



Bilkent University

Department of Computer Engineering

CS 491- Senior Design Project

Project Name: PANDETECT

Analysis 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	3
2. Current Systems	4
2.1 XOVIS Face Mask Detector	4
2.2 MAGGY Social Distancing Safeguard	4
2.3 Aerialtronics Face Mask Detection System	5
3. Proposed System	5
3.1 Overview	5
3.2 Functional Requirements	6
3.3 Nonfunctional Requirements	6
3.3.1 Usability	6
3.3.2 Cost	6
3.3.3 Reliability	6
3.3.4 Security	7
3.3.5 Performance	7
3.3.6 Portability	7
3.3.7 Extendibility	7
3.3.8 Marketability	7
3.4 Pseudo Requirements	7
3.5 System Models	8
3.5.1 Scenarios	8
3.5.1.1 Scenario 1: Sign Up	8
3.5.1.2 Scenario 2: Send a Form to Sign Up	9
3.5.1.3 Scenario 3: Sign In	9
3.5.1.4 Scenario 4: Open Map Interface	10
3.5.1.5 Scenario 5: View Mask Usage Percentage in Selected Places	11
3.5.1.6 Scenario 6: View Social Distancing Percentage in Selected Places	11
3.5.1.7 Scenario 7: View Mask Usage and Social Distancing Statistics of Selected Places in Different Time Intervals	12
3.5.1.8 Scenario 8: Watch Own Place's Camera's Live Stream	13
3.5.1.9 Scenario 9: Report a Bug/ Suggestion	14
3.5.1.10 Scenario 10: View the Business Owners' Sign Up Forms	14
3.5.1.11 Scenario 11: Create the Account of the Business Owner	15
3.5.1.12 Scenario 12: View the Reported Bugs and Suggestions	15
3.5.2 Use Case Model	17
3.5.3 Object and Class Model	19
3.5.4 Dynamic Models	21
3.5.4.1 State Diagram for Public Users	21
3.5.4.2 State Diagram for Business Owners	23
3.5.4.3 Activity Diagram	25
3.5.4.4 Sequence Diagram of Public User Operations	26

3.5.4.5 Sequence Diagram of Business User Operations	27
3.5.5 User Interface - Navigational Paths and Screen Mock-ups	29
3.5.5.1 Sign In Screen of Public Users and Business Owners	29
3.5.5.2 Sign Up Screen of Public Users	30
3.5.5.3 Application Form Screen of Business Owners	31
3.5.5.4 Main Screen	32
3.5.5.5 Current Data Screen of Public Users and Business Owners	33
3.5.5.6 Past Data Screen of Public Users and Business Owners	34
3.5.5.7 Report Screen of Public Users and Business Owners	35
3.5.5.8 Stream Screen of Business Owners	36
3.5.5.9 Stream Screen of Business Owners in Desktop Application	37
4. Other Analysis Elements	37
4.1 Consideration of Various Factors in Engineering Design	37
4.1.1 Public Health	38
4.1.2 Public Safety	38
4.1.3 Public Welfare	38
4.1.4 Economic Factors	38
4.1.5 Social Factors	38
4.1.6 Environmental Factors	39
4.1.7 Cultural Factors	39
4.1.8 Global Factors	39
4.2 Risks and Alternatives	39
4.3 Project Plan	40
4.3.1 Work Package 1: Design Analysis and Specifications	40
4.3.2 Work Package 2: Prototype Implementation and High-Level Design	41
4.3.3 Work Package 3: Testing the Prototype	42
4.3.4 Work Package 4: Implementation	42
4.3.5 Work Package 5: Testing the System	42
4.3.6 Work Package 6: Release of the System	43
4.4 Ensuring Proper Teamwork	45
4.5 Ethics and Professional Responsibilities	45
4.6 Planning for New Knowledge and Learning Strategies	45
5. Glossary	46
6. References	46

1. Introduction

Whereas only healthcare workers used to wear the mask before, now it is made mandatory by most governments to wear a mask and comply with social distance because of the COVID-19 virus which caused a pandemic. Governments naturally 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, it can be continuously checked whether the regulations are being complied with, and statistics will be generated according to the camera device's field of view. Although PANDETECT will be developed with COVID-19 in mind, it will be a program 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. It consists of two separate applications one in the form of a desktop and one in the form of a mobile application. The main goal of the mobile application is to identify places that do not comply with 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.

Desktop application allows us to collect data that will be displayed on the mobile application. At the same time, it allows instant camera views to be seen on the computer in the relevant place. The live stream of cameras will inform the place owners by indicating the persons who comply with rules or who do not by putting marks on them such as showing rectangles on the people's faces. Place owners or managers can also have information about instant data using the desktop application.

In the mobile application, there will also be a feature for the place owners. After the related system is set up for the related place, an account will be given the authority to become a place owner account. That way place owners or managers will have the opportunity to see camera streaming of their places in the PANDETECT mobile application.

2. Current Systems

Since COVID-19 has been in our lives for only a year, technological products for automated mask and social distance control are very few and very new. There exist some mask detection and social distance algorithms that have been developed, and applications have been made since the first quarter of 2020. 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, the products in the market are not at a sufficient level. These kinds of products are generally small-scale and not sufficient to provide large-scale supervision. Also, these applications generally only provide either mask detection or social distance detection. 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 social distance is not respected. It uses the 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 System

3.1 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. 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. Tools such as OpenCV, Keras, and Tensorflow will be used for the mask and social detection implementation purposes. The mobile application will get data from the database system. For the mobile application, we will use Javascript language. 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 a 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.

PANDETECT will provide map information in the mobile application form, and desktop form for place owners. Users will enter the system by providing their username and passwords. 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 for the number of people in the environment will be given. Place owners on the other hand can watch the stream of cameras which are in his/her place from their local desktop.

3.2 Functional Requirements

- The application will require a signing up procedure for the business owners, since they will be the ones who can see the stream of their own camera devices.
- The application also needs a signing up and login procedure for public users.
- The business owners may see the stream of the camera devices that have been setup in their business place on the same screen.
- The business owners can select individual camera scenes to display on the screen of desktop computers..
- Any user will be able to see the statistical information about the places' compliance with the regulations.
- The application will produce daily reports showcasing the statistics obtained.
- Users can see previous statistics for the camera device's field of view.

3.3 Nonfunctional Requirements

3.3.1 Usability

- The application can be used on both iOS and Android platforms.
- The application should provide an easy to use interface.
- Desktop part o the application should be easy to use as well.

3.3.2 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 the desktop application.

3.3.3 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.

3.3.4 Security

- User credentials will be stored encrypted in a secure database.
- Streaming service for business owners will not be available to the public.

3.3.5 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.

3.3.6 Portability

- Both major mobile platforms will be supported as the front-end will be written in platform agnostic framework React Native, which supports all of the popular mobile operating systems (Android, iOS).

3.3.7 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.
- The application may be extended to involve the detection of coughing.

3.3.8 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.

3.4 Pseudo Requirements

- The application must operate on all kinds of mobile devices (iOS and Android).
- The application should operate with every existing mobile system written at least 5 years before.

- The application will be written in JavaScript programming language. React Native framework which is also written in Javascript will be used for the front-end which makes the application cross-platform compatible.
- Backend system will use Redis system in order to sustain real-time requirements.
- The language of the application will be English.
- Sensitive information (user picture, e-mail, address) of the customers cannot be shared with third party applications.
- The software development part of the project will be executed in Object Oriented Programming paradigm.
- The development process will be handled with Scrum.
- For version control purposes Git will be used and it will be hosted on Github.
- The system will make use of third-party libraries such as PyTorch, OpenCV and others.
- All used software tools and libraries will be licenced and necessary permission will be taken from the copyright owner if needed.

3.5 System Models

3.5.1 Scenarios

3.5.1.1 Scenario 1: Sign Up

Use Case: Sign Up

Actors: Public Users

Entry Condition(s): The public users open the application and click the “Sign Up” button.

Exit Condition(s): The public users sign up to the system successfully, and they are directed to the Main Screen.

Flow of Events:

1. The user clicks the “Sign Up” button which will appear on the Sign In Screen.
2. The user enters an email address, name and password.
3. The user clicks to the “Done” button.

Alternative Flow of Events:

A. Sign Up Fails:

- a. The user enters an invalid information to one of the required boxes to sign up to the system (invalid email address, name, business name, business location etc.)

- b. The relevant error message is displayed to the user.
 - c. The user enters a different input and clicks the “Done” button.
- B. Internet Connection is Lost:
 - a. The user loses the connection during the sign up process.
 - b. An error message, “The connection is lost!”, is shown to the user.
 - c. When the connection is stable, the user enters all required inputs again and clicks the “Done” button.

3.5.1.2 Scenario 2: Send a Form to Sign Up

Use Case: Send a Form to Sign Up

Actors: Business Owners

Entry Condition(s): The business owners open the application and click the “Apply” button.

Exit Condition(s): The business owners sign up to the system successfully, and they are directed to the Main Screen.

Flow of Events:

1. The business owner clicks the “Apply” button which will appear on the Sign In Screen.
2. The business owner enters an email address, name, password, the business information such as the name and address of the place to the application form.
3. The business owner clicks to the “Done” button.

Alternative Flow of Events:

- A. Invalid Inputs:
 - a. The business owner enters an invalid information to one of the required boxes to the application form (invalid email address, name, business name, business location etc.)
 - b. The relevant error message is displayed to the user.
 - c. The user enters a different input and clicks the “Done” button.
- B. Internet Connection is Lost:
 - a. The user loses the connection during the sign up process.
 - b. An error message, “The connection is lost!”, is shown to the user.
 - c. When the connection is stable, the user enters all required inputs again and clicks the “Done” button.

3.5.1.3 Scenario 3: Sign In

Use Case: Sign In

Actors: Public Users, Business Owners

Entry Condition(s): The users click the “Sign In” button.

Exit Condition(s): The users sign in to the system successfully, and are directed to the Main Screen of the application.

Flow of Events:

1. The user clicks the “Sign In” button at the Sign In Screen.
2. The user enters his/her email address and password correctly and clicks the “Done” button.
3. The application directs the user to the Main Screen.

Alternative Flow of Events:

- A. The User Enters Wrong Credentials:
 - a. The user enters a wrong email address or password so that the email address and password do not match.
 - b. An error message, “The email address or password is wrong!”, will be displayed to the user.
 - c. The user enters the right credentials and clicks the “Done” button.
- B. Internet Connection is Lost:
 - a. The user loses the connection during the sign in process.
 - b. An error message, “The connection is lost!”, is shown to the user.
 - c. When the connection is stable, the user enters all his/her credentials again and clicks the “Done” button.

3.5.1.4 Scenario 4: Open Map Interface

Use Case: Open Map Interface

Actors: Public Users, Business Owners

Entry Condition(s): The users sign in to the application.

Exit Condition(s):

1. The users sign out from the application.
2. The users select a place to view its details.

Flow of Events:

1. The user signs in to the application.
2. A map interface is displayed to the user, showing the places which the system has mask usage and social distancing data.

Alternative Flow of Events:

A. Internet Connection is Lost:

- a. The user loses the connection while the map is being displayed.
- b. An error message, "The connection is lost!", is shown to the user.
- c. When the connection is stable, the map is displayed to the user.

3.5.1.5 Scenario 5: View Mask Usage Percentage in Selected Places

Use Case: View Mask Usage Percentage in Selected Places

Actors: Public Users, Business Owners

Entry Condition(s): The users select a place from the map by clicking its "View Details" button.

Exit Condition(s):

1. The users go to the Main Screen.
2. The users click the "Past Data" button to view the past data of the selected place.

Flow of Events:

1. The user selects a place from the map.
2. The user clicks the "View Details" button of the selected place.
3. The selected place's mask usage percentage which is updated constantly, is displayed to the user at the Current Data Screen.

Alternative Flow of Events:

A. The Mask Usage Percentage is not Available:

- a. From the map, the user selects a place which does not have a mask usage detail.
- b. A pop up window appears that informs the user that there is no available data to be displayed.

B. Internet Connection is Lost:

- a. The user loses the connection before the mask usage percentage is displayed.
- b. An error message, "The connection is lost!", is shown to the user.
- c. When the connection is stable, the mask usage percentage of the selected place is displayed to the user.

3.5.1.6 Scenario 6: View Social Distancing Percentage in Selected Places

Use Case: View Social Distancing Percentage in Selected Places

Actors: Public Users, Business Owners

Entry Condition(s): The users select a place from the map by clicking its "View Details" button.

Exit Condition(s):

1. The users go to the Main Screen.
2. The users click the “Past Data” button to view the past data of the selected place.

Flow of Events:

1. The user selects a place from the map.
2. The user clicks the “View Details” button of the selected place.
3. The selected place’s social distancing percentage which is updated constantly, is displayed to the user at the Current Data Screen.

Alternative Flow of Events:

- A. The Social Distancing Percentage is not Available:
 - a. From the map, the user selects a place which does not have a social distancing detail.
 - b. A pop up window appears that informs the user that there is no available data to be displayed.
- B. Internet Connection is Lost:
 - a. The user loses the connection before the social distancing percentage is displayed.
 - b. An error message, “The connection is lost!”, is shown to the user.
 - c. When the connection is stable, the selected place’s compliance percentage of the social distancing rules is displayed to the user.

3.5.1.7 Scenario 7: View Mask Usage and Social Distancing Statistics of Selected Places in Different Time Intervals

Use Case: View Mask Usage and Social Distancing Statistics of Selected Places in Different Time Intervals

Actors: Public Users, Business Owners

Entry Condition(s): The users select a place from the map by clicking its “View Details” button.

Exit Condition(s):

1. The users go to the Main Screen.
2. The users click the “Past Data” button to view the past data of the selected place.

Flow of Events:

1. The user selects a place from the map.
2. The user clicks the “View Details” button of the selected place.
3. The selected place’s mask usage and social distancing percentages in the past 1 hour are displayed to the user.

4. The user selects a different time interval, and views the statistics of the past 1 day, 1 week or 1 month.

Alternative Flow of Events:

- A. The Mask Usage and Social Distancing Percentage is not Available in the Selected Time Interval:
 - a. From the map, the user selects a place which does not have monthly, weekly or hourly data of the mask usage and social distancing detail which may happen if the place is new in the system.
 - b. A pop up window appears that informs the user that there is no available data to be displayed.
- B. Internet Connection is Lost:
 - a. The user loses the connection before the monthly mask usage and social distancing percentage are displayed.
 - b. An error message, "The connection is lost!", is shown to the user.
 - c. When the connection is stable, the selected place's compliance percentage of the social distancing and mask usage rules in the selected time interval is displayed to the user.

3.5.1.8 Scenario 8: Watch Own Place's Camera's Live Stream

Use Case: Watch Own Place's Camera's Live Stream

Actors: Business Owners

Entry Condition(s): The business owners click the "Watch Stream" button in the Main Screen.

Exit Condition(s): The business owners return back to Main Screen.

Flow of Events:

1. The business owner clicks the "Watch Stream" button in the Main Screen.
2. The live stream of the business owners own place is displayed to the business owner.

Alternative Flow of Events:

- A. Internet Connection is Lost:
 - a. The user loses the connection while the live stream is being displayed to the business owner.
 - b. An error message, "The connection is lost!", is shown to the user.
 - c. When the connection is stable, the business owner's place's own live stream is displayed.

3.5.1.9 Scenario 9: Report a Bug/ Suggestion

Use Case: Report a Bug/ Suggestion

Actors: Public Users, Business Owners

Entry Condition(s): The users click the “Report” button in the Main Screen.

Exit Condition(s): The users return back to Main Screen.

Flow of Events:

1. The users click the “Report” button in the Main Screen.
2. A blank form is displayed to the user.
3. The user fills his/her suggestions or reports a bug of the application.
4. The user clicks the “Send” button.

Alternative Flow of Events:

- A. Internet Connection is Lost:
 - a. The user loses the connection before the report form is displayed.
 - b. An error message, “The connection is lost!”, is shown to the user.
 - c. When the connection is stable, the user fills the form again and clicks the “Send” button.

3.5.1.10 Scenario 10: View the Business Owners’ Sign Up Forms

Use Case: View the Business Owners’ Sign Up Forms

Actors: Admin Users

Entry Condition(s): The admin users click the “View Sign Up Forms” button in the Main Screen.

Exit Condition(s): The admin users return back to Main Screen.

Flow of Events:

1. The admin users click the “View Sign Up Forms” button in the Main Screen.
2. The sign up forms that the business owners have filled before, is displayed to the admin user.

Alternative Flow of Events:

- A. Internet Connection is Lost:
 - a. The admin user loses the connection before the business owners’ sign up forms are displayed.
 - b. An error message, “The connection is lost!”, is shown to the user.
 - c. When the connection is stable, the business owners’ sign up forms are displayed to the admin user.

3.5.1.11 Scenario 11: Create the Account of the Business Owner

Use Case: Create the Account of the Business Owner

Actors: Admin Users

Entry Condition(s): The admin users click the “Create Account” button in the Business Owner Forms Screen.

Exit Condition(s): The admin users return back to Main Screen.

Flow of Events:

1. The admin users click the “Create Account” button in the Business Owner Forms Screen.
2. The admin user creates an account for the Business Owner with the business information gathered from the form.

Alternative Flow of Events:

- A. The Business Owner’s Place is Already in the System:
 - a. The place in the form is already in the system.
 - b. The form is discarded to avoid creating multiple same entries.
- B. Internet Connection is Lost:
 - a. The admin user loses the connection while creating an account for the business owner.
 - b. An error message, “The connection is lost!”, is shown to the user.
 - c. When the connection is stable, the admin user creates an account for the business owner.

3.5.1.12 Scenario 12: View the Reported Bugs and Suggestions

Use Case: View the Reported Bugs and Suggestions

Actors: Admin Users

Entry Condition(s): The admin users click the “View Reports” button in the Main Screen.

Exit Condition(s): The admin users return back to Main Screen.

Flow of Events:

1. The admin users click the “View Reports” button in the Main Screen.
2. The reported bugs and suggestions of public users and business owners are displayed in the Reports Screen.

Alternative Flow of Events:

- A. Internet Connection is Lost:
 - a. The admin user loses the connection while viewing the reports.

- b. An error message, “The connection is lost!”, is shown to the user.
- c. When the connection is stable, the reported bugs and suggestions of public users and business owners are displayed.

3.5.2 Use Case Model

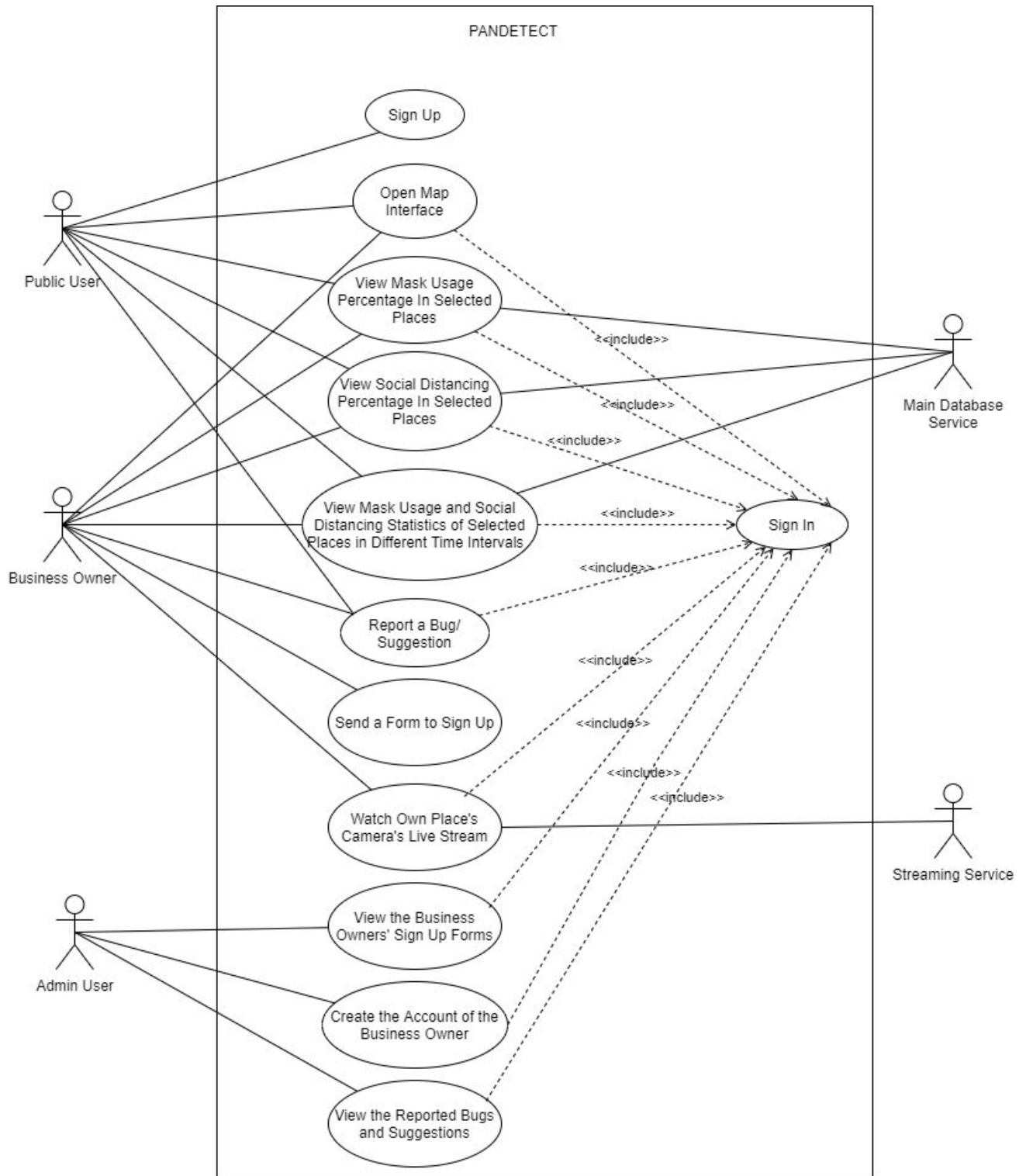


Figure 1: Use Case Diagram

In PANDETECT system, there are three types of users which are public users, business owners and admin users. The public users and business owners can both open the map interface, view mask usage percentage in selected places from the map, view social distancing percentage in selected places from the map and view mask usage and social distancing statistics of places in different time intervals such as in 1 hour, 1 day, 1 week and 1 month. They can also report a bug or make suggestions to improve the application. The statistics of mask usage and social distancing will be gathered from the main database service. Also, the public users can sign up to the system directly, however, business owners send a form to the admin user to sign up. Only if admin users accept this form, they can have an account. The business owners can additionally watch their own streams and streaming service provides this functionality. The admin users can view the business owners' sign up forms, create the account of the business owner and view the reported bugs and suggestions of other users. For all these functionalities, users must sign in to the system.

3.5.3 Object and Class Model

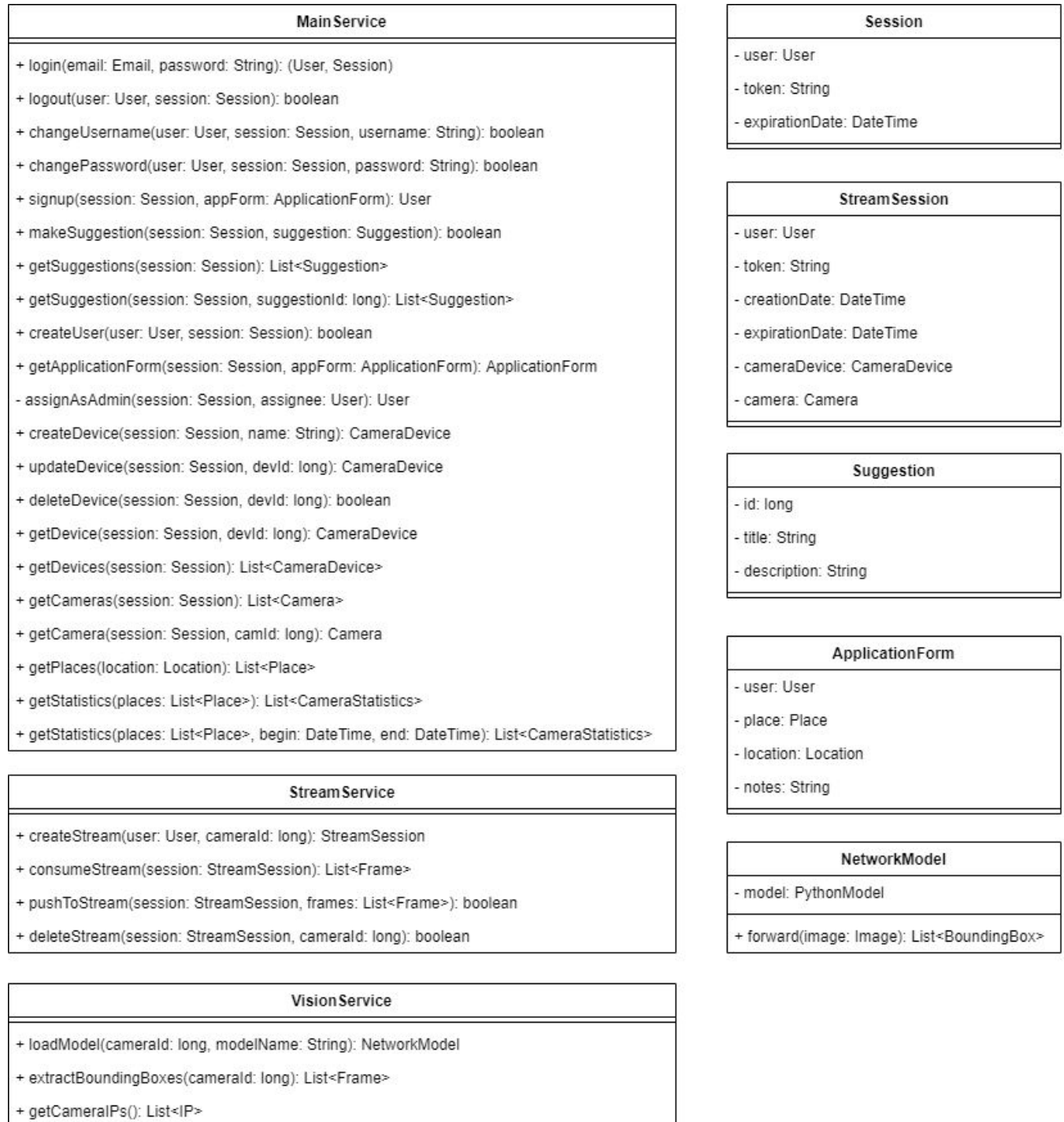


Figure 2: First Case of Class and Object Diagram

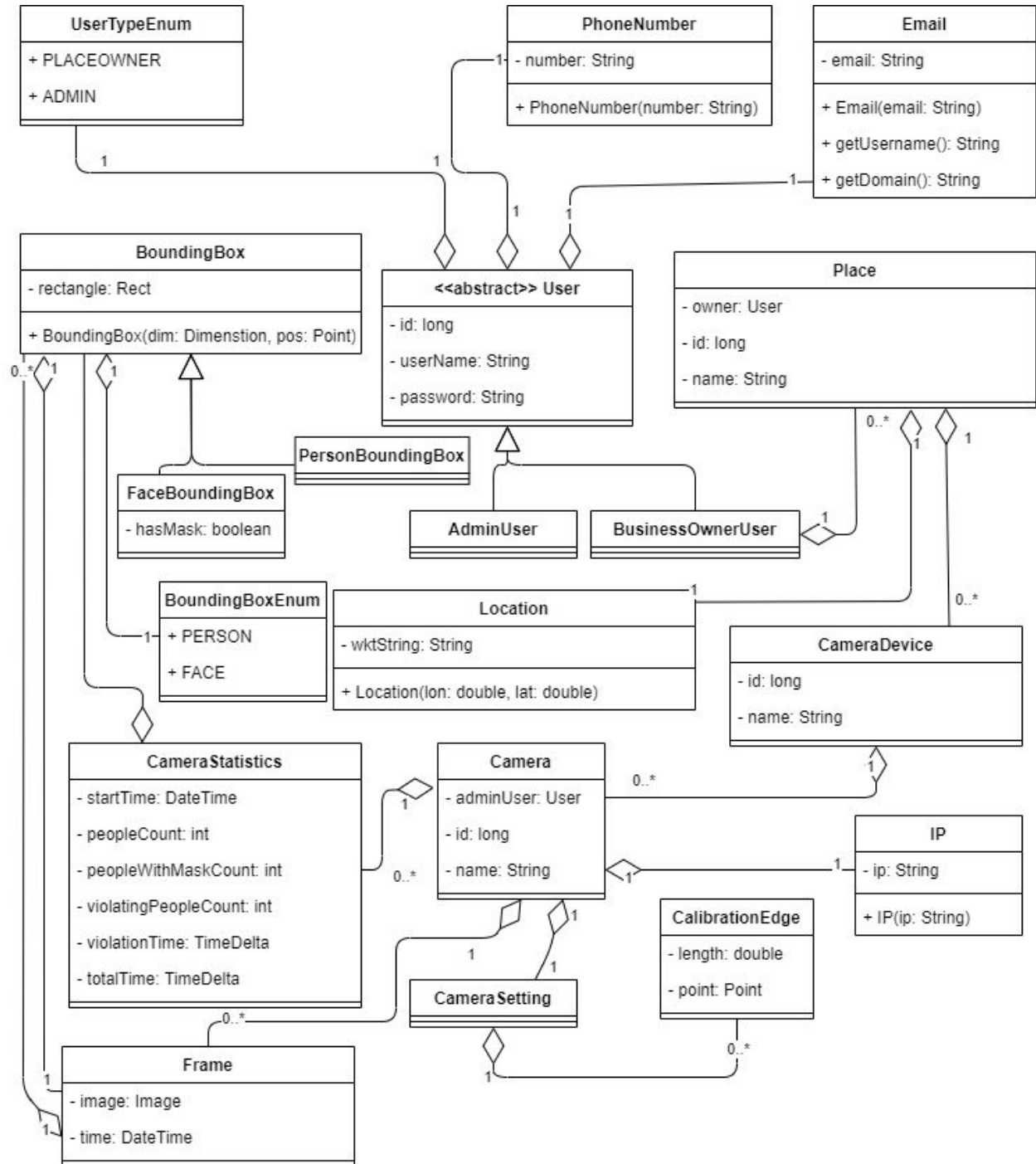


Figure 3: Second Case of Class and Object Diagram

3.5.4 Dynamic Models

3.5.4.1 State Diagram for Public Users

Figure 2 shows the state diagram for public users. Public users firstly open the application. If they have not signed up before, they sign up to the application as a public user. Then, they sign in and open the map interface that shows the location of places that the system has statistical information. The public users select a place and they view the mask usage and social distancing percentage of a place, view statistics of complying the rules of social distancing and mask usage in different time intervals, or report a bug/make a suggestion for the application. Then, the public user can exit the application or return to the map, continue selecting places from the map and see detailed information about the selected place.

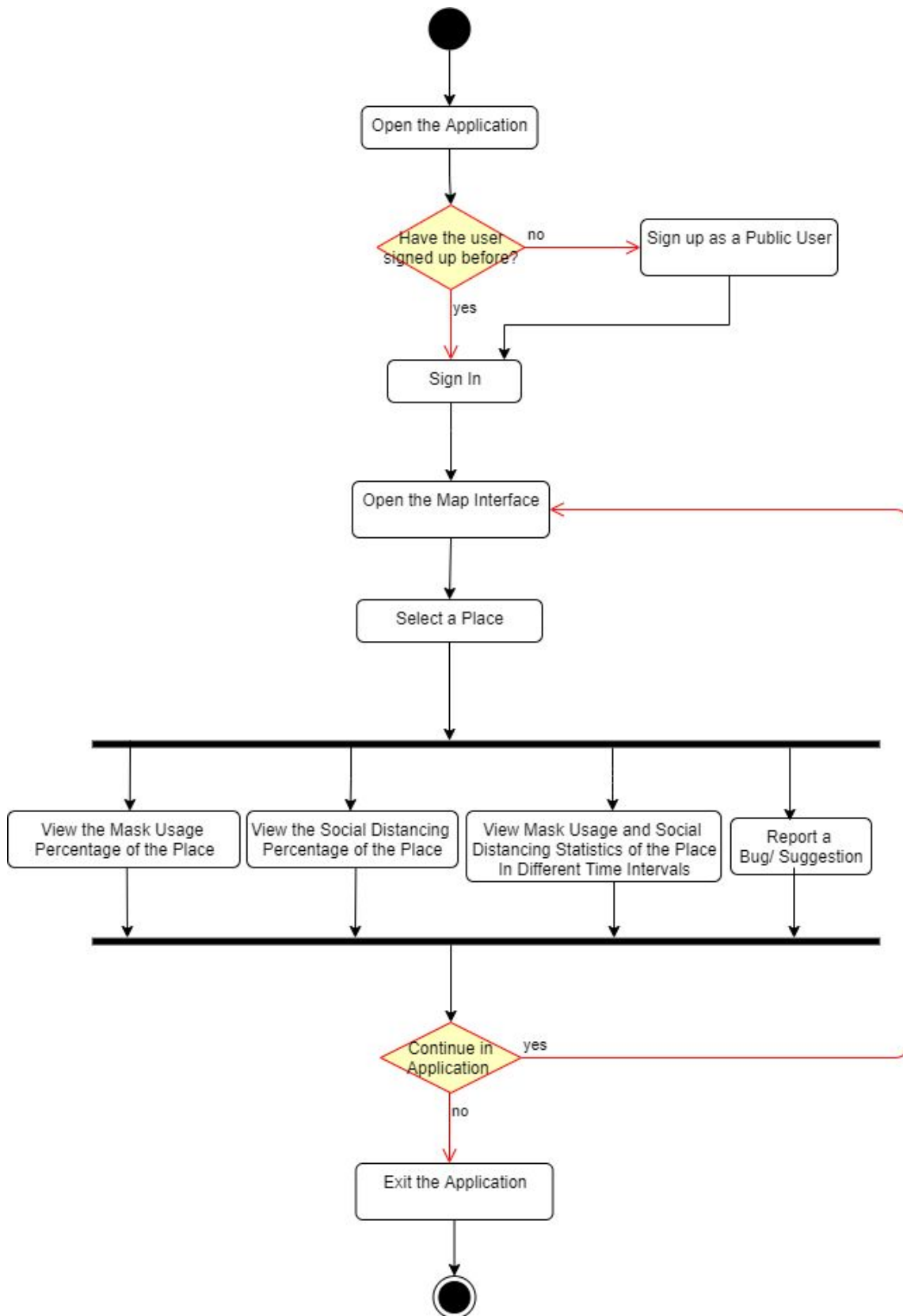


Figure 4: State Diagram for Public Users

3.5.4.2 State Diagram for Business Owners

Figure 3 shows the state diagram for business owners. They firstly open the application and send an application form to sign up to the system if they have not done that before. If the admin creates an account for them, they sign in and open the map interface showing the places. They can select a place from the map and view the mask usage percentage or social distancing percentage of the selected place, view statistics about the compliance to the mask usage and social distancing rules in different time intervals, or report a bug/make a suggestion for the application. They can also view their own businesses' statistics and watch their own camera stream. Finally, they can exit the application or continue by opening the map and selecting a place.

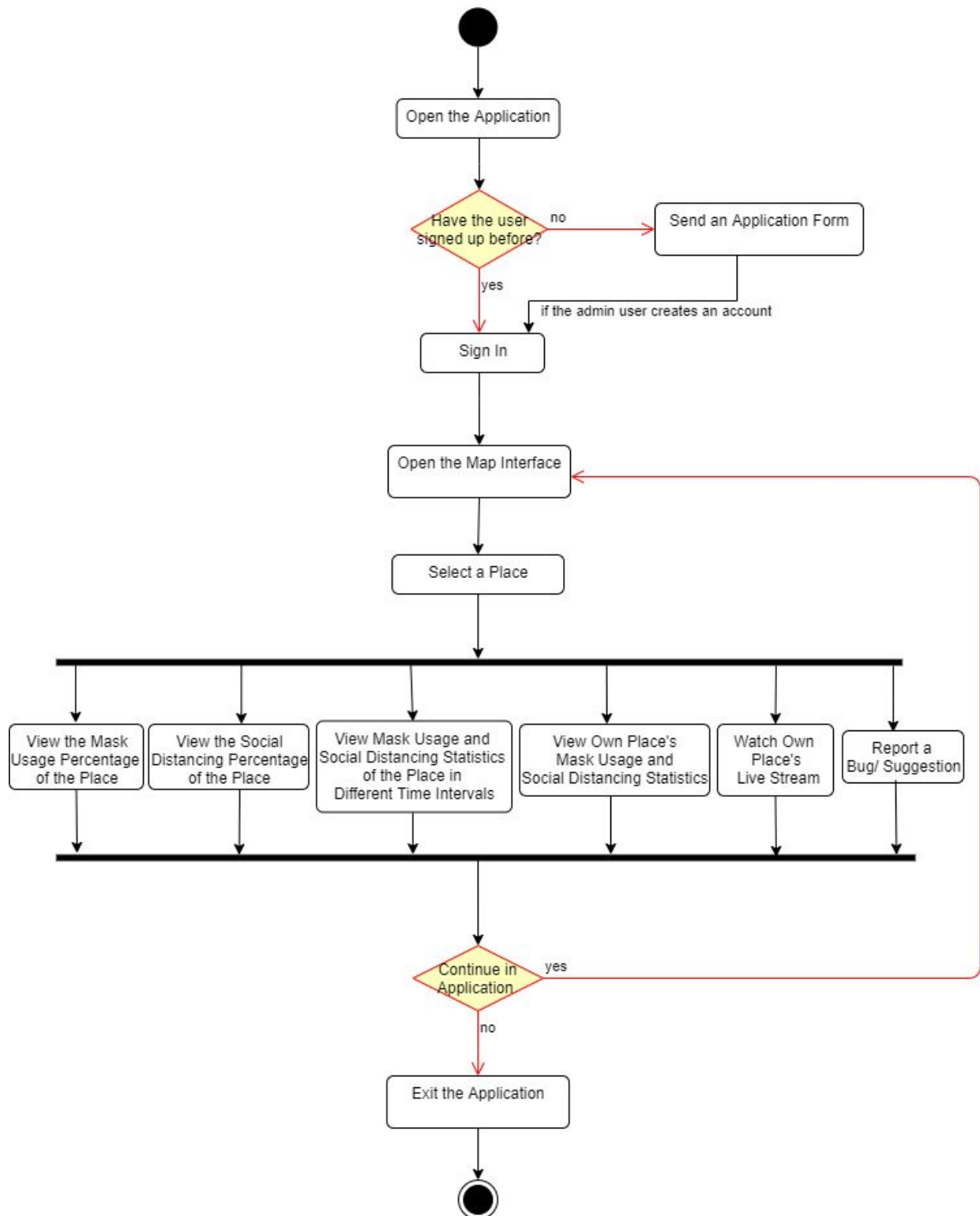


Figure 5: State Diagram for Business Owners

3.5.4.3 Activity Diagram

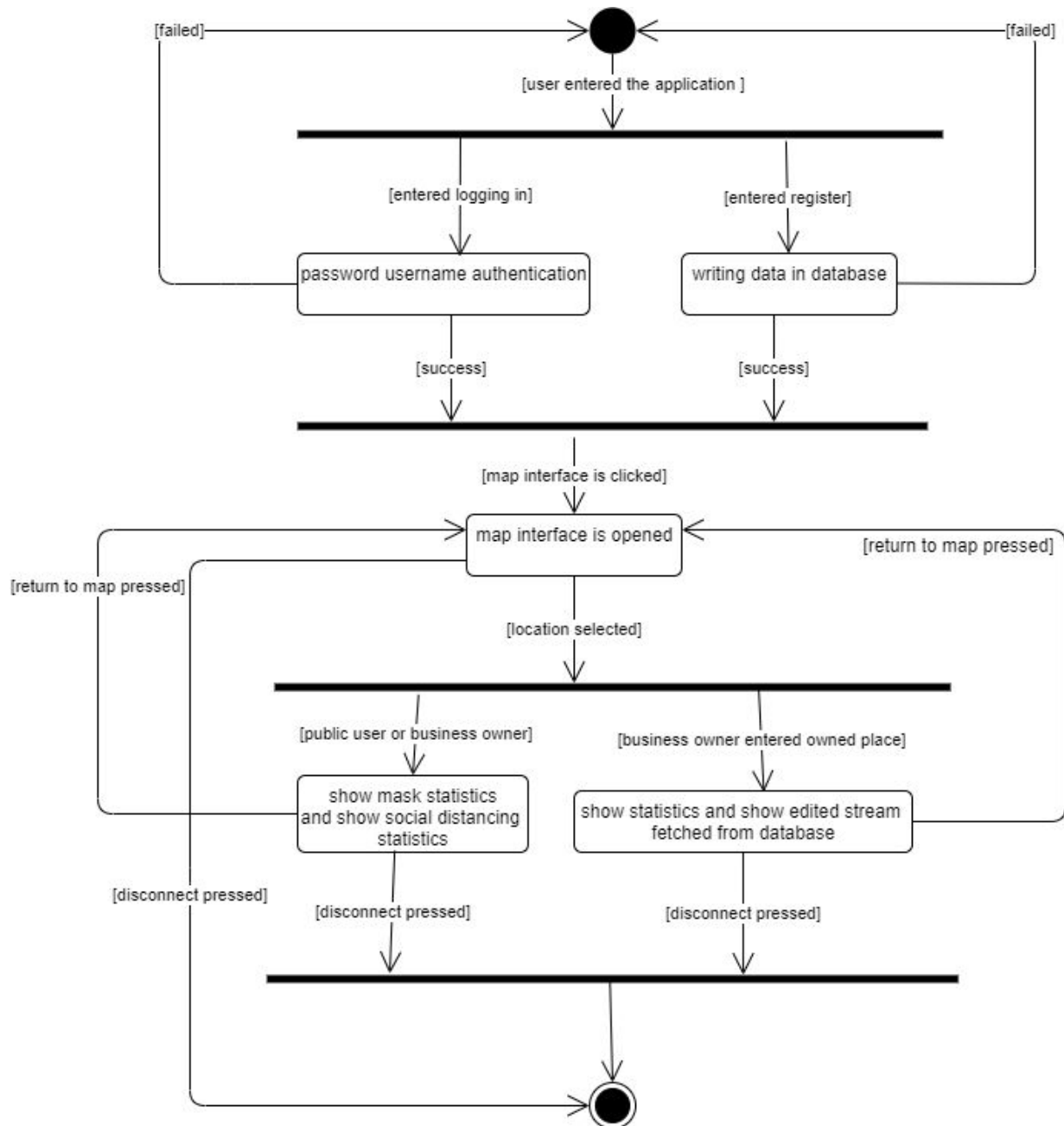


Figure 6: Activity Diagram

When the user enters the application, they are welcome with multiple options; one is logging in and other is signing up, if the signing up is selected by the user, it will fetch the data given by the user and write it to the database. On the other hand, if logging in is selected, the system will fetch the data and search for the username in the database, if the password is corresponding

with the password in the database it will authorize logging in, otherwise the system will reject and direct the user to logging in again. After these are succeeded, the system will direct the users to the map interface, standart, public users can select any location from the map interface and upon selecting a location system will give the statistics of how many people are not wearing masks or, how many people are not obeying the social distance. The place owners can also view these in logging in, but in addition to that, they can also see the live stream of their place's camera devices's stream. The stream includes drawing boxes for individuals who do not obey the rules on top of normal video. After these options users can disconnect or exit to the map interface in order to see another location.

3.5.4.4 Sequence Diagram of Public User Operations

Our first sequence diagram shows the logging procedure of a public user. The logging in is sent to the database for authentication. After authentication, it will direct the public user to the session, after that, the main service will call places in order to get the Maps API. With Maps API shown, the user can select a place and upon selection, the Main Service will fetch the data of cameras from that location. After seeing the result, the user exits the program.

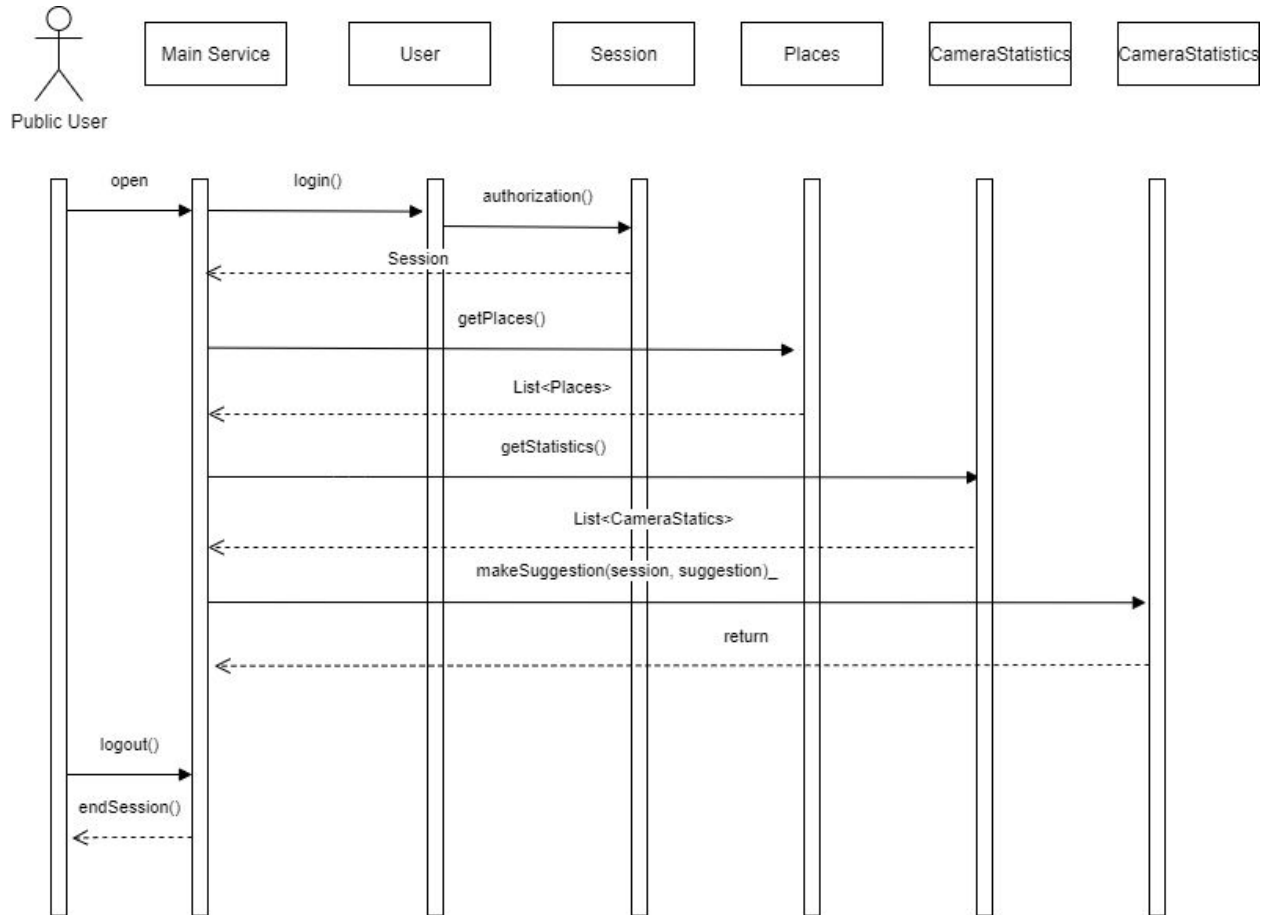


Figure 7: Sequence Diagram of Public User Operations

3.5.4.5 Sequence Diagram of Business User Operations

This sequence diagram shows the interactions of a business owner using our product. Initially, they can login to the system and a session is created for them, then they can either view their devices and cameras, or update them if they wish so. They can also get the statistics about mask usage and social distance compliance in their business. They can also get the general view from the map and see other places on Google Maps.

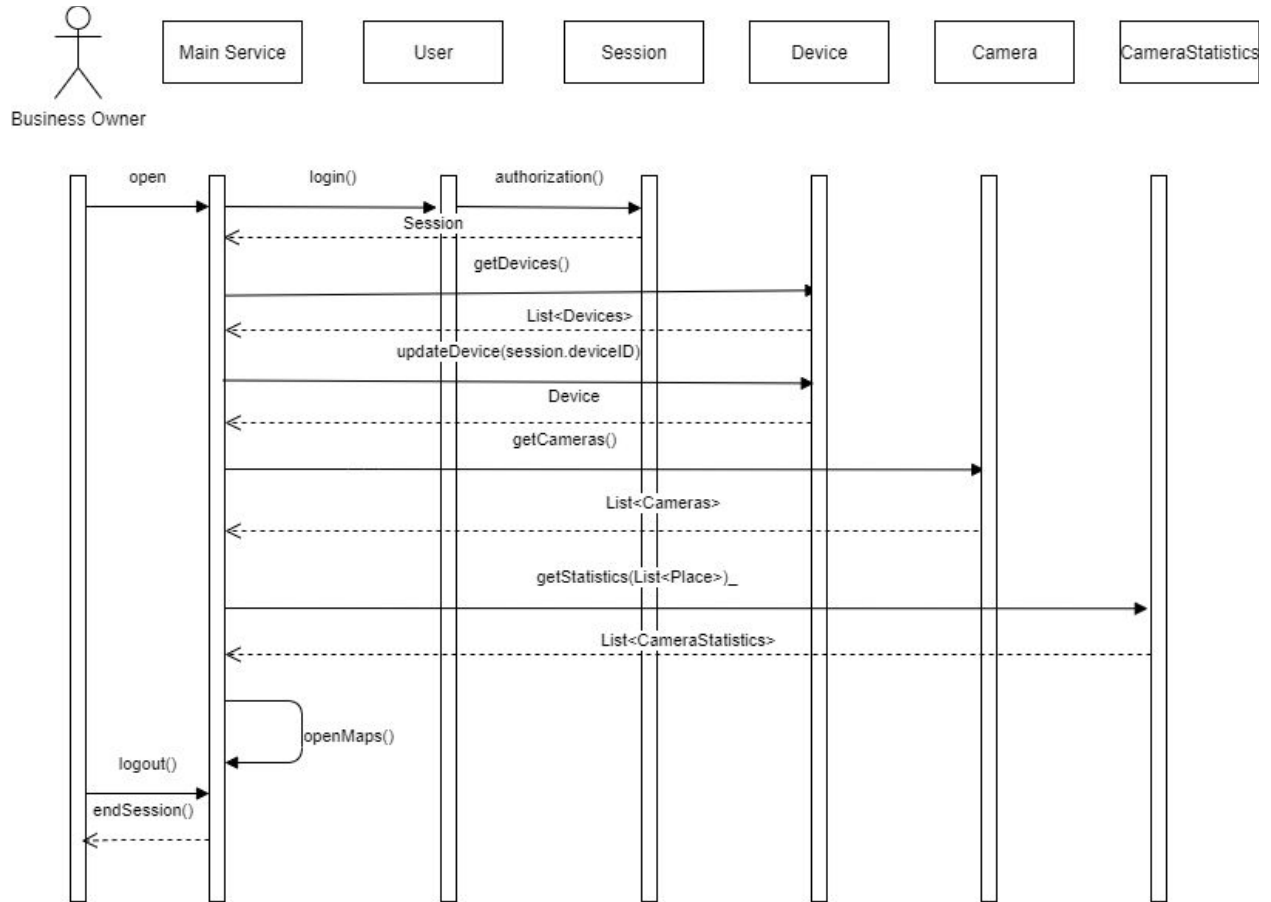


Figure 8: Sequence Diagram of Business User Operations

3.5.5 User Interface - Navigational Paths and Screen Mock-ups

3.5.5.1 Sign In Screen of Public Users and Business Owners

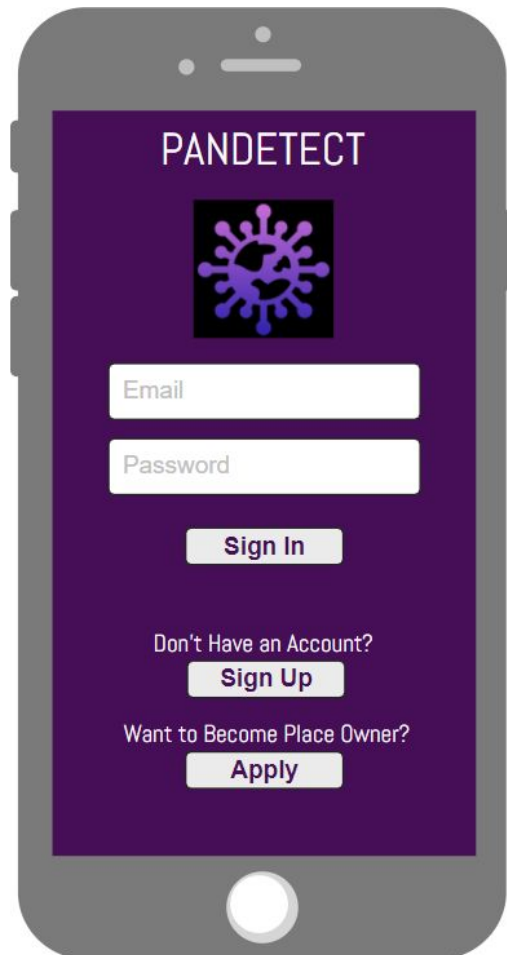


Figure 9: Sign In Screen of Public Users and Business Owners

This screen is the landing page of the mobile application. On this page, users must sign in with their email address and password. Users who do not have an account must first register with the “Sign Up” button. The “Apply” button will direct those who want to use our system in their own venues to the page where they can submit a form.

3.5.5.2 Sign Up Screen of Public Users

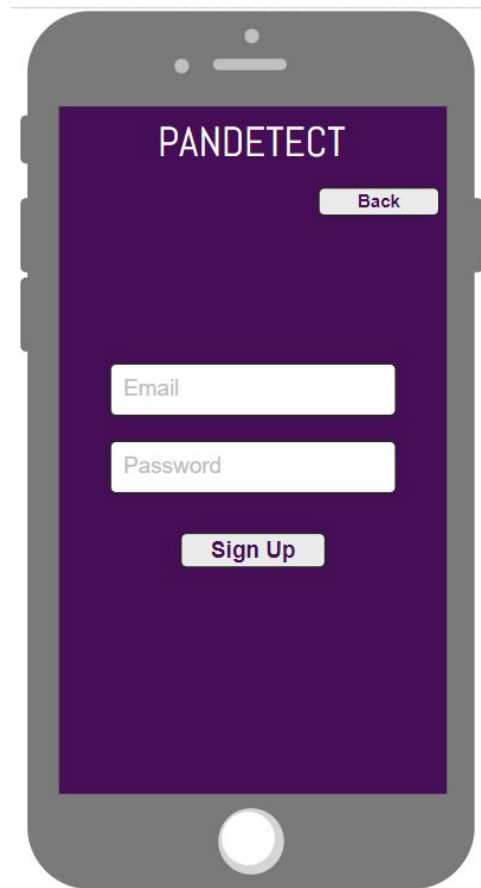
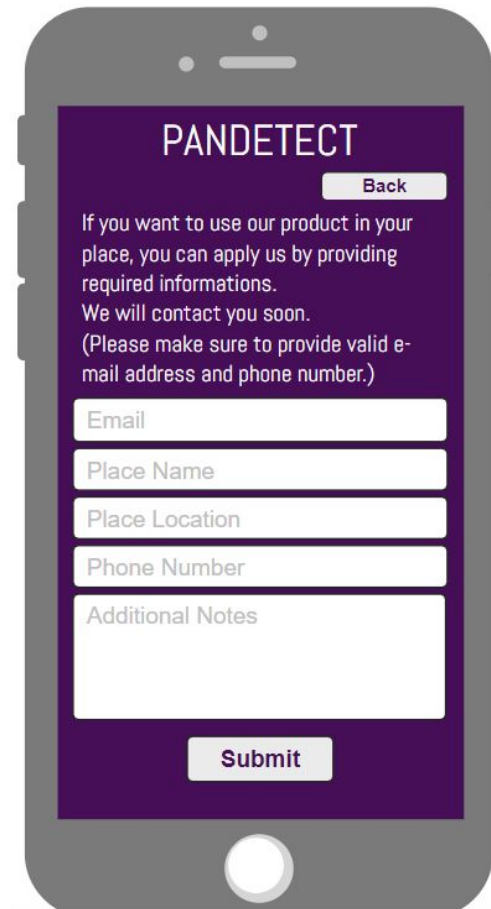


Figure 10: Sign Up Screen of Public Users

This screen allows public users who do not have an account to register to our application.

3.5.5.3 Application Form Screen of Business Owners



PANDETECT [Back](#)

If you want to use our product in your place, you can apply us by providing required informations.
We will contact you soon.
(Please make sure to provide valid e-mail address and phone number.)

Email

Place Name

Place Location

Phone Number

Additional Notes

[Submit](#)

Figure 11: Application Form Screen of Business Owners

This screen has been created for users who want to use our system at workplaces. Business owners will be able to contact us by providing the necessary information.

3.5.5.4 Main Screen

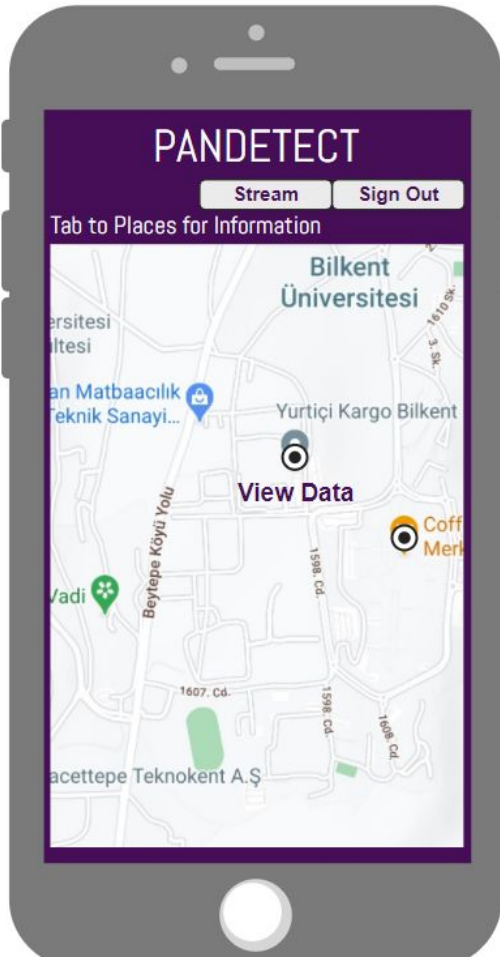


Figure 12: Main Screen of Business Owners

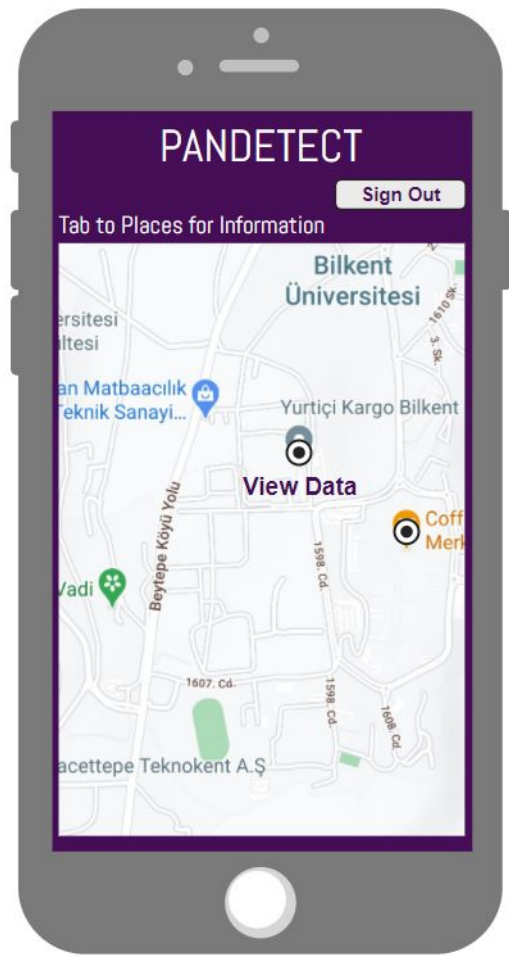


Figure 13: Main Screen of Public Users

This screen is the main screen that users will see after logging in. Users will be able to see the places that use our system by browsing on the map. When they touch the point they want to get information, the view data button will appear so that they can see the relevant data. By clicking this button, they will be directed to the page where they can access data for the relevant venue. Figure 12 is the screen that can be seen by the business owners using our system and it has a “Stream” button in addition to the public user screen. Figure 13 contains the view of the screen visible to public users. The “Stream” button will be made visible for business owners that are approved by us.

3.5.5.5 Current Data Screen of Public Users and Business Owners

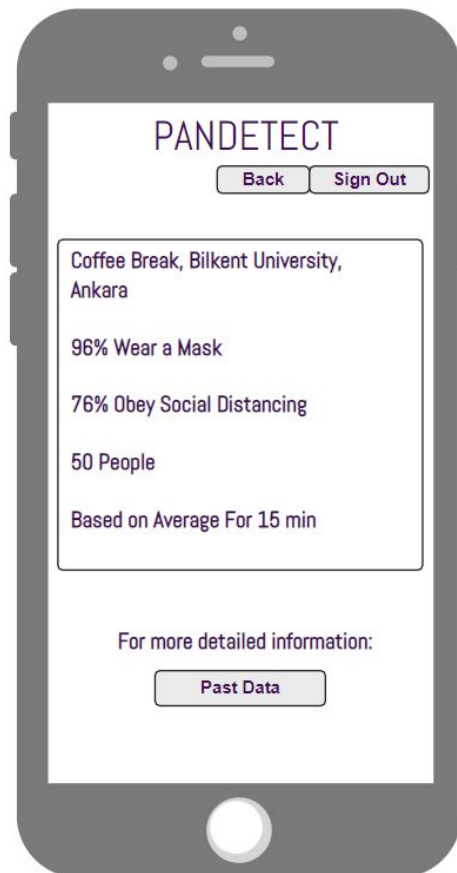


Figure 14: Current Data Screen of Public Users and Business Owners

This screen contains data about the place clicked on the map. Depending on the methods and algorithms we have determined, percentages for the compliance to the mask and social distance rules will be shown on this screen. The data will be calculated every 15 minutes for each place where the system is installed, and the average values will be shared accordingly.

3.5.5.6 Past Data Screen of Public Users and Business Owners

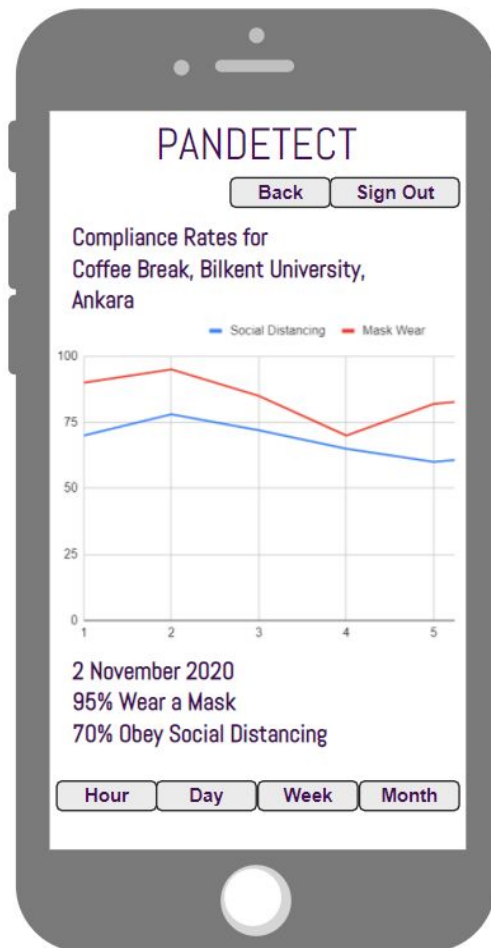


Figure 15: Past Data Screen of Public Users and Business Owners

This screen has been created for those who want to get more detailed information about the relevant area. It will graphically present the historical data of the place to be informed. Chart intervals can be adjusted hourly, daily, weekly, and monthly. Mask and social distance compliance floats will be shown on the y-axis, and range values will be shown on the x-axis. Values can be seen by scrolling left or right on the graph. In Figure 15, it is assumed that point 2 was clicked and sample values are shown according to that day.

3.5.5.7 Report Screen of Public Users and Business Owners

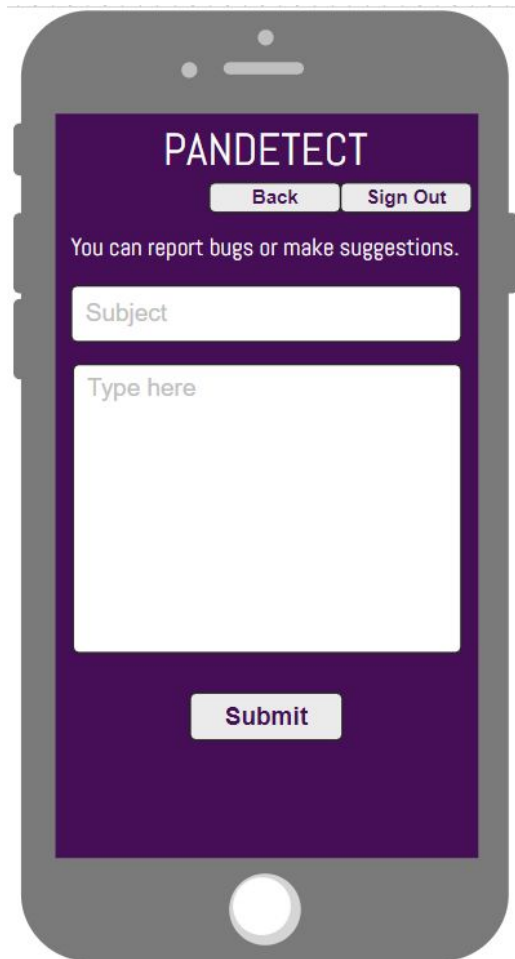


Figure 16: Report Screen of Public Users and Business Owners

Users who want to give us feedback/suggestions about the data in our application or report a bug will be able to send us their feedback from this screen.

3.5.5.8 Stream Screen of Business Owners

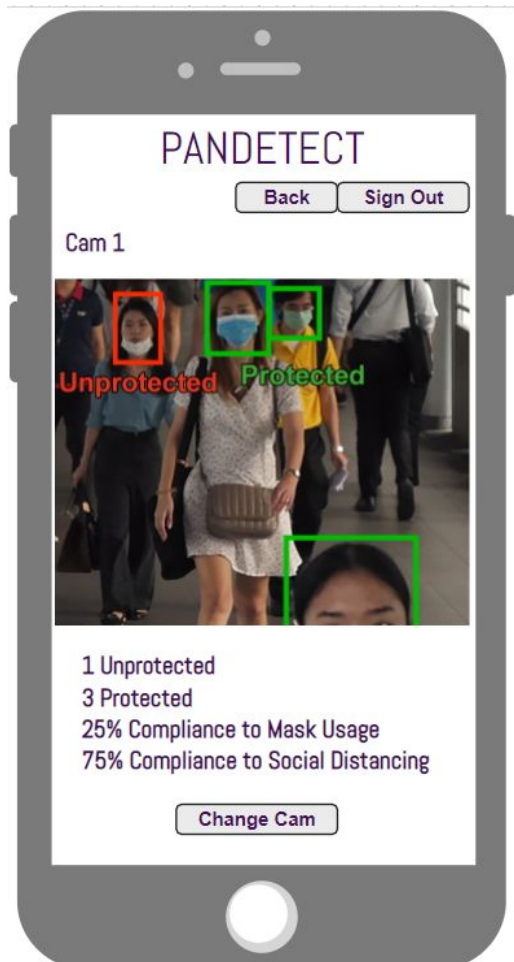


Figure 17: Stream Screen of Business Owners

This screen is only visible to the accounts that have been confirmed by us to provide data to our system which are the business owners. The camera streams of the place and the data provided for each camera are available on this page. Business owners will be able to switch between cameras in their places by pressing the “Change Cam” button.

3.5.5.9 Stream Screen of Business Owners in Desktop Application

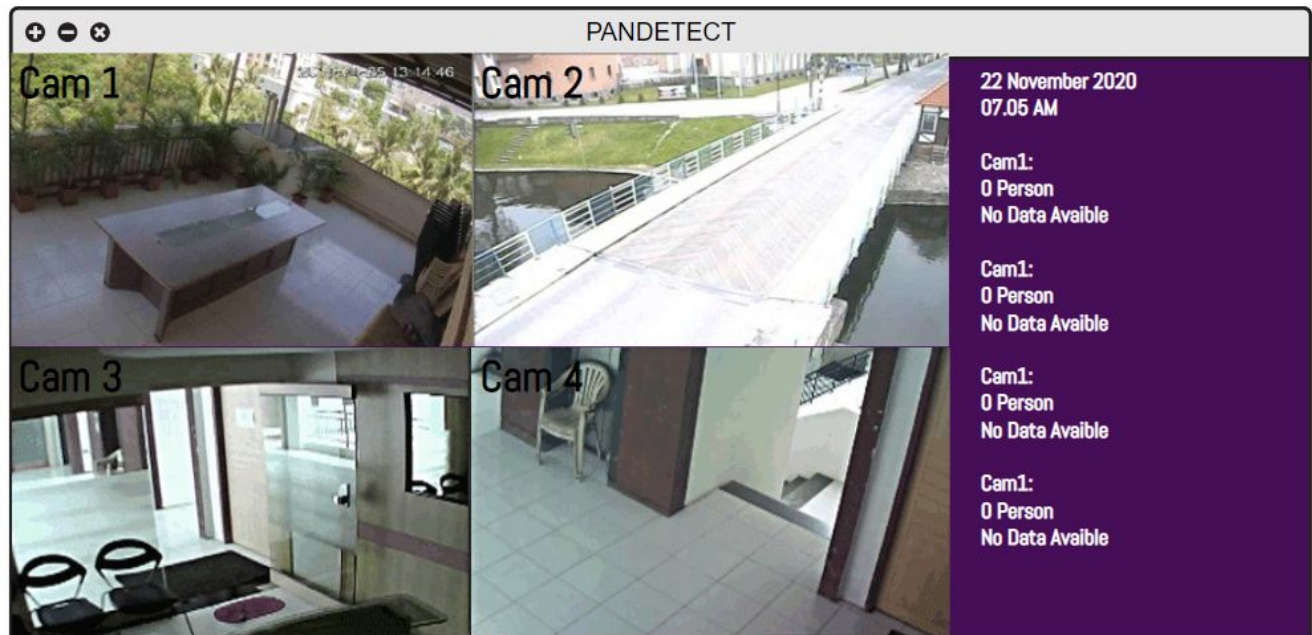


Figure 18: Stream Screen of Business Owners in Desktop Application

Since our system works as an integrated system, it also offers a simple-looking desktop application only for business owners apart from the mobile application. From the main computer to which the cameras are connected, that is, the data is provided, the camera views of the place will be presented instantly and the data will be displayed instantly on this screen. This screen will only be available to the owner of that place to watch the cameras and instant data from the place's computer which provides a wider screen size.

4. Other Analysis Elements

4.1 Consideration of Various Factors in Engineering Design

The factors which affect the end product PANDETECT will be analyzed below.

4.1.1 Public Health

The only problem that may affect the health in a negative way, may result from looking too much into the mobile phone's screen, which can harm and exhaust the eyes of the user if done too long without breaks. Other than this, the main aim of the application is to provide the user with information of the place's COVID-19 rules compliances. Thus, the users may decide not to go to the places where the rules are not complied, which will decrease their possibility of encountering the virus. The important point is to update the compliance percentages constantly to let the users make the decision of going to that place at that moment.

4.1.2 Public Safety

The data that the developers and administrators acquire will not be shared such as the username, password, email address. In addition, the video footage of the places will only be shared with its owner and these footage will not be shared publicly or used for other purposes.

4.1.3 Public Welfare

The application will not affect the total welfare of society in a negative manner. On the contrary, it may affect it positively since, with PANDETECT, the places such as restaurants and markets are supervised better. The places with high compliance to the mask usage and social distancing rules may attract more customers and their welfare might be affected positively.

4.1.4 Economic Factors

In order to make sure everyone can use the application, it will be free for the public users. For the business owners, the application will be charged, since the software and camera must be set up in their business place.

4.1.5 Social Factors

The application will highly likely increase the social interactions between individuals in the real world, since PANDETECT gives the location of relatively safe spaces via mask and social distancing statistics. Therefore, the distribution of people in a place might be affected due to users' decisions of going there or not.

4.1.6 Environmental Factors

PANDETECT will not be affected by environmental factors. Since it is a mobile application that uses cameras to gather required data, it does not have a direct connection with the environment. The used software or camera devices also will not have a negative impact on the environment.

4.1.7 Cultural Factors

The application is not affected by any cultural differences since it is only interested in the compliance of mask usage and social distancing rules. The rules will not differ between cultures, so, there is not a specific cultural factor which should be thought of.

4.1.8 Global Factors

The application can be used by any country in the world as long as the business place agrees to put a camera on their places and the software is set up properly. Thus, the application is not restricted to places in Turkey, it can be adapted to other countries easily. In addition, the application's language is English, which provides a wider range of users since English is a global language.

4.2 Risks and Alternatives

PANDETECT requires us to process the video in place. What that means is that, we shall process the data with the camera devices and then, send the data to our database. We are currently in search of a system that provides us to make data transfer between a camera device, a database and a mobile device. We have few alternatives like Redis, but we are not sure if that will solve the issue.

None of us (the developers) have worked a networking system this big and we could not conjecture what will happen in the end. We currently have not solved the issue of how we can go inside the NAT protection barrier in routers and select the desired device etc.

Finally, since the video stream processing will be done in mobile devices, we need to find libraries which can work with React Native framework, which none of us are familiar with.

Risk	Impact Level	Effects on the Project	B Plan
Real-time data transfer in streaming	5/10	We shall change our backend which will cost us some time.	The system will fetch data without drawings of libraries or, we will suffer a delay.
NAT barrier	9/10	If we could not access the mobile cameras there would not be anything to see in the application.	We can make the data go to a local mainframe and then distribute it to application.
Lack of libraries, vision functions	3/10	There are libraries for our use but if some functions are lacking we will implement them which costs us time.	Implement our functions for some libraries.

Figure 19: Risks Table

4.3 Project Plan

Work Package	Start Date	End Date	Leader	Members Involved
1	20.09.2020	21.10.2020	Selen Uysal	All Members
2	22.10.2020	31.12.2020	Batuhan Tosyalı	All Members
3	22.10.2020	05.12.2020	İrem Seven	All Members
4	22.01.2020	15.05.2021	Ufuk Bombar	All Members
5	22.01.2020	20.05.2021	Berk Güler	All Members
6	21.05.2020	30.05.2020	Selen Uysal	All Members

Figure 20: Project Plan Table

4.3.1 Work Package 1: Design Analysis and Specifications

In this stage, we as a group devote ourselves in order to finish the design, analysis and specifications.

TASK1: First task was to find and discuss the project idea which we ended up with PANDETECT application.

TASK 2: After that, we determined what kind of technologies and concepts/areas we need to know to develop the system. We decided that we need Computer Vision Libraries such as OpenCV, a maps API in order to see and interact with apps which is Google Maps API, and a backend system written in TypeScript in order to communicate with devices and send streams in real time which will be Redis. Also, a framework in order to make the application work in all of the mobile operating systems is needed and it will be React Native.

TASK 3: Since the systems and services determined, the use cases, a rough class diagram and dynamic diagrams such as activity and state diagrams should be demonstrated in order for our supervisors and clients to understand the concepts which are decided to use.

TASK 4: Then, we need to divide the workload among us. The proper workload for now is ambiguous, for now we decided that Redis will be implemented by Ufuk Bombar, React Native will be written by Berk Güler and Batuhan Tosyalı and OpenCV will be implemented by İrem Seven and Selen Uysal.

TASK 5: Lastly, the decision of which version control to use is made, which is Git hosted on Github.

DELIVERABLES: Analysis & specification reports, mockups which are subject to change along with distribution of work among group members to obtain maximum efficiency.

4.3.2 Work Package 2: Prototype Implementation and High-Level Design

Here, we will implement the barebones of the system in order to generate a prototype.

TASK 1: Implementing OpenCV algorithm in order to catch the individuals in the camera, detect if they have masks, detect if there are individuals who do not reside social distancing, and create statistics according to that.

TASK 2: Simultaneously, working with Redis is vital, since we need to fetch the data obtained from OpenCV and we need to save statistics. The important part is that we also may need to stream the data for the business owners to see the camera's stream. Therefore, implementing Redis has vital importance.

TASK 3: It is not decided who would implement Google Maps API but, we also need to work on it parallelly in order to achieve applications main purpose which is using a map interface.

TASK 4: Meanwhile, we also need to design an interface for the application, which should be done in React Native in order for the app to be cross-platform.

TASK 5: We also need to work on high-level design documentation while implementing the system. We need to divide the system into smaller subsystems and select strategies to make communication between systems easier.

DELIVERABLES: High level design report will be completed and also, Minimum Viable Product (MVP) will be available for presentation including a mobile & desktop application.

4.3.3 Work Package 3: Testing the Prototype

Here we will find the bugs, limitations, glitches in the system for a couple of days.

TASK 1: We need to work really hard in order to find the bugs in the system, especially we need to double check if there are any privacy issues.

TASK 2: We need to test the system to its best in order to see at most how many users can open the application at the same time without making the server dysfunctional.

TASK 3: We need to see how many cameras that we can have live stream.

DELIVERABLES: Almost bug free application will be obtained after continuous testing and bug fixing.

4.3.4 Work Package 4: Implementation

For the rest of our time, we shall finish implementation properly with now known limits.

TASK 1: If the OpenCV algorithm could be improved in terms of efficiency, we will improve it.

TASK 2: If the capacity of cameras and users could be increased in any way, we shall focus on increasing it, since we aim that our product can/will be used worldwide.

TASK 3: The OpenCV algorithms should be implemented with every angle and with every level of light and resolution in multiple places before deploying the product.

TASK 4: The low level design documentation will be written during the implementation.

DELIVERABLES: Low level design report will be completed. A fully functioning application will be available.

4.3.5 Work Package 5: Testing the System

Here we will find the bugs, limitations, glitches in the system for a couple of days.

TASK 1: Again, we should check for the presence of bugs in the system..

TASK 2: We shall determine an upper bound for number of users and number of streams at the same time for the end product and deploy the product accordingly.

TASK 3: Testing of the OpenCV algorithm in various places such as cafes, restaurants etc. to find remaining bugs (if any left) and fixing it.

DELIVERABLES: Bug free application will be obtained after continuous testing and bug fixing.

4.3.6 Work Package 6: Release of the System

TASK 1: After the deployment of the system, a new report consisting of unambiguous diagrams and final version of the class diagram and the dynamic diagrams are released.

TASK 2: Lastly, we will prepare for a proper presentation for the faculty members.

DELIVERABLES: Final report and presentation slides are completed.

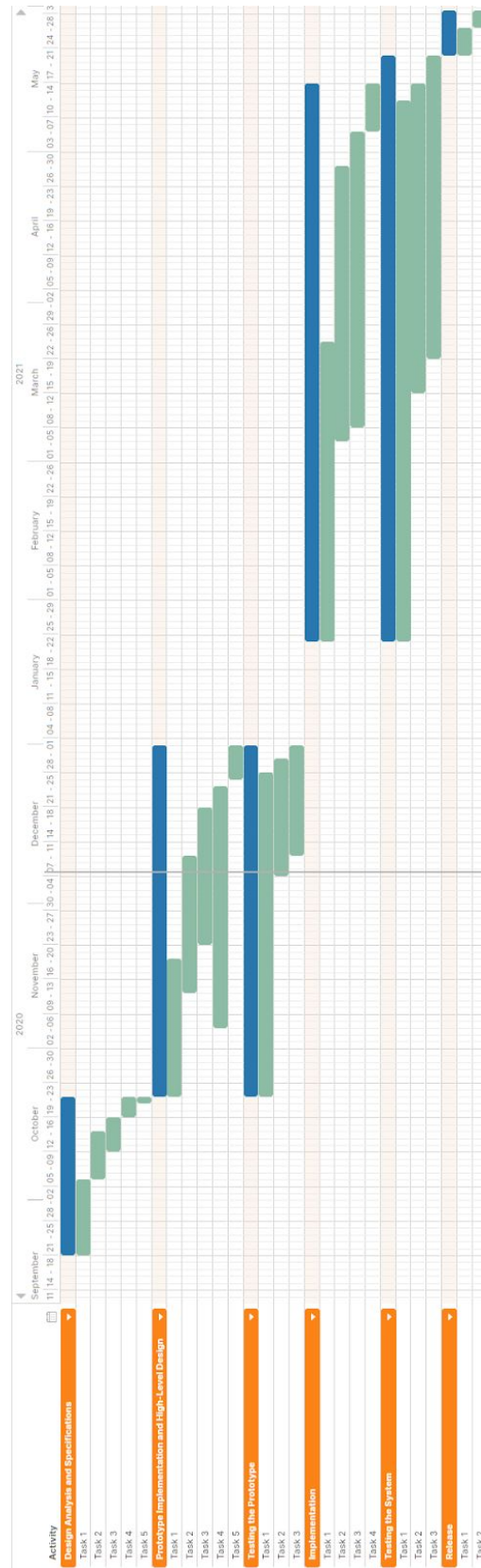


Figure 21: Gantt Chart

4.4 Ensuring Proper Teamwork

In order to make everyone participate we divided the total work among us. We divided the total work such that everyone could specialize in a certain subject. We have distinct subjects that need separate attention such as: Networking, Database Management, Computer vision (and may be a bit of Machine Learning). Furthermore we will be using huge frameworks and APIs such as Google Maps API and Native.js. To sum up, we will need to distribute these huge subjects among us, for example, every one of us will work mainly in a subject and be the lead developer of that subject whereas he/she will also take participation in the other roles as a minor developer.

4.5 Ethics and Professional Responsibilities

The data acquired throughout the application will not be shared with any companies, systems etc. The data that will not be shared includes, username, password, location, email of the user and the video streams that have been shot by camera devices. We will be getting permissions just to save the data to our database, not to use it in any place where this data is used should be seen when users sign in for the first time to the application. Also, users will be informed about the permissions when users first sign in.

Our system will abide by the rules of KVKK (Kisisel Verilerin Korunmasi Kanunu), therefore, there should not be any concerns about ethics in PANDETECT.

Lastly, while developing the system we would like to avoid copyright issues, therefore, we will be using open source libraries for the most time and will be getting permissions from copyright owners if needed.

4.6 Planning for New Knowledge and Learning Strategies

As explained above, there are few areas of computer engineering/science that will be used here, such as database systems, networking and computer vision. In terms of networking, most of us have theoretical knowledge about networking but none of us used any networking libraries, especially in mobile development. Therefore, we should read documentation about networking in mobile platforms. But that is not it, we also need to find a robust library in order to fetch image data and send it to mobile applications in order to get the statistics and stream with bounding boxes depending on which type of user you logged in as. Lastly, we as group members need to increase our knowledge for React Native and API's like Google Maps API.

For these, we will study from their related documentation and watch tutorials on Coursera and YouTube. YouTube will be especially helpful to us, since there are a lot of tutorials teaching from scratch. Thus, we will be using an online learning strategy. Since we will mostly rely on the sources that we find on the Internet and ask our supervisor for key points when needed, our main learning strategy will be self directed learning. We will also be using visualizing learning strategy since we are drawing UML diagrams showing the system's classes, objects and dynamic relations, which will help us to visualize better and start implementation without any misunderstandings between group members.

5. Glossary

React Native: It is an open-source mobile application framework in order to write code for Android, iOS and Windows which is developed by Facebook [6].

OpenCV: (Open Source Computer Vision Library) is an open source computer vision and machine learning software library [4].

Redis: (Remote Dictionary Server) is a data structure server in order to hold the data in the RAM. Redis is also an open-source NoSQL based database system [7].

TypeScript: TypeScript is an open-source project backed by the tech giant Microsoft. Its basic premise is JavaScript with types. TypeScript compiles directly to JavaScript [5].

6. References

[1] “*Face Mask Detection*,” Accessed on: Nov .21, 2020. [Online] Available:

<https://www.xovis.com/en/products/detail/face-mask-detection/>

[2] “*MAGGY social distancing safeguard device*,” Retrieved Nov. 21, 2020. [Online] Available:

https://www.maggylife.eu/product/maggy-social-distancing-safeguard-device-free-lanyard/?gclid=CjwKCAiA7939BRBMEiwA-hX5J4B3EADE_zAaD98mk5gBMq6J0rwibDVpLol7hrfZBwp-Ohm6zOa3xRoCeJcQAvD_BwE

[3] “*FACE MASK DETECTION SOFTWARE*,” Accessed on: Nov. 21, 2020. [Online] Available:

<https://www.aerialtronics.com/en/products/face-mask-detection-software#featuresfacemask>

[4] “*About OpenCV*,” Accessed on: Nov. 21, 2020. [Online] Available: <https://opencv.org/about/>

[5] “*About OpenCV*,” Accessed on: Nov. 21, 2020. [Online] Available: <https://www.typescriptlang.org/>

[6] “*React Native*,” Accessed on: Nov. 21, 2020. [Online] Available: <https://reactnative.dev/>

[7] “*Redis*,” Accessed on: Nov. 21, 2020. [Online] Available: <https://redis.io/>