequence diagram used to show dynamic models in the software.



3.5.5 User interface - navigational paths and screen mock-ups

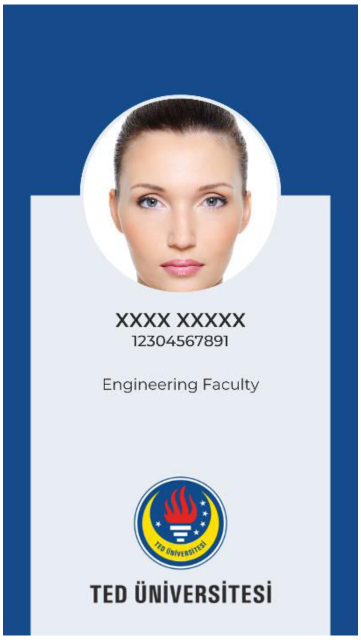
User Interface:

1. A login screen or home page where users can initiate the facial recognition process by presenting their face to the camera.
2. A live video feed that shows the user's face on the screen.
3. A feedback mechanism that updates the user on the status of the recognition process, like a progress bar or text message.
4. A confirmation screen that shows the user's identification and, if it is recognized, grants entry to the university.
5. A screen that refuses access to the institution and notifies the user of a failed recognition attempt.
6. An authorized person can add, remove, or change user face models in the database using a database management panel.
7. A panel where authorized workers can change system settings like the camera sensitivity or recognition threshold.
8. A help or support screen that explains how to use the software or get in touch with technical support.

Navigational Paths:

The specific design and functioning of the system may affect the user interface's navigational paths. For instance, a user might have to carry out the procedures in a particular order, such as putting their face in front of the camera, waiting for identification, and then getting a message of approval or rejection. As an alternative, the system might direct the user through the recognition process via voice commands or other feedback techniques. The design and functionality of the system, including the selection of colors and fonts, the positioning of buttons and icons, and the dimensions and organization of the video stream and feedback messages, may also affect the screen mock-ups.

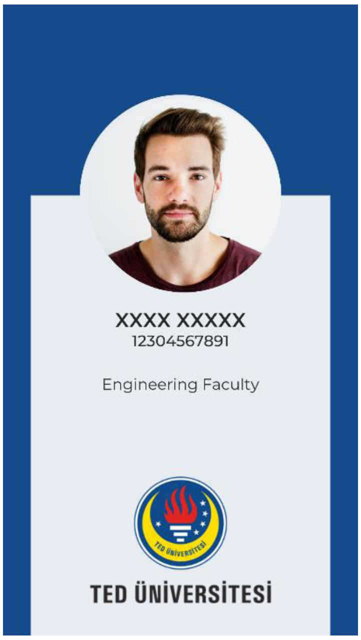
Mock-ups:



XXXX XXXXX
12304567891

Engineering Faculty

TED ÜNİVERSİTESİ



XXXX XXXXX
12304567891

Engineering Faculty

TED ÜNİVERSİTESİ

4. Glossary

- Scenarios: A set of planned actions with the goal of achieving a specific objective.
- Functional Requirements: Specific features or functions that software developers must include in a product to enable users to complete tasks.
- Non-Functional Requirements: Requirements that are not directly related to the software's functionality, but rather to its performance attributes such as reliability, efficiency, usability, maintainability, and portability.
- Use Case Model: A description of how a proposed system will function and what it will be capable of doing.
- Class Model: A visual representation of a system's structure, including the relationships between objects and classes within the system.
- Dynamic Model: A model used to illustrate and analyze a system's behavior over time.
- The user interface (UI) is the component of a device or software that enables interaction and communication between humans and computers. It serves as the point of contact where users can input data, receive feedback, and perform tasks using the device or software.

5. References

1) https://www.lucidchart.com

2) https://www.github.com

3) https://homes.cs.washington.edu

4) https://www.wikipedia.org

5) https://visuresolutions.com

6) https://app.diagrams.net/

7) https://hudsonsteelworks.com/pseudo-requirements/

8) https://uim.fei.stuba.sk/wp-content/uploads/2018/02/Object-oriented-Software-Engineering-3rd-Edition.pdf