Bilkent University

Department of Computer Engineering

CS 491- Senior Design Project

Project Name: PANDETECT

High Level Design Report

Group Members: Selen Uysal, Berk Güler, İrem Seven, Ufuk Bombar, Batuhan Tosyalı

Supervisor: Selim Aksoy

Innovation Expert: Ahmet Eren Başak

Jury Members: Çiğdem Gündüz Demir, Can Alkan

1. Introduction

Due to the COVID-19 pandemic which we have faced in 2020, there have been major changes in our lifestyle. Since COVID-19 is a virus that can spread very easily, it affected many people all around the world. In addition to the easy spread of the virus, it poses a big problem due to its high lethal effect. There are some rules that people must follow from the beginning of the pandemic in order to protect themselves and society from the virus. Two of the most important of these are wearing masks and keeping social distance. For this reason, states require wearing masks and impose curfews at regular intervals. This means that although the rules are tried to be applied, there are cases where the mask and social distance control cannot be fully achieved. Therefore, we aimed to make a system to prevent these situations. It is important to note that although PANDETECT will be developed with COVID-19 in mind, it will be a system that may be used for other pandemics in the coming years.

PANDETECT is an integrated system that will detect whether people in a particular area obey social distance rules and whether they wear masks or not. Being an integrated system, PANDETECT consists of two separate applications one in the form of a desktop and one in the form of a mobile application. Our system aims to extract data about whether the rules are followed or not in specific locations in order to present it to users in the mobile application's map view. All users who want to take advantage of our system can choose to stay away from areas that could put their health at risk. Business owners, on the other hand, can use our system if they want to have more effective control in their spaces, or if they want to provide data to their customers as an indication of the safety levels of their places. With the help of the desktop application, they can watch the stream of cameras, while getting instant data, which will enable them easy and effective control. Likewise, individuals or institutions responsible for regulations can use our system to audit the people and places more effectively.

1.1 Purpose of the System

To keep a safe environment during the pandemic, governments have the duty of inspecting the public places and closed areas such as restaurants, to see whether they comply with the COVID-19 regulations to the law enforcement officers but it requires a lot of ground-work and it is known that humans are more error-prone compared to a machine. In order to maximize the efficiency of the regulations, the PANDETECT system will continuously check whether the regulations are being complied with, and statistics will be generated according to the camera device's field of view.

The main goal of the mobile application is to identify places, by extracting data, that do not comply with the pandemic rules such as restaurants, workplaces, and public areas. In this way, it is expected to increase the control in the places where do not comply with the rules. This information will be shared on a map in the PANDETECT mobile application using the data of the places where rules are followed and not followed. That way people can choose not to go to areas where the rules are not followed according to the map provided. Thus, our system aims to inform users about compliance with the rules in some specific places. Also, the provided data can be used to increase regulations by the government. Similarly, business owners can use the system to make sure they have control over their places. With the help of the desktop application, it will be easier for them to control the violation situations. That way it will be easier for them to keep a healthy environment.

The spread of viruses is very similar regardless of their types. Thus, we aimed to use our system for not only the COVID-19 pandemic but also for future pandemic conditions. Also, it is important for us to implement real-life solutions with computer vision concepts. We believe that implementing such real-life solutions will contribute to intellectual and scientific knowledge.

1.2 Design Goals

1.2.1 Cost

The application will be free for the users, but they can only access the statistical information, not the camera feed. The application will be charged for business owners who use our system in their places, which will enable them to use the desktop application, and to access streaming videos from the mobile application.

1.2.2 Speed

In terms of speed, considering we use the time for face and mask recognition is fairly good and satisfying for now, in the deployment of the product we may want to write a faster algorithm for visualization.

1.2.3 Security

Since we are abiding by the rules of KVKK (Kişisel Verilerin Korunması Kanunu) in our system, we only use people's images and places' video footages to generate mask recognition and social distancing statistics. User credentials will be stored encrypted in a secure database. Streaming service for business owners will not be available to the public. Also, our mobile application will not ask for any payment information since it is free for public users. The business owners will make the payment during the set up of the camera and software, therefore, there will not be any security problem in terms of payment.

1.2.4 Logistics

The money required for our system which is currently only the camera devices are provided via the group members. For our implementation, we may need to buy IP cameras which are around 500 - 2000 TL each depending on the quality.

1.2.5 Reliability

The application should be stable. The application should produce results with probabilistic certainty. The database should reliably store information without data loss. The database should reliably transfer data without data loss. Also, we will constantly analyze and revise our choices in terms of algorithms, detection models, and camera devices.

1.2.6 Usability

The system should be available to all of the users who use smartphones, and the operating system used by the device will not affect the application's availability. The application can be used on both iOS and Android platforms. The application should provide an easy to use interface. Desktop part of the application should be easy to use as well.

1.2.7 Performance

Camera devices will analyze the video stream to reduce performance loss and prevent overload on the server. This way the processing load will be distributed. The servers should be optimized for live stream features for business owners.

1.2.8 Extendibility

In future, the application may be extended to involve thermal detection with the help of thermal cameras. That way it can be detected if the people have fever in the corresponding place by the application.

1.2.9 Marketability

Business owners may use this product to attract customers when they show they comply with rules. Government may use this product to automate the inspections of the places.

1.3 Definitions, Acronyms, and Abbreviations

Table 1: Definitions, Acronyms and Abbreviations

Term	**Definitions, Acronyms, and Abbreviations**
OpenCV	OpenCV (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library” [4]. It provides many useful functions for implementing computer vision related programs.
Computer Vision	Computer vision is a field that trains computers to interpret and understand the visual world by using videos and digital images. It can be considered as a subfield of artificial intelligence, so with the help of deep learning models, machines can identify and classify objects from the digital or real world [5].
Tensorflow	Tensorflow is an open-source machine learning platform [6].
Keras	Keras is a high-level deep learning API, running on top of Tensorflow that is a machine learning platform [6].
Python	Python is an object-oriented, high-level programming language. It is an interpreted language and has dynamic semantics. Python is mostly used for programming in fields such as artificial intelligence, deep learning, and computer vision [7].

Javascript	JavaScript is a scripting or programming language that is mostly used in web programming. It can be also used to program multi-platform mobile applications [8].
Redis	Redis is an open-source in-memory data structure store, which can be used as a database, cache, and message broker [9].
PostgreSQL	PostgreSQL is an open-source object-relational database system. It uses SQL language to safely store complicated data [10].
ESP32-Cam	The ESP32 is a low-cost system-on-chip (SoC) series that can be used in the development of IoT projects and embedded systems. It has Wi-Fi and Bluetooth capabilities that can provide many functionalities [11].
KVKK	KVKK (Kişisel Verileri Koruma Kanunu) is the law in Turkey that regulates the protection of personal data and outlines the obligations. It deals with personal data privacy and protection [12].

1.4 Overview

PANDETECT will detect if there are people who are not wearing masks and determine their number. With certain algorithms, a ratio or percentage will be given whether the mask rule is followed in a place. At the same time, using the distance values on the image obtained, it will be determined whether the social distance is complied with or not. Tools such as OpenCV, Keras, and Tensorflow will be used for the mask and social detection implementation purposes.

PANDETECT will provide the information both in the mobile application form for all users, and the desktop form for business owners. Public users can only view map information consisting of compliance rates such as the ratio of people who wear masks and who did not, and the ratio of social distance rule compliance for that time zone and month. Also, since users may not prefer to go to crowded areas, the density of the people in the environment will be given as numbers or percentages. For the mobile application, we will get data from the database system. We will use Javascript language to implement the PANDETECT mobile application.

PANDETECT desktop application will allow us to collect data that will be displayed on the mobile application, that is in the map. At the same time, it allows instant camera views to be seen on the computer in the relevant place. Thus, business owners can watch the stream of cameras which are in their place from their local desktop. The live stream of cameras will inform the business owners by indicating the people who comply with rules or who do not by putting marks on them such as showing rectangles on the people's faces. Business owners or managers can also have information about instant data using the desktop application. For the desktop system, Python language will be used for detection methods and providing data to the database system. The obtained data will be analyzed and corresponding information will be provided in the related map location, which is in the mobile application.

In the mobile application, there will also be a feature for the business owners. After the related system is set up for the related place, an account will be given the authority to become a business owner account. Business owner users will enter the mobile application by providing their username and passwords. That way business owners or managers will have the opportunity to see camera streaming of their places in the PANDETECT mobile application. To be able to provide live streaming through mobile application, Redis will be used. The data flow to the mobile application will be provided by the main computer placed in each contracted place. The PANDETECT system that we will install on this computer will provide up-to-date data to the mobile application every 15 minutes. Our goal in doing this is to declare instant changes while providing as much up-to-date data as possible.

2. Current Software Architecture

There exist some mask detection and social distance algorithms that have been developed, and applications have been made since the beginning of the pandemic situation. Publications about mask detection and social distance algorithms are still being provided and researches are still being conducted. Since the necessity to wear a mask has just entered our lives, there are not many products in the market for this purpose. Most of the products are generally small-scale and are not at a sufficient level to provide large-scale supervision. Also, these applications generally only provide either mask detection or social distance control. Our system will provide both mask detection and social distance control at the same time, so it will be separated from other applications in this respect. The fact that we will not only provide mask and social distance control but also will show the collected data of the places on a map with various places, distinguishes our application from other applications. Below, it is given some detector applications that are related to our project scope. These applications contain only one of the features that our system will offer as we mentioned before.

2.1 XOVIS Face Mask Detector

XONIS sells the product as an add-on to artificial intelligence-enabled sensors. It provides mask detection through sensors. The product does not provide any compliance information and it is not attached to a map. Therefore, it is only similar to our product with its mask detection concept [1].

2.2 MAGGY Social Distancing Safeguard

The Maggy social distance detector is a small wearable device. The device is attached to the neck and warns the person with sound and vibration when the social distance is not respected. It uses distance sensor technology. It only warns the people who wear it [2].

2.3 Aerialtronics Face Mask Detection System

Aerialtronics provides a software system for mask detection that runs on computers. The system is similar to our system in a way that it also uses IP cameras to analyze the face mask detection rather than sensors. This system only provides mask control and does not provide social distance control. In addition, our system differs from this product in terms of functionality since it will present the data obtained to users via the mobile application [3].

3. Proposed Software Architecture

3.1 Overview

Overall, our main system is divided into three main sections; mobile application, backend system, and the camera device. The mobile application tier consists of accessing the database from the smartphone and providing different user interfaces based on different user types that are logged in. Backend system consists of Redis database, SQL service and a controller to pass data for the other devices. The camera system consists of the logic between IP cameras and ESP32 microcontrollers.

3.2 Subsystem Decomposition

Subsystem decomposition of the device system, backend system and mobile application are given in Figure 1, 2 and 3, respectively. These figures depict the layers and subsystems. Detailed information about these subsystems will be given in Section 4.

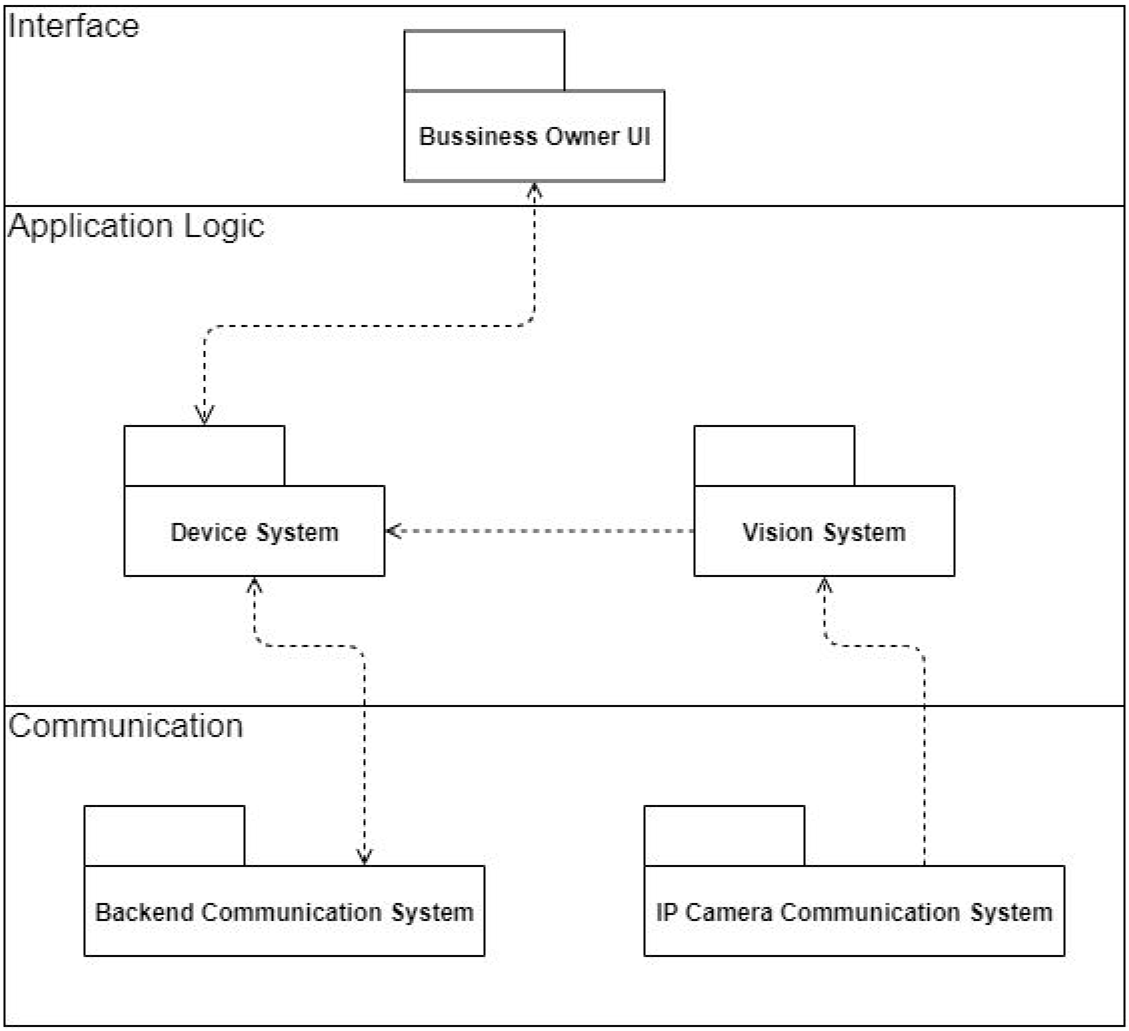

Figure 1: Subsystem Decomposition of the Device System

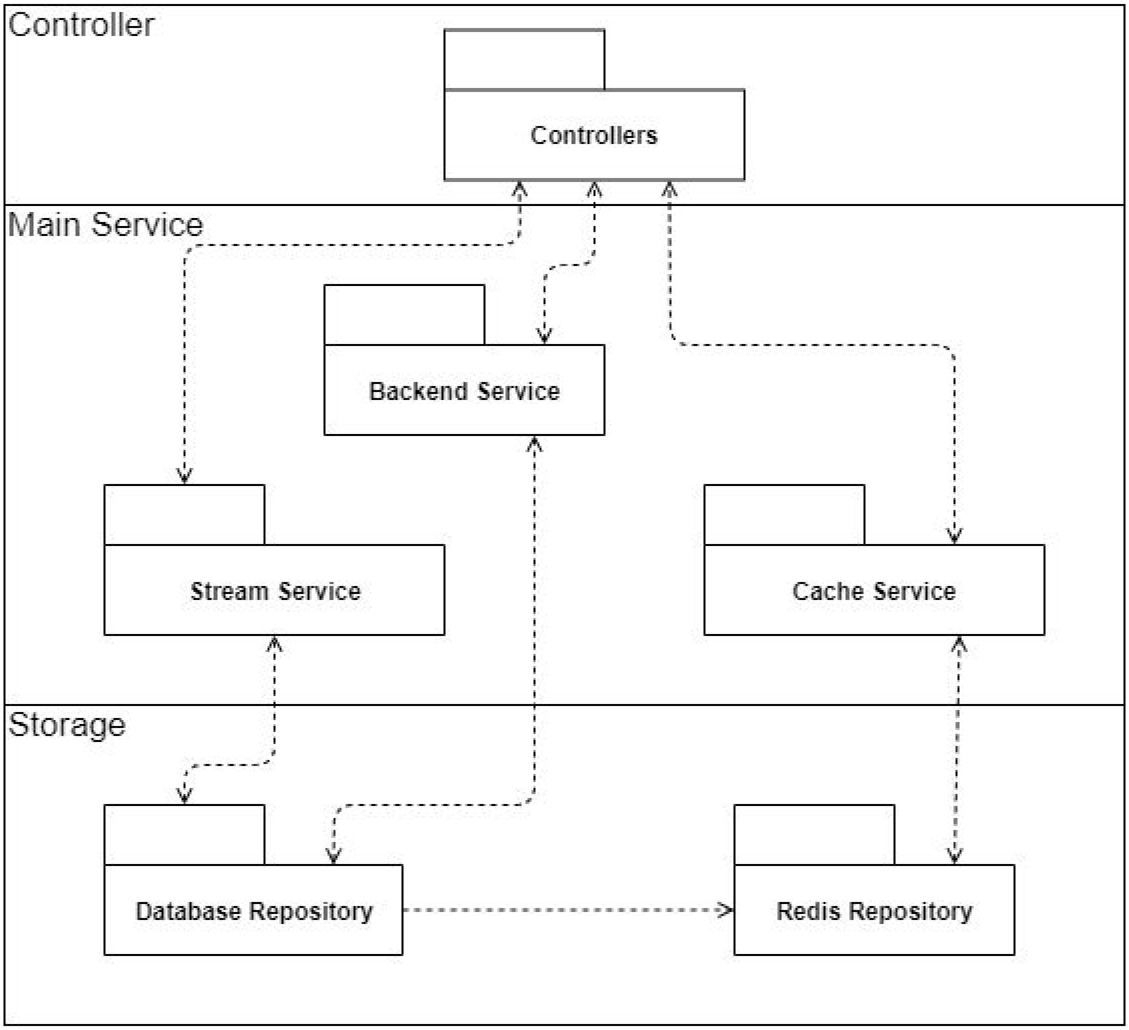

Figure 2: Subsystem Decomposition of the Backend System

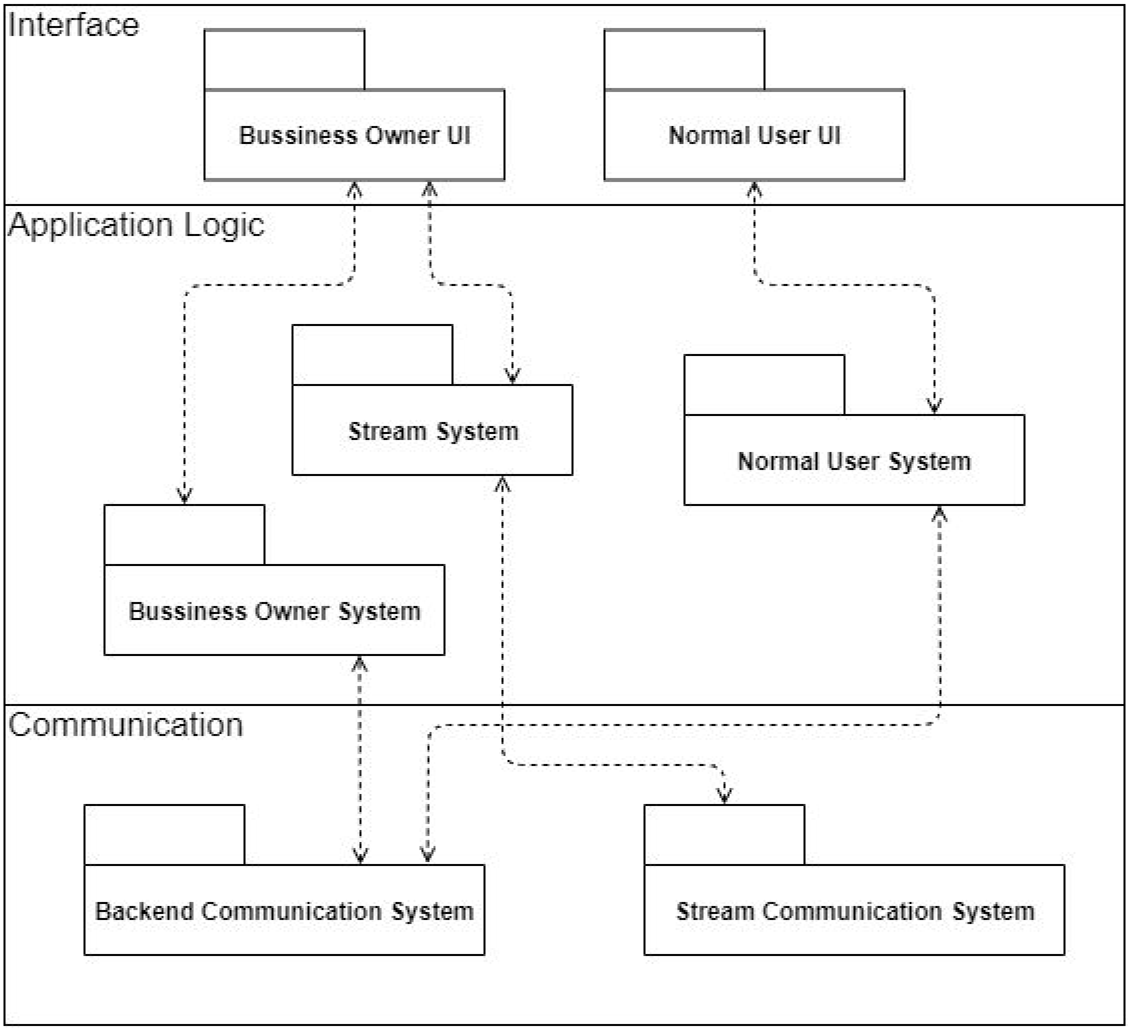

Figure 3: Subsystem Decomposition of the Mobile Application

3.3 Hardware/Software Mapping

The mapping diagram can be seen in Figure 4. The backend will live in the backend server with the redis and postgresql database. The mobile application will be located in the phones of the public and business owners' phones. As the camera controller device, we will utilize an on board computer with a well enough processor for image processing. This controller will be in communication with the IP cameras via the local wifi. ESP32-CAM module will be used as the ip cameras.

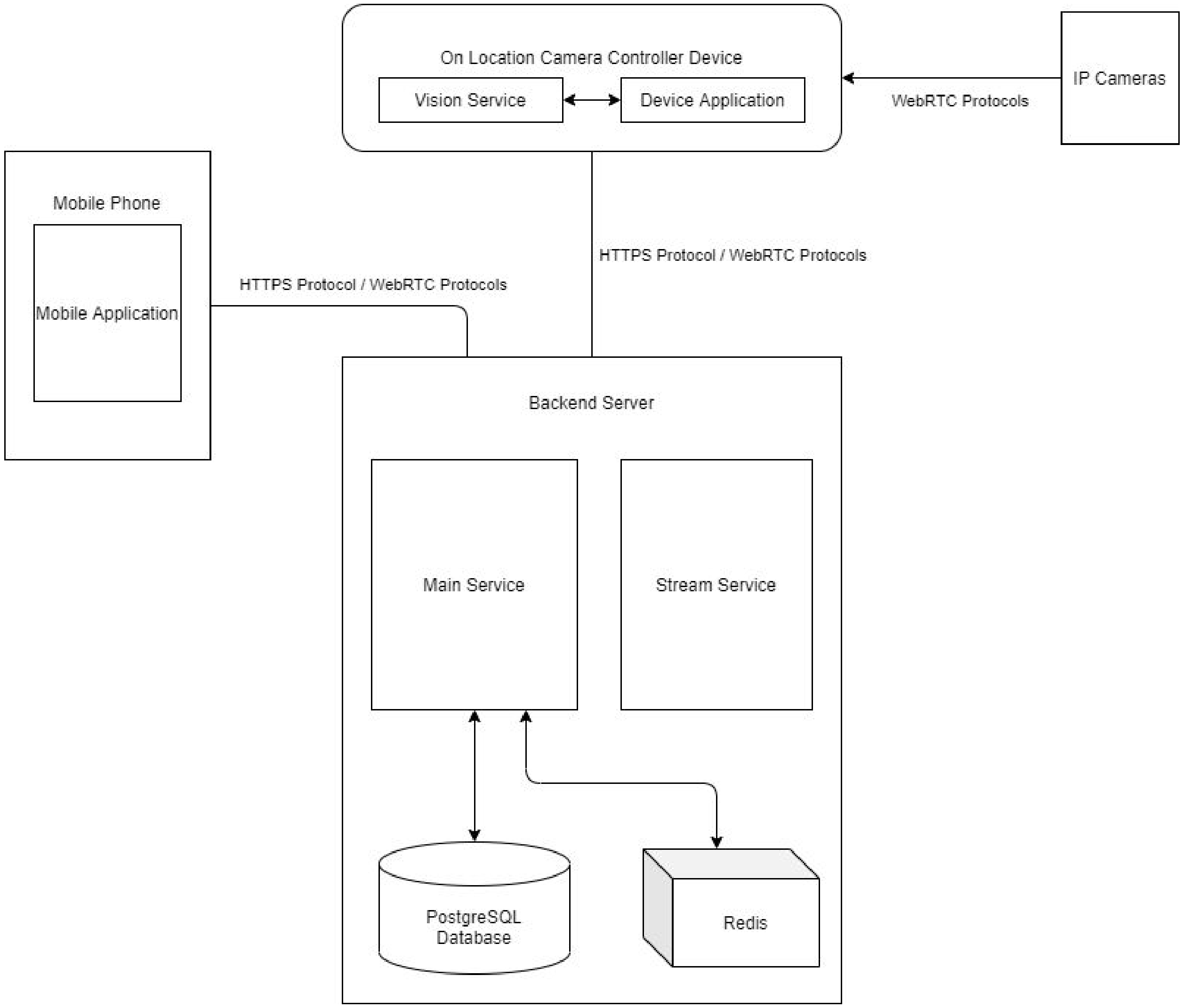

Figure 4: Hardware Software Mapping Diagram

3.4 Persistent Data Management

We will be using Redis[x] as an in-memory database, i.e. non-persistent storage for purposes such as caching. We will be utilizing the PostgreSQL as our relational database. By this way, we will ensure the integrity and the persistence of our data. The database will be lived in the Google Cloud platform servers.

In the design of the database, we will make sure to have additional tables to store non-occasionally accessed objects. We will also index our data to fasten the retrieval.

3.5 Access Control and Security

In terms of security, we will not ask public users for special permissions, since abiding by the rules of KVKK we will only use the stream as datas. In terms of other types of users such as business owners, since the streams are shown locally, the stream cannot be seen by any third party application or user.

3.6 Global Software Control

The public users as the name suggests are not required to log in to use our system. However, we need to require the synchronization between mobile application device controller device and the backend system. This is why we will use a well known protocol called HTTP. Backend will return JSON to HTTP requests coming from the mobile application. Also for streaming, the WebRTC framework will aid us to increase the performance. Another tool we will use is the JWT tokens. We will use this technology to make our communication more secure.

3.7 Boundary Conditions

3.7.1 Initialization

Both users must be using the mobile application of PANDETECT and sign in to the system. If they do not have an account, they must sign up to the system.

3.7.2 Termination

Both users can terminate the application by signing out from the application or closing it.

3.7.3 Failure

Public users may get some errors while they are opening map interface, viewing mask usage and social distance percentage in selected areas in different time intervals, reporting a bug/suggestion in the application. Additional to the signing in and signing up, business owners may get errors while they are sending a form to sign up or watching their own place's camera's live stream. These errors may occur because of an internet connection, already having a user with entered email, camera device's condition or just a system failure. For all the possible errors, detailed information about the error will be displayed to the users to avoid any confusion.

4. Subsystem Services

4.1 Device System

4.1.1 Interface (Business Owner UI)

The interface of the device system will let business owners to see all of their camera live feeds in one screen in divided squares all together or one camera in full screen. Thus, it allows business owners to manage and observe everything from a single interface.

4.1.2 Application Logic

This layer is responsible for transmitting the vision related information from Vision System to Device System.

4.1.2.1 Device System

Device system is responsible for acting as a bridge between interface layer and communication layer. For instance, this system validates inputs before sending it to the backend system. Device system also takes part in streaming.

4.1.2.2 Vision System

This is the system that draws rectangular boxes around people that are detected. It will also be responsible for checking whether detected people are not wearing their mask properly and whether they are complying with social distancing rules etc.

4.1.3 Communication

This layer will be responsible for communication with the backend server and the ESP32-Cam module.

4.1.3.1 Backend Communication System

Backend communication system is responsible for sending HTTP requests to our API. Then, it normalizes the returned response from the server and sends it to the device system.

4.1.3.2 IP Camera Communication System

This system uses WebRTC. The only duty of this system is to get the video stream from the ESP32-CAM module to transmit the information to the vision system to be fed to the neural network.

4.2 Backend System

4.2.1 Controller

The controller layer will hold all of the controllers in other words endpoints. When invoked these endpoints will trigger a function where it does a specific task.

4.2.2 Main Service

The main service layer holds all the services or functionalities of the backend system as a whole. The important difference between the controller is that the services can be shared by the controllers. One controller may invoke more than one service functionality from time to time.

4.2.2.1 Backend Service

This service is the logical layer of the backend system. It has all the functionalities such as altering the information in the relational database. In addition to the manipulation of entities in the database, this service will also register the statistical data to Redis to be fetched later. However, these low level operations will be done by the Storage Layer.

4.2.2.2 Stream Service

Stream service will register the stream into the database when a stream is opened. Then, it will act as a bridge between the mobile application which is logged in as a business owner and the camera control device. Note that the camera device will also gather the data from the ESP32-Cam module using the same streaming technology, WebRTC.

4.2.2.3 Cache Service

Cache service will be a middle layer system. On some API calls the database access will be timely inefficient, this is why this service will first check the result is available in the cache. If the result is available it will be gathered from Redis. If not, the call will continue to the services.

4.2.3 Storage

This layer will enable us to communicate with the relational database and Redis. In this repository, there will be high level functions for service layer functions to call.

4.2.3.1 Database Repository

This repository is an abstraction to encapsulate the database functionalities within a system. Some of the functions here may also invoke other repository functions. For example, deleting an entity may result in adding a log to a table. In this case, more than one method can be called. Also, changing a value in the relational database should require changing it in the Redis also.

4.2.3.2 Redis Repository

This repository is an abstraction for accessing the Redis, which is an in-memory database. The caching operations are also accessible here.

4.3 Mobile Application System

4.3.1 Interface

The interface of the mobile application will serve the following purposes: first, it will enable public users to see the mask and social distance statistics of businesses. Second, it will enable business owners to login and operate on the system. These operations include streaming video, registering new businesses and cameras.

4.3.1.1 Public User UI

The interface of the mobile application will serve to the public users without any need to login. In the first page, the UI will provide a map and all the statistics of different businesses. When a business is selected, a page with detailed information about the business will be presented to the user.

4.3.1.2 Business Owner UI

In addition to the public user UI, there will be a login as a business owner button. When clicked, an additional interface will be unlocked by the user. These pages will include the streaming page which users select one of their cameras and stream. Another page will provide Bluetooth connection options as well as the QR code reader for the identification number of the

ESP32-Cam. This page will ease the installation of ESP32-Cam module. Similarly, another page will provide the functionality to register a new business to the system.

4.3.2 Application Logic

Application logic will handle the low level business logic.

4.3.2.1 Stream System

This system will act as a middleware between the Stream Communication System and the Business Owner UI. The actual video stream will be filtered to include the calculated bounded boxes. The image then delivered the Business Owner UI.

4.3.2.2 Business Owner System

This part of the application will be responsible for accessing the Bluetooth and camera for camera registering operation. We store all the information here to minimize the communication between the application and the backend server.

4.3.2.3 Public User System

The public user system will convert the requests made by the user in the interface to REST API calls. The calls are made to the backend server. This part of the system will also be accountable for checking the validity of the inputs.

4.3.3 Communication

This layer will be responsible for authentication and encryption of the communication. The communication between the backend server and the mobile application will be secured.

4.3.3.1 Backend Communication System

This system will handle all the REST API calls with the backend server. As discussed, this system will use authentication methods to provide safe and secure communication. This is ensured by the tokens held by this system. Additionally, the HTTP protocol will be used.

4.3.3.2 Stream Communication System

This system will only be invoked if the user is logged in as a business owner. In that case, the streaming will be handled by this system. The system will also be connected securely to the backend server using WebRTC protocols. HTTP protocol will not be used by this system.

5. Consideration of Various Factors in Engineering Design

The factors which affect the end product PANDETECT will be analyzed below. In addition, Figure 5 demonstrates the factors, their effects and the level of these effects which are ranked between 1 and 10.

5.1 Public Health

Since the main aim of the application is to inform and warn users about the places where the social distance and mask rules are not complied with, the users may decide not to go to these places. This may decrease the possibility of encountering the virus and may affect the users' health in a good way. To accomplish this aim, updating the users constantly about the compliance percentages is crucial. In addition, the only possible factor that the application can be harmful to the users occurs when the users look at their mobile screen too much without breaks and exhaust their eyes.

5.2 Public Safety

To accomplish the safety of the application, the personal data of the users such as their email addresses, usernames and passwords will strictly be protected and not shared. Also, the video footage of the business owners' places will only be available to them, and will not be shared with other users or used for other purposes.

5.3 Public Welfare

Since the application shares the COVID-19 rules' compliance rates of the places, the places where the compliance rates are high may attract more customers. Thus, this may affect their welfare in a positive way. On the contrary, the places with low compliance rates may lose customers since the users may not prefer to go to these places. Other than these, PANDETECT does not affect the total welfare of the public.

5.4 Economic Factors

The application will be free for public users to provide access to everyone. However, for the business owners, the application will be charged since setting up the camera and software have to be arranged in their business places.

5.5 Social Factors

PANDETECT will possibly affect the social interactions between individuals in real life. This is because the application informs the users with the places' COVID-19 rules' compliance statistics and the distribution of the people in a place may be affected by the decisions of people of going there or not.

5.6 Environmental Factors

PANDETECT will not have a direct impact on the environment or be affected by environmental factors since it is a mobile application that uses camera devices to get data. Also, the required camera devices for places and the used software will not affect the environment in a negative way.

5.7 Cultural Factors

Since the application only demonstrates the compliance percentage and statistics of the mask usage and social distance, there are not any cultural factors that should be considered. Moreover, the COVID-19 rules do not differ between different cultures, so the application does not need to be modified for other cultures.

5.8 Global Factors

The application's language will be in English. Since it is a global language, a wide range of users around the world can use it. The application is not restricted with Turkey and can be used by any country as long as the required software and camera devices are set up for each place. Therefore, it can easily become an application which is used globally.

Table 2: Factors, Effects and Level of Effects

Factors	**Effects**	**Level of Effects**
Public Health	Tries to prevent getting COVID-19, may exhaust eyes	9
Public Safety	The personal data of users (username, password, email address, video footage) are protected	8
Public Welfare	Businesses lose or attract customers	5
Economic Factors	Free for public users, charged for business owners	4
Social Factors	Distribution of people in places are affected	8
Environmental Factors	Software and camera devices are not harmful to the environment	2
Cultural Factors	COVID-19 regulations do not differ between cultures (No cultural factor can affect the application)	1
Global Factors	Adaptiveness of the application and language preference makes it globally used	5

6. Teamwork Details

6.1 Contributing and Functioning Effectively on the Team

We do believe that a healthy and functioning team consists of devoted members who are specialised in areas which are distinct but not mutually exclusive from each other. With this approach, all of the project members focus on one distinct job area. However, the team members can also get help from other team members who are working on a closely related task. The work distribution of the project team members are given in Figure 5. İrem Seven and Selen Uysal will work on the frontend and backend systems, Berk Güler and Ufuk Bombar will work on the backend and computer vision parts and Batuhan Tosyalı will work on frontend and computer vision parts of the project. For the work done so far, each team member contributed equally, and did their parts. For the completed reports so far, a detailed information about work distribution for each section were in the following:

<u>Project Specification Report:</u> For each section, we discussed and brainstormed together. Since it is the first report, we had to clarify the project scope, requirements and write the sections according to them.

<u>Project Analysis Report:</u>

- Introduction, Current Systems: İrem Seven
- Proposed Systems
    - Overview: İrem Seven
    - Functional, Non-functional, Pseudo Requirements: Batuhan Tosyalı, Ufuk Bombar, Selen Uysal, Berk Güler
    - System Models: Everyone
- Other Analysis Elements: Batuhan Tosyalı, Selen Uysal

<u>High Level Design Report:</u>

- Introduction: İrem Seven, Batuhan Tosyalı
- Current Software Architecture: İrem Seven
- Proposed Software Architecture: Ufuk Bombar, Berk Güler, İrem Seven, Selen Uysal
- Subsystem Services: Ufuk Bombar, Berk Güler
- Consideration of Various Factors in Engineering Design: Selen Uysal

- Teamwork Details: Selen Uysal, Batuhan Tosyalı

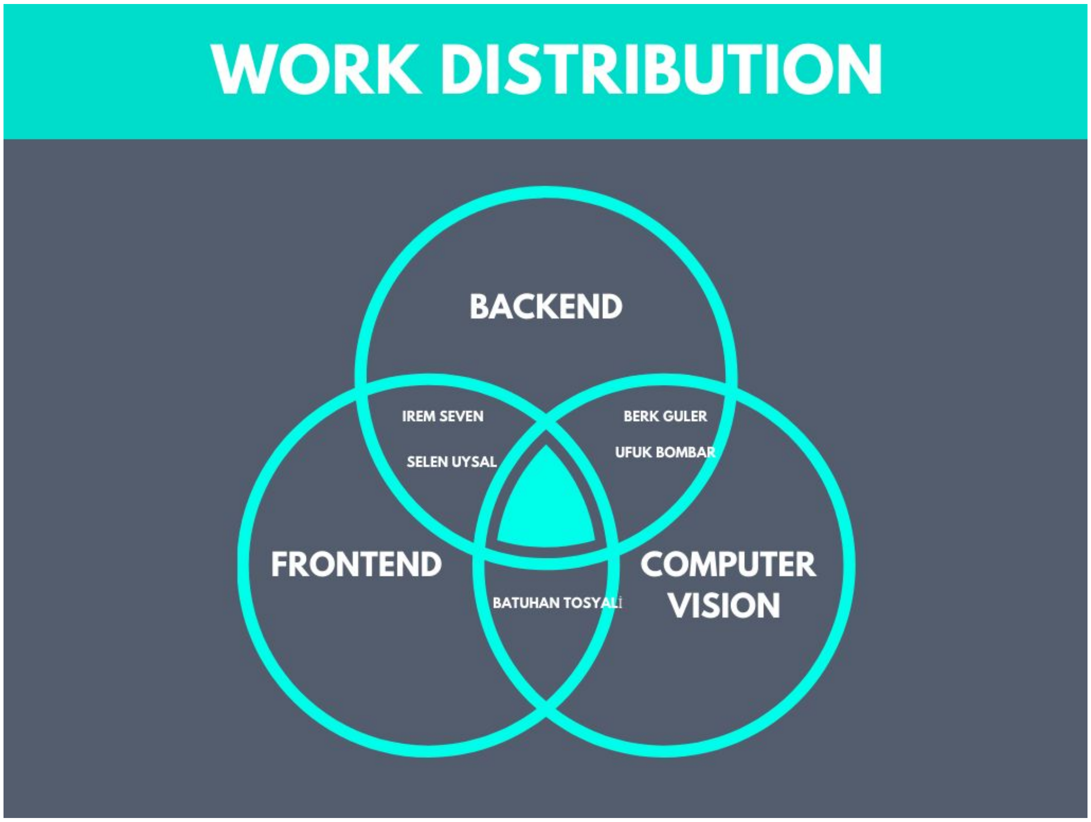

Figure 5: Work Distribution of Team Members

6.2 Helping Creating a Collaborative and Inclusive Environment

In order to achieve a collaborative and inclusive environment, we firstly give importance that every team member understands their responsibilities and tasks to prevent working on some other member’s task which would result in conflicts. Secondly, all team members meet quite frequently to make sure everyone understands the software requirements of their parts and start developing with a clear mind. These meetings also help to find creative ideas and possible ways to accomplish tasks through brainstorming and discussions. In addition, weekly meetings are done to specify the weekly goals and objectives which keeps everyone at the same page. The meetings are done on Zoom or Discord, and WhatsApp is used for instant messaging. In addition, with the help of the Git technology, we have multiple branches on our GitHub page which enables all team members to work on their part of the project without interrupting others. This also helps to monitor each team member's progress and give feedback personally. Moreover, we also maintain some collaboration with pair programming and peer review when needed. Pair programming is done by writing code together which helps finding the bugs and increases code quality. Peer reviewing also increases the cohesion among the group and helps team members to learn from one another.

6.3 Taking Lead Role and Sharing Leadership on the Team

In our team, each team members acknowledge the project leader in that project duration and follow the tasks that the leader gives. This is especially important since without a leader, the decisions cannot be made efficiently, or the tasks that should be completed within a time frame cannot be monitored clearly. Every 3 weeks through 1- 14, the leadership of the team changes periodically to ensure that we operate in a democratic and iterative system. Therefore, Table 3 shows the following leadership table due to weeks.

Table 3: The Distribution of Leadership

Weeks	**Team Members**
Week 1, 2, 3	Batuhan Tosyalı
Week 4, 5, 6	Ufuk Bombar
Week 7, 8, 9	Berk Güler
Week 10, 11, 12	Selen Uysal
Week 13, 14	İrem Seven

As it can be seen, we tried to distribute even responsibilities and give leadership to every team member periodically. The responsibilities of the team leaders for the relevant weeks are the following:

- Identifying requirements
- Establishing objectives
- Balancing competing demands for quality, scope, time and cost considering the project triangle
- Following the deadlines
- Redistributing the tasks among the team members
- Logging the updates coming from the team members
- Organizing the GitHub repository

- Distribute the tasks of the report
- Monitoring the team members’ works

7. References

[1] “Face Mask Detection,” *Xovis*. [Online]. Available: https://www.xovis.com/en/products/detail/face-mask-detection/. [Accessed: 27-Dec-2020].

[2] “MAGGY social distancing safeguard device,” *Maggy*, 07-Dec-2020. [Online]. Available: https://www.maggylife.eu/product/maggy-social-distancing-safeguard-device-free-lanyard/?gclid+=CjwKCAiA7939BRBMEiwA-hX5J4B3EADE_zAaD98mk5gBMq6J0rwibDVpLoI7hrfZBwp-Ohm6+zOa3xRoCeJcQAvD_BwE. [Accessed: 27-Dec-2020].

[3] “Face Mask Detection Software,” *Aerialtronics*. [Online]. Available: https://www.aerialtronics.com/en/products/face-mask-detection-software. [Accessed: 27-Dec-2020].

[4] “About,” *OpenCV*, 04-Nov-2020. [Online]. Available: https://opencv.org/about/. [Accessed: 27-Dec-2020].

[5] “Computer Vision: What it is and why it matters,” *SAS*. [Online]. Available: https://www.sas.com/en_us/insights/analytics/computer-vision.html. [Accessed: 27-Dec-2020].

[6] K. Team, “Keras documentation: About Keras,” *Keras*. [Online]. Available: https://keras.io/about/. [Accessed: 27-Dec-2020].

[7] “What is Python? Executive Summary,” *Python.org*. [Online]. Available: https://www.python.org/doc/essays/blurb/. [Accessed: 27-Dec-2020].

[8] “Learn web development,” *Learn web development* | *MDN*. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. [Accessed: 27-Dec-2020].

[9] *Redis*. [Online]. Available: https://redis.io/. [Accessed: 27-Dec-2020].

[10] “About,” *PostgreSQL*. [Online]. Available: https://www.postgresql.org/about/. [Accessed: 27-Dec-2020].

[11] M. Schwartz, “Getting Started with the ESP32,” *Home*, 17-Nov-2020. [Online]. Available: https://makecademy.com/getting-started-esp32. [Accessed: 27-Dec-2020].

[12] P. byA. Coos, “All You Need to Know About Turkey's Personal Data Protection Law (KVKK),” *Endpoint Protector Blog*, 10-Dec-2020. [Online]. Available: https://www.endpointprotector.com/blog/everything-you-need-to-know-about-turkeys-personal-data-protection-law/. [Accessed: 27-Dec-2020].