MANİSA
CELAL BAYAR
UNIVERSITY

CSE 3105 / CSE 3137

OBJECT-ORIENTED ANALYSIS AND DESIGN

FALL 2022

COURSE PROJECT: *SafeHomeSystem*

Requirements Analysis Document

Group 21

Yaren Mamuk - 190315021

Uğurcan Çırak - 190315051

Merthan Erler - 190315033

Sefa Altınok - 190315020

Hurşit İçke - 190315068

Elif Merve Arslan – 180315022

4 December 2022

Table of Contents

1 Introduction

It is a project created so that we can get immediate support when a thief enters our house. By detecting the thief's entry situation, we have turned the actions that need to be taken into a project. Our aim is to prevent the loss of property and life when the thief enters, and to bring the criminals to justice with the help of law enforcement officers. We decided to develop the project because we thought that the systems used at the moment did not provide sufficient security. We believe that we will provide high security thanks to the system we have established. We will provide the necessary improvements for our project by evaluating its negative and positive aspects.

2 Current System

The systems we currently use are in a poor state of security. Because the system is based on warning the people in the house by making a sound only when the thief enters the house. This system can be easily disabled. It's a highly vulnerable system. The system we are considering to design is a system with strict security measures and following certain action paths in case of the slightest problem. Its biggest advantage is that it generates an emergency call when the required sequence of operations is not applied. It is also a system that raises awareness of the landlord. Even if the thief has not entered the house, if the owner has left the door or window open, the system is automatically activated and a notification is sent to the owner, and if the owner has left it open, he takes the necessary action by closing the door or window immediately. The system we designed is more secure than the systems currently used, and it also has many advantages.

3 Proposed System

<The third section, *Proposed system,* documents the requirements elicitation and the analysis model of the new system. It is divided into four subsections:>

3.1 Functional Requirements

Login with id and password after user registration.

The user can control the camera and sensors through the system.

The homeowner can report a burglary incident on the system.

The homeowner by entering his/her password, it notifies the system that there is no case.

The system sends a notification to the user in case of a possible burglary.

The system sends a notification to the homeowner after reporting the incident to the police.

3.2 Nonfunctional Requirements

Response time should be less than 3 minutes. (PERFORMANCE)

The system must be available 24 hours a day. (PERFORMANCE - RELIABILITY)

If the homeowner does not inform the system 10 minutes, the system will inform the police station that there is an incident. (PERFORMANCE)

The system can be use by everyone easily, also interface is too clear, so users don’t need any additional documentation. (USABILITY)

The system has rapid response time. The system locks all doors if someone broke into the house. (PERFORMANCE - RELIABILITY)

3.3 System Models

3.3.1 Scenarios

SCENARIO-1

1- Mahmut's house has an emergency system to prevent burglary. Mahmut lives in the state of Florida and there are a lot of hurricanes here.

2- On a Sunday, Mahmut is at home. The weather outside is heavy thunderstorms and torrential rain.

3- The window of Mahmut's house is broken due to the sound caused by a lightning bolt falling near Mahmut's house.

4- While Mahmut is trying to understand what is happening, information comes to his phone about the glass being forced to open

5- When Mahmut goes to look at the window, he realizes that the window is broken.

6- By entering the password to the system, it informs the system that there is no burglary incident, and the system does not report anything, the system does not notify the police and security.

PARTICIPATING ACTORS	Initiated by lightning Communicates by homeowner
FLOW OF EVENTS	1. The system detects that someone forces to open the windows or doors. 2. The system sends a message to the homeowner that the doors or windows are being forced. 3. The homeowner looks at the incoming message. The homeowner goes to check the window because he/she is at home. 4. The homeowner understands that the window is broken and that's why the system sends a message. 5. The host enters his password through the system and informs the system that there is no burglary incident. 6. The system understands that there is no burglary incident and does not notify the police station 7. The system gives feedback to the homeowner about that there is no burglary incident, and it does not notify the polis station.
ENTRY CONDITION	1. An object hit the window of the house and the window is broken
EXIT CONITION	1. The homeowner decides that there is no burglary incident and enters his/her password into the system.
QUALITY REQUIREMENTS	1. The homeowner enters his/her password less than 3 minutes.

SCENARIO-2

1- Ahmet, the homeowner, uses the emergency system that prevents burglary. Ahmet activates the sensors before leaving the house and completes the configuration of the system properly.

2- Ahmet leaves the house to go to work. After a while, Veli, the burglar, starts to watch Ahmet's house and begins to make plans to break into Ali’s house.

3- After Veli realizes that no one is at home, he tries to open the outer door of the house and forces it.

4- Sensors detect that the door is being forced and the system sends a message to Ahmet's phone.

5- Sensors detect that the door is forced by the burglar and the system sends a message to Ahmet's phone**.**

6- The burglar enters the house by forcing the door. After the burglar broke into the house, the system locks all the doors in the house. It causes the burglar to be locked in any room. While this happened, the police save time in order to catch the burglar, it also causes the burglar to waste time**.**

7- After Ahmet receives the message, he/she checks his/her house through the cameras on the system. After Ahmet monitors the cameras, Ahmet realizes that someone has entered his house and report a burglary incident through the sytem.

8- The system asks Ahmet to enter how many people there are at house if there is someone at the house.

9- After Ahmet creates the case through the system, both the police station and the security of the site are informed through the system. Site security takes the necessary measures until the police arrive.

PARTICIPATING ACTORS	Initiated by burglar Communicates by homowner, security and police station (or system)
FLOW OF EVENTS	1. The system detects that someone forces to open the windows or doors. 2. The system sends a message to the homeowner that the doors or windows are being forced by someone. 3. The burglar breaks into the house by forcing to the door. 4. After the system realizes that someome broke into the house, the system locks all the doors at the house. The system sends another message to the homeowner that the system has locked all doors in the house. 5. The homeowner looks at the incoming messages through the system. Then the homeowner controls the cameras in the house through the system. 6. The homeowner realizes that there is someone that broke into the house and creates a burglary incident through the system. 7. The system asks the homeowner to enter the number of people present if there is someone at the house. 8. The homeowner enters the number of people who are at the house. Then the homeowner reports the incident to the system. 9. After the system receives the burglary incident, the system notifies the burglary incident by calling the police station. 10. After the system notifies the police, the system calls the security number that the homeowner gave when registering and reports the burglary incident to the security. 11. The system gives a notification to the homeowner that the system has successfully forwarded the burglary incident to the police station.
ENTRY CONDITION	1. The burglar forces to open the door or window.
EXIT CONITION	1. After the system reports the case to the police station, it gives feedback about the reported burglary incident to the homeowner
QUALITY REQUIREMENTS	None

SCENARIO-3

1- Hazar’s house has an emergency system to prevent burglary. Hazar gets stressed due to his busy work life. Thereupon he takes leave to rest. He plans to stay a little away from the city with his family and go to the village. Because the weather is hot, they leave one window of the house open and go to the village.

2- After a certain period of time, Hazar's phone receives a notification that the windows remain open.

3- But because Hazar is on the way, he cannot see the notification sent by the system and enter his password into the system.

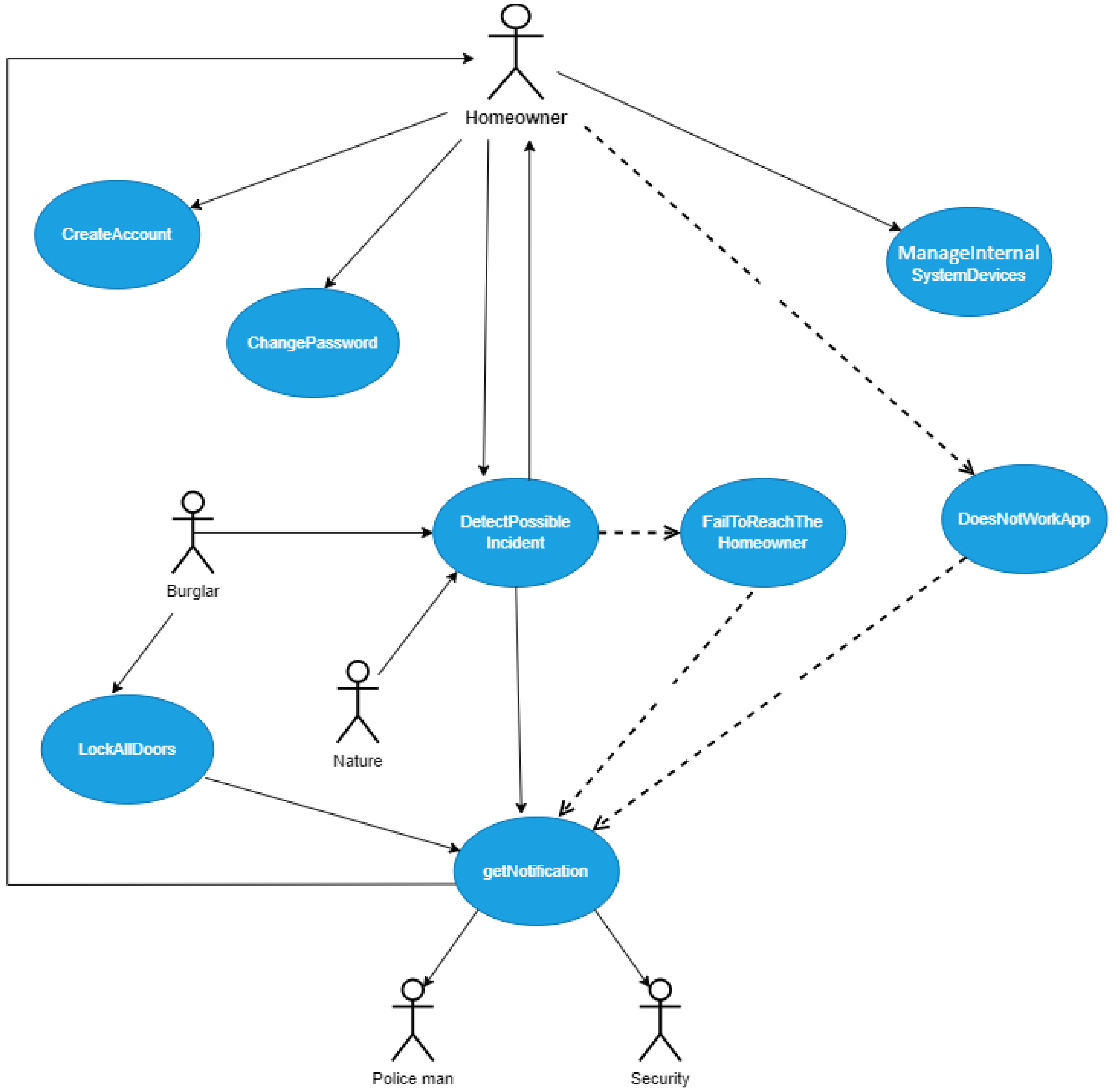
4- The System calls Hazar from the phone because Hazar did not respond to the notification on time.

5- Hazar answers the phone and informs the system that he left the window open deliberately.

6- The system figures out that there is no burglary incident and does not notify the police and security.

PARTICIPATING ACTORS	Initiated by opened window Communicates with homeowner
FLOW OF EVENTS	1. The system detects that the window has been open for a long time. 2. The system sends a message to the homeowner informing him/her that the window has been open for a long time 3. The homeowner cannot look at the incoming notification in time. 4. The system calls the homeowner’s phone number because the host has not entered his/her password or reported a burglary incident. 5. The host answers the phone. The host verbally informs the system that he left the window open on the phone. 6. The system figures out that there is no burglary incident and does not notify the police station.
ENTRY CONDITION	1. The window of the house is left open for a long time.
EXIT CONITION	1. The host decides that there is no case of theft and verbally informs that there is no burglary incident.
QUALITY REQUIREMENTS	1. The homeowner needs to answer the system phone call less than 10 minutes.

3.3.2 Use Case Model



<u>DetectPossibleIncident</u>, <u>ManageInternalSystemDevices</u> and <u>LockAllDoors</u> use cases are critical use cases for the system.

THE SYSTEM FEATURES

USE CASE: CreateAccount

USE CASE NAME	CreateAccount
PARTICIPATING ACTORS	Initiated by The Homeowner
FLOW OF EVENTS	1. The homeowner completes the create of an account process. 2. The homeowner types userID , password and security phone number (If any) 3. System sends a verification e-mail. 4. The homeowner log in its e-mail and verify the account. 5. System validates the homeowner account.
ENTRY CONDITION	1. The homeowner opens the system
EXIT CONDITION	1. The homeowner has created his/her account successfully and the system validate the homeowner's account.
QUALITY REQUIREMENTS	1. If the userId that the homeowner has entered has been alreday taken, the system will return an information message and inform the homeowner that the userId has already been taken by another user. 2. The password length must be at least 8, and it needs to include at least one capital letter. If the password does not provide the requirements, the system asks the homeowner to enter another password.

USE CASE: ChangePassword

USE CASE NAME	ChangePassword
PARTICIPATING ACTORS	Initiated by The Homeowner
FLOW OF EVENTS	1. The homeowner enters his or her userID and password. 2. The homeowner enters his or her current password and new password. 3. The system sends an e-mail to verify changing the password 4. The homeowner logs in its e-mail and verify the change.
ENTRY CONDITION	1. The homeowner opens the system
EXIT CONDITION	1. The homeowner has created his/her account successfully
QUALITY REQUIREMENTS	1. If the homeowner enters inappropriate userId and password combination, the system sends an information message. If the homeowner enters inappropriate userId and password combination 3 times, the system will block the homeowner to enter userId and password for 10 minutes. 2. If the new password is the same with the current password, the system requires the homeowner to enter another password. 3. The new password length must be at least 8, and it needs to include at least one capital letter. If the password does not provide the requirements, the system asks the homeowner to enter another password.

USE CASE: ManageInternalSystemDevices

USE CASE NAME	ManageInternalSystemDevices
PARTICIPATING ACTORS	Initiated by The Homeowner Communicates with the sensors and cameras
FLOW OF EVENTS	1. The homeowner views all sensors status and can activate or deactivate sensors. 2. The homeowner views all cameras status and can monitor his/her house through cameras.
ENTRY CONDITION	1. The homeowner needs to have account for the system 2. The homeowner needs to log-in the system successfully
EXIT CONDITION	1. The homeowner saves changes that he/she made on sensors and cameras.
QUALITY REQUIREMENTS	1. Cameras and sensors need to work properly. 2. The homeowner can change sensors’ status once an hour.

USE CASE: DetectPossibleIncident

USE CASE NAME	DetectPossibleIncident
PARTICIPATING ACTORS	Initiated by the homeowner, burglar, or nature Communicates with the system
FLOW OF EVENTS	1. The system sends a message to **The Homeowner**. 2. If the homeowner opens windows-doors deliberately, the system asks homeowner to enter his/her password. 3. After the homeowner enters his/her password, the system does not create an incident and the system gives feedback to the homeowner. 4. If the homeowner did not open doors-windows, the homeowner can monitor his/her house by using monitoring his/her house feature from **'ManageInternalSystemDevices'** use case. The homeowner becomes sure about that there is a burglar that is trying to break into his/her house. 5. The homeowner starts to report a burglary incident. 6. The system asks the homeowner how many people are at the house. The homeowner enters how many people are at the house and the system creates the report. 7. The system reports the incident to the police station and informs by using '**GetInformation** use case. 8. The system reports the incident to the security and informs by using '**GetInformation** use case. 9. The system gives feedback to the homeowner that the incident reported by using '**GetInformation'** use case.
ENTRY CONDITION	1. The system detects burglar that is breaking into the house, or some windows/doors are open for so long.
EXIT CONDITION	1. The system gives feedback to the homeowner.
QUALITY REQUIREMENTS	1. The homeowner needs to inform the system less than 3 minutes. 2. The homeowner has configurated the system and the system operates well. 3. All sensors must be activated and work properly. 4. The system must obtain the phone number of the homeowner

USE CASE: LockAllDoors

USE CASE NAME	LockAllDoors
PARTICIPATING ACTORS	Initiated by the burglar Communicates with the sensors
FLOW OF EVENTS	1. The burglar enters the house. 2. The system detects that the burglar has entered the house and sends a notification to the homeowner that the burglar has entered. 3. The system locks the doors 10 seconds after the burglar enters the house. And the homeowner will be notified that the system locks all doors. 4. The owner checks whether there is someone in the house through the cameras. If there is no one at home, it opens the doors by entering the password through the system 5. After the homeowner his/her password and opens all the doors, the system sends a notification about the opening of the doors.
ENTRY CONDITION	1. The burglar enters the house by forcing the door or windows
EXIT CONDITION	1. The system sends notification to the homeowner.
QUALITY REQUIREMENTS	1. The system locks the doors inside the house 10 seconds after realizing that the burglar has entered the house.

USE CASE: GetNotification

USE CASE NAME	GetNotification
PARTICIPATING ACTORS	Initiated by The Homeowner
FLOW OF EVENTS	1. The system sends messages to police station and security if the homeowner approves there is a burglary incident and report the incident by using the system. The system gives feedback to the homeowner that the system has reported the incident to the police station. 2. The system gives feedback to the homeowner if the homeowner approves there is not a burglary incident and does not report the incident by using the system.
ENTRY CONDITION	1. The system needs to detect a possible incident.
EXIT CONDITION	1. The system gives feedback to the homeowner.
QUALITY REQUIREMENTS	None

USE CASE: DoesNotWorkApp

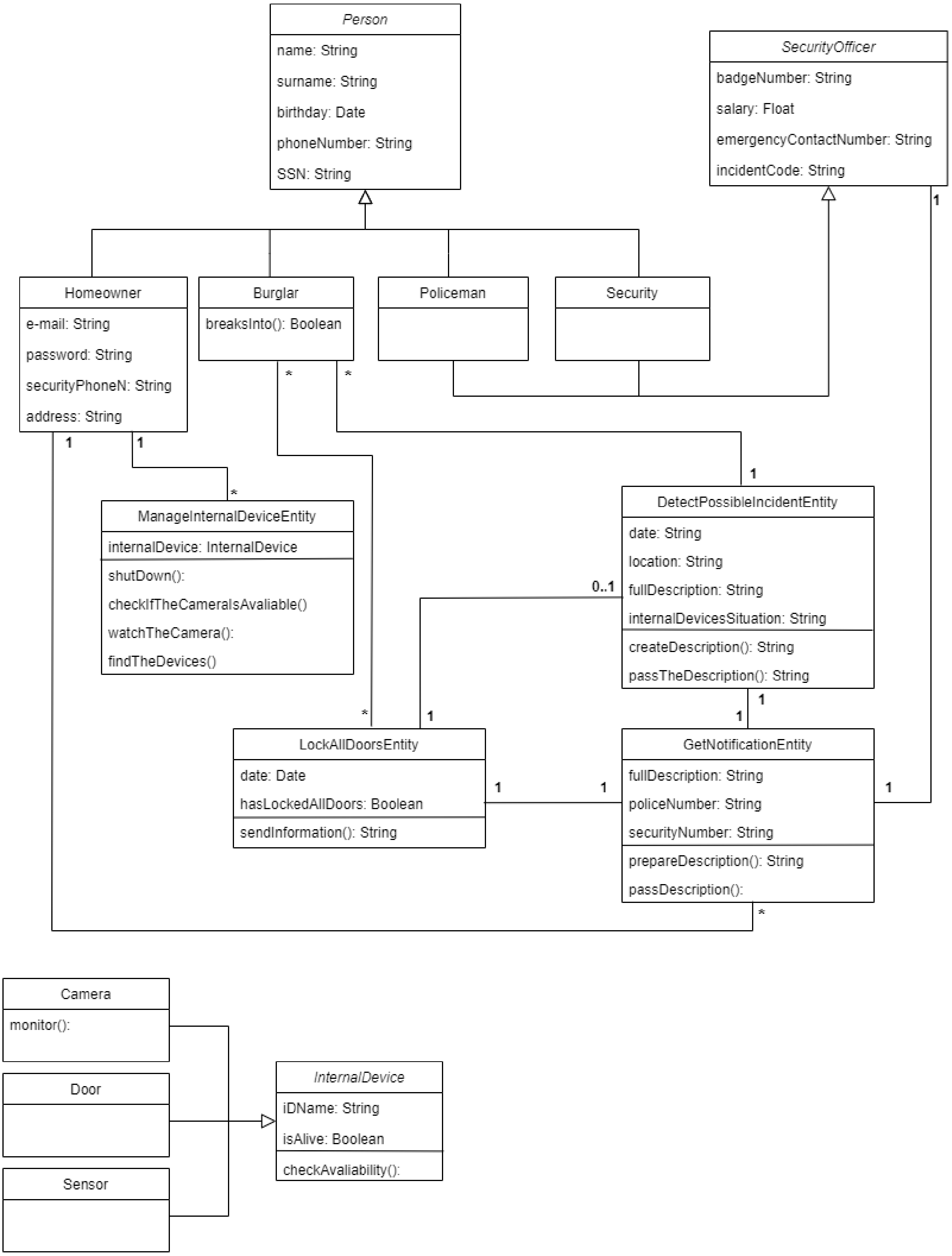
USE CASE NAME	DoesNotWorkApp
PARTICIPATING ACTORS	Initiated by the homeowner
FLOW OF EVENTS	1. When the homeowner is going to report a burglary incident, the system freezes, and the homeowner cannot report the incident. 2. Even if the homeowner tries to open and close the application, the homeowner will not get a result. 3. The system tries to contact the homeowner from the phone the homeowner gave when registering with the system, because the homeowner did not enter his/her password after a certain period of time or did not report a burglary incident. 4. The homeowner reports the situation verbally on the phone. The system figures out whether there is a burglary incident in accordance with what the landlord reported on the phone. 5. The system notifies the police if there is a burglary incident. If there is none, the system will not notify the police.
ENTRY CONDITION	1. The homeowner tries to report a burglary incident.
EXIT CONDITION	1. The system reaches the homeowner through his/her phone number. OR 2. The system cannot reach the homeowner through his/her phone and the system reports a burglary incident to the police station even though there is no burglary incident.
QUALITY REQUIREMENTS	None

USE CASE: FailToReachHomeowner

USE CASE NAME	FailToReachHomeowner
PARTICIPATING ACTORS	Initiated by the homeowner
FLOW OF EVENTS	1. The system sends a notification message to the homeowner, but the homeowner does not respond to the message sent by the system. 2. The system tries to contact the homeowner from the phone he gave when registering with the system, since the homeowner did not enter his password after a certain period of time or did not report a burglary incident. 3. The homeowner does not answer the phone for a long time. The system thinks as if a burglary incident has been reported because the system cannot reach the homeowner and informs the police station. 4. The system, if the homeowner has entered the number of the security when registering in the system, the system notifies the security.
ENTRY CONDITION	1. The system sends a notification to the homeowner
EXIT CONDITION	1. The system cannot reach to the homeowner and reports a burglary incident to the police station even though the system does not know if there is a burglay incident or not. (The homeowner has not notified the system by entering his/her password or answering his/her phone when the system calls.)
QUALITY REQUIREMENTS	1. The homeowner needs to inform the system less than 13 minutes. (3 minutes for entering his/her password and 10 minutes for answering the phone)

3.3.3 Object Model

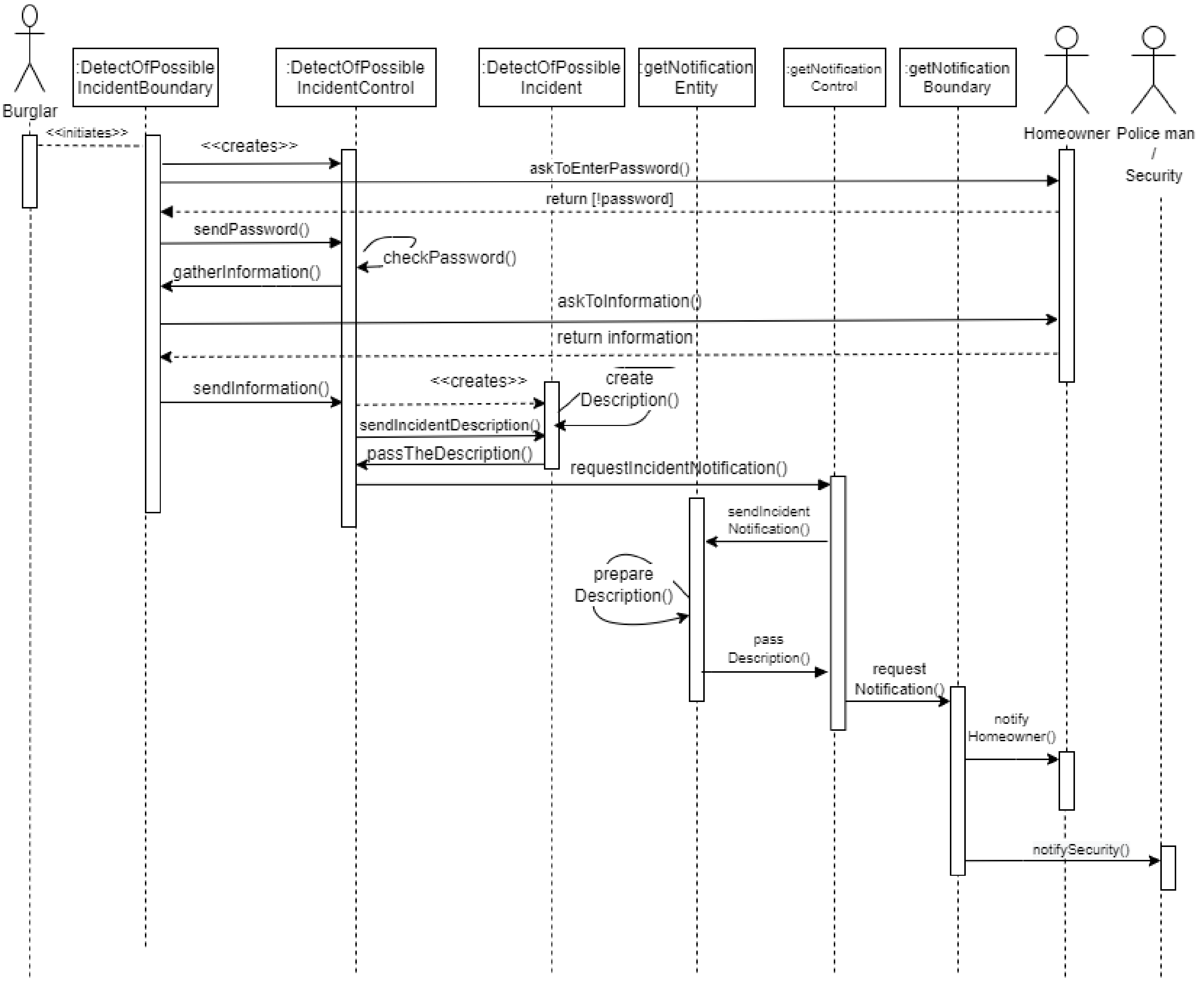
CLASS DIAGRAM



ManageInternalDeviceEntity, **LockAllDoorsEntity**, **DetectPossibleIncidentEntity** and **getNotificationEntity** has their boundary and control objects (**ManageInternalDeviceBoundary**, **ManageInternalDeviceControl**, **LockAllDoorsBoundary**, **LockAllDoorsControl**,**DetectPossibleIncidentBoundary**, **DetectPossibleIncidentControl**, **getNotificationBoundary**, **getNotificationControl**. They are shown in sequence diagrams). Generally, **homeowner** initiates boundary objects. These boundary object communicates with homeowner, and police – security, and control objects. **Burglar** can initiate some boundary objects by triggering. Boundary objects asks some input to homeowner, and they send the inputs to control objects. Control objects fill the gap between boundary and entity objects. Control objects evaluate the inputs and take specific actions. Control objects may can communicate with entity and boundary object after evaluating the inputs.

3.3.4 Dynamic Models

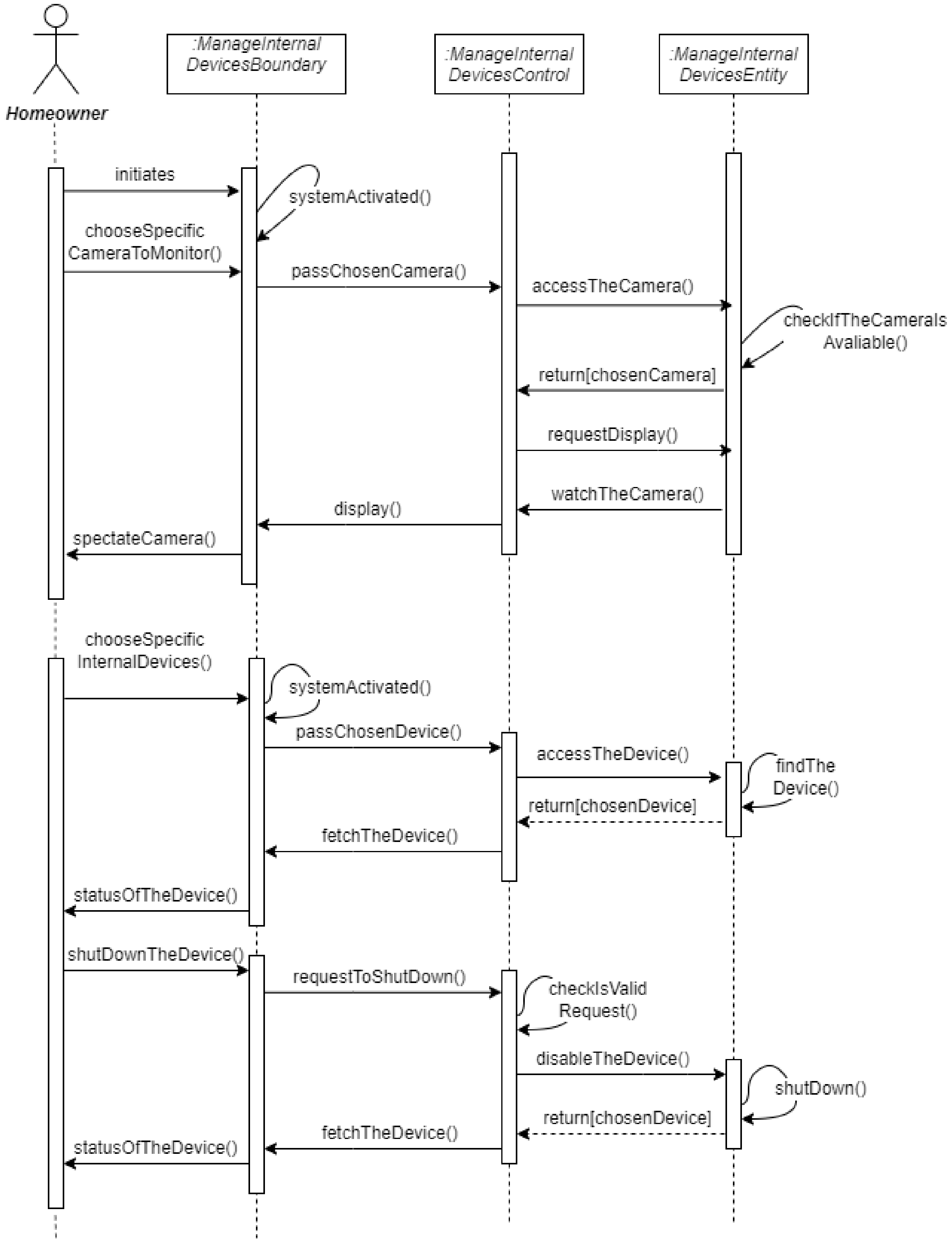
DETECT POSSIBLE INCIDENT SEQUENCE DIAGRAM



Burglar initiates **DetectPossibleBoundary** and. When the burglar initiates **DetectPossibleBoundary**, it creates an object from **DetectPossibleIncidentControl** to communicate **DetectPossibleIncident**. **DetectPossibleIncidentBoundary** asks homeowner to enter his/her password to figure out if there is an incident or not. (In this sequence diagram, homeowner does not enter his/her correct password or homeowner does not realize the triggered notification. Also, **homeowner** can monitor his/her house by using cameras, **ManageInternalDevices** use case. **Homeowner** can enter his/her password after monitoring his/her house. We did not add **ManageInternalDevices** in this sequence diagram because it is our critical use case) **DetectPossibleIncidentBoundary** sends the entered password to **DetectPossibleIncidentControl**. **DetectPossibleControl** checks entered password if it is valid or not. After checking entered password, it communicates with **DetectPossibleIncidentBoundary** to gather information from **homeowner**.

DetectPossibleIncidentBoundary communicates with Homeowner to gather information. Homeowner enters some information that he/she needs to give to the system, and **DetectPossibleDoundary** sends this information to **DetectPossibleIncidentControl**. **DetectPossibleIncidentControl** creates **DetectPossibleIncident** object and pass this information to **DetectPossibleIncident**. **DetectPossibleIncident** creates incident description, and it sends the incident description to **DetectPossibleIncidentControl**. **DetectPossibleIncidentControl** communicates with **getNotificationControl** to requests notification and pass the incident description. **getNotificationControl** object pass the description to the **getNotificationEntity**. **getNotificationEntity** prepares the notification and pass it to the **getNotificationControl**. **getNotification** control sends the notification to **getNotificationBoundary** and **getNotificationBoundary** notify homeowner and police-security.

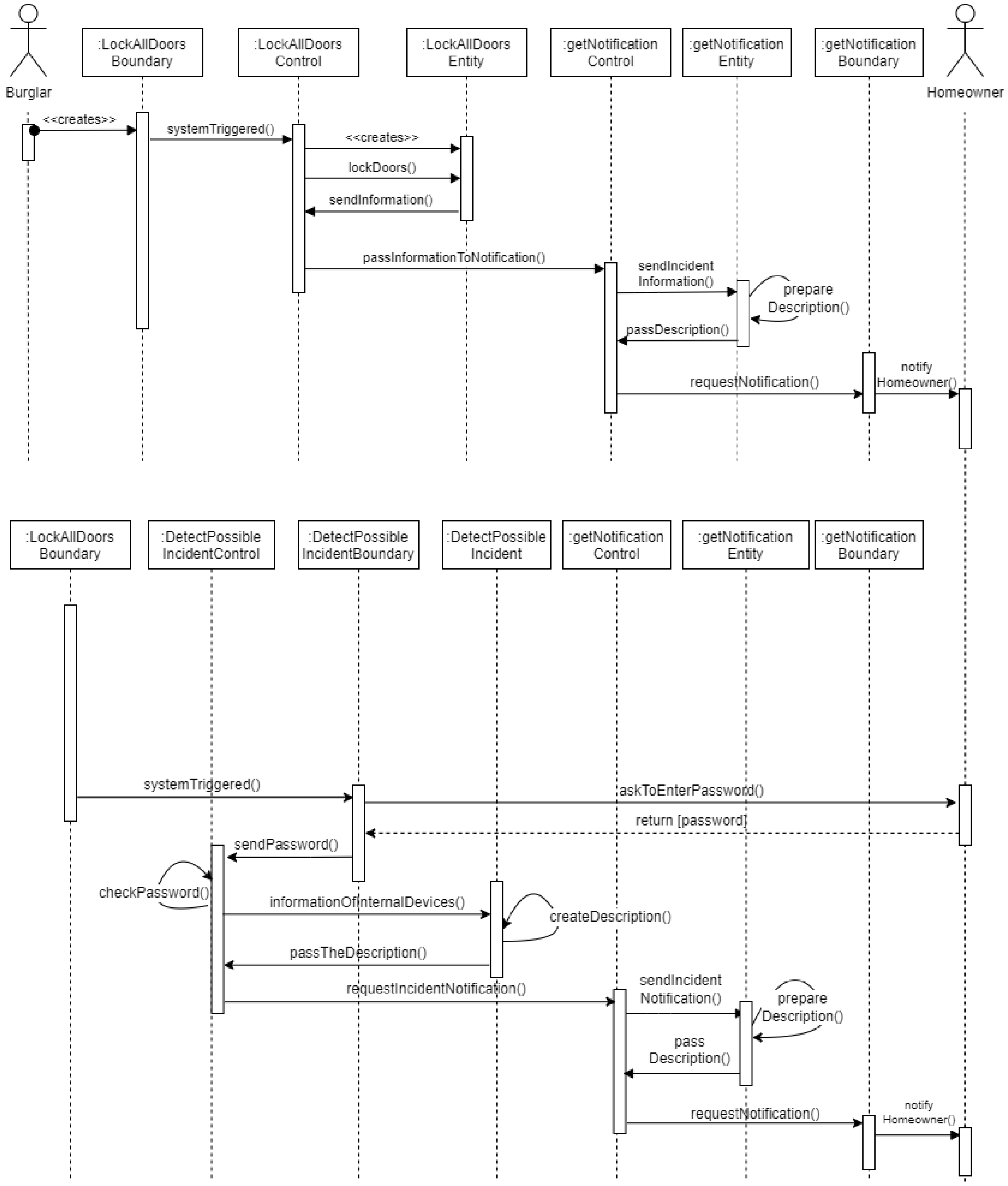
MANAGE INTERNAL DEVICES SEQUENCE DIAGRAM



Homeowner can monitor his/her house and check internal devices it they are working properly or not. **Homeowner** initiates **ManageInternalDevicesBoundary** and system activates. **Homeowner** chooses a camera that he/she wants to monitor. **ManageInternalDevices** sends the name of chosen camera to the **ManageInternalDevicesControl. ManageInternalDevicesControl** object accesses to **ManageInternalDevices** to fetch the chosen camera. **ManageInternalDevicesEntity** object checks if the chosen camera is available to monitor and returns chosen camera to the **ManageInternalDevicesControl**. **ManageInternalDevices** asks **manageInternalDevicesEntity** to monitor the chosen camera. **ManageInternalDevicesEntity** object fetches the display from chosen camera and sends the display to **ManageInternalDevicesControl**. **ManageInternalDevicesContol** sends the display information taken from the **ManageInternalDevicesEntity** to **ManageInternalDevicesBoundary** object. **ManageInternalDevicesBoundary** sends the display information to homeowner and homeowner monitors his/her house.

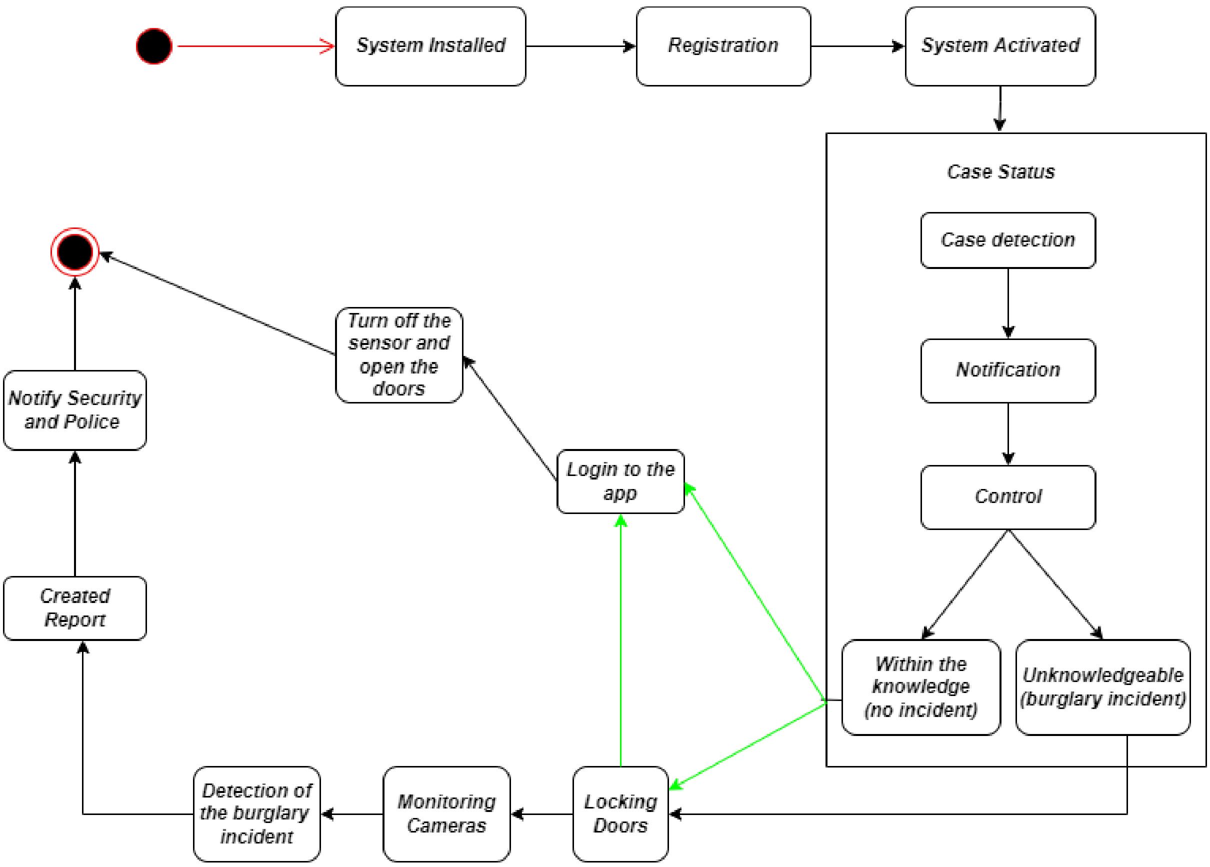
Homeowner can check internal devices (cameras, sensors, and doors) if they are working properly or not. **Homeowner** chooses an internal device. He/she initiates **ManageInternalDeviceBoundary** and system activates. **ManageInternalDevicesBoundary** sends the name of chosen internal device to **ManageInternalDevicesControl**. **ManageInternalDevicesControl** object accesses to **ManageInternalDevices** to fetch the chosen internal device. **ManageInternalDevicesEntity** finds the chosen internal device and returns it to the **ManageInternalDevicesControl**. **ManageInternalDevicesControl** object sends the chosen internal device to the **ManageInternalDevicesBoundary**. **ManageInternalDevicesBoundary** sends the chosen internal device information to the homeowner. Homeowner can check the chosen device status/condition and he/she can shut down the internal device. He/she sends an input to **ManageInternalDevicesBoundary** to shut down the chosen internal device. **ManageInternalDeviceBoundary** sends the shut down request to the **ManageInternalDevicesControl**. **ManageInternalDevicesControl** checks the request if it is valid or not, and then **ManageInternalDevicesControl** sends the request **ManageInternalDevicesEntity**. **ManageInternalDevicesEntity** find the chosen internal device and shut down. After shutting down the chosen internal device, it returns the chosen internal device and sends it to **ManageInternalDevicesControl** object. **ManageInternalDevicesControl** objects sends the information about the chosen internal device to **ManageInternalDevicesControl**. **ManageInternalDevicesControl** displays the status/condition of the chosen internal device and **homeowner** can check that whether his/her request is valid, or the system took an action for his/her request properly.

LOCK ALL DOORS SEQUENCE DIAGRAM



Burglar initiates **LockAllDoorsBoundary**. And the system locks the doors 10 seconds after detecting. This is the first point of operation of the system. Opens the **LockAllDoorsBoundary LockAllDoorsControl** class to control the system. In this way, the system is triggered. and system controls. From this point on, **LockAllDoorsControl** creates the **LockAllDoorsEntity** class to hold the information. and all information about the doors, such as the information that the doors are closed, are thrown in and returned. At this point the doors are locked. All that remains is to notify the homeowner. For this, **LockAllDoorsControl** links with the **getNotificationControl** class. Again, information is sent to the 'entity' object of **getNotification** and returned. then the **getNotificationBoundary** object is required to communicate with the homeowner. **getNotificationControl** connects the **getNotificationBoundary** class to contact the homeowner. finally, **getNotificationBoundary** sends the notification to the homeowner. in this way, the notification that the doors are locked is sent to the homeowner. it also triggers the **lockAllDoorsBoundary DetectPossibleIncidentBoundary** simultaneously after the doors are closed. and **DeteckPossibleIncidentBoundary** contacts the homeowner, asking the homeowner, who has watched for any threats in the home at the time, enter a password to open the doors. Seeing that there is no problem at home, the homeowner enters the password, and it is returned to the **DetectPossibleIncident** class. The password is sent from there to **DetectPossibleIncidentControl** and its accuracy is checked. Information-requiring operations are performed using **DetectPossibleIncidentEntity**. then the doors are opened and the notification that the doors are opened reaches the homeowner using the **getNotification** class.

GENERAL STATECHART DIAGRAM FOR SAFEHOMESYSTEM


System Installed
Registration
System Activated
Case Status
Case detection
Notification
Control
Within the knowledge (no incident)
Unknowledgeable (burglary incident)
Turn off the sensor and open the doors
Login to the app
Notify Security and Police
Created Report
Detection of the burglary incident
Monitoring Cameras
Locking Doors

First, the system is set up by the user. After the user completes the installation, he opens an account in the system and registers himself. After the registration process is completed, the user can activate the system and start using it.

The system can detect an adverse situation. After noticing this situation, it sends notifications to the necessary places and the user who receives this notification can control it. There are 2 different situations, user-informed and non-user-informed. If the user knows this situation, he/she closes the sensors by entering the password through the application and the locked doors are opened. In other cases, the user accesses the cameras through the application, detects how many people are inside, and reports the situation to the necessary places. The security or the police, who receive the notification, come to the scene immediately.

System Installed = The system is set up by the homeowner/user.

Registration = The user creates a record and enters the information to use the system.

System Activated = After the registration process is completed successfully, the system is activated by the user.

Case detection = Thanks to sensors, the system detects that something is wrong in the house.

Notification = After the detection process, it sends a notification to the user.

Control = The user controls it through the application.

Within the knowledge (no incident) = In cases where it is aware of, it turns off the sensors by entering a password into the application.

Unknowledgeable (burglary incident) = In situations that he is not aware of, he immediately enters the application for control, opens the cameras

Locking doors = This situation, which is a feature of the system, automatically locks all interiors in any situation to protect the house.

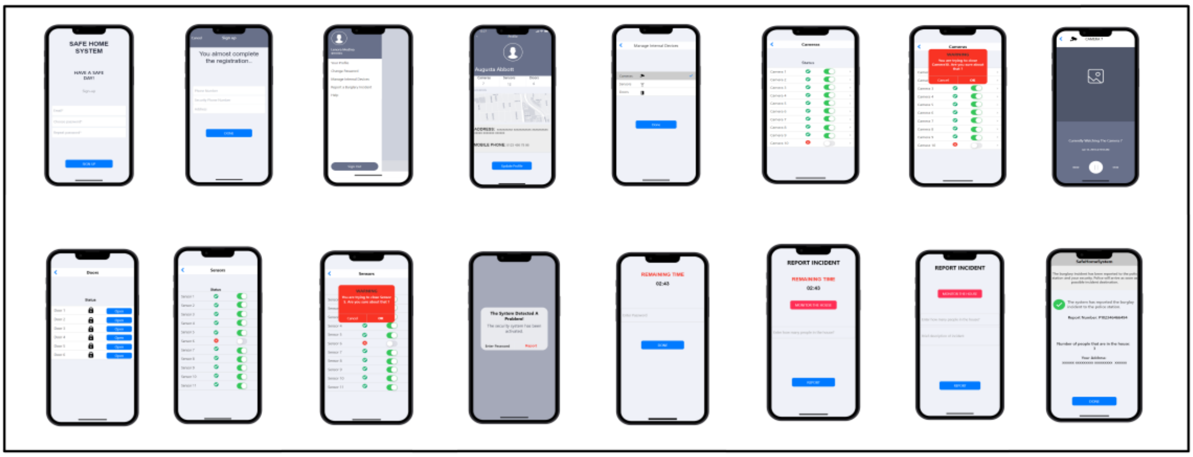
Monitoring Cameras = User can monitor the interior of the house through the application. You can get information about the situation.

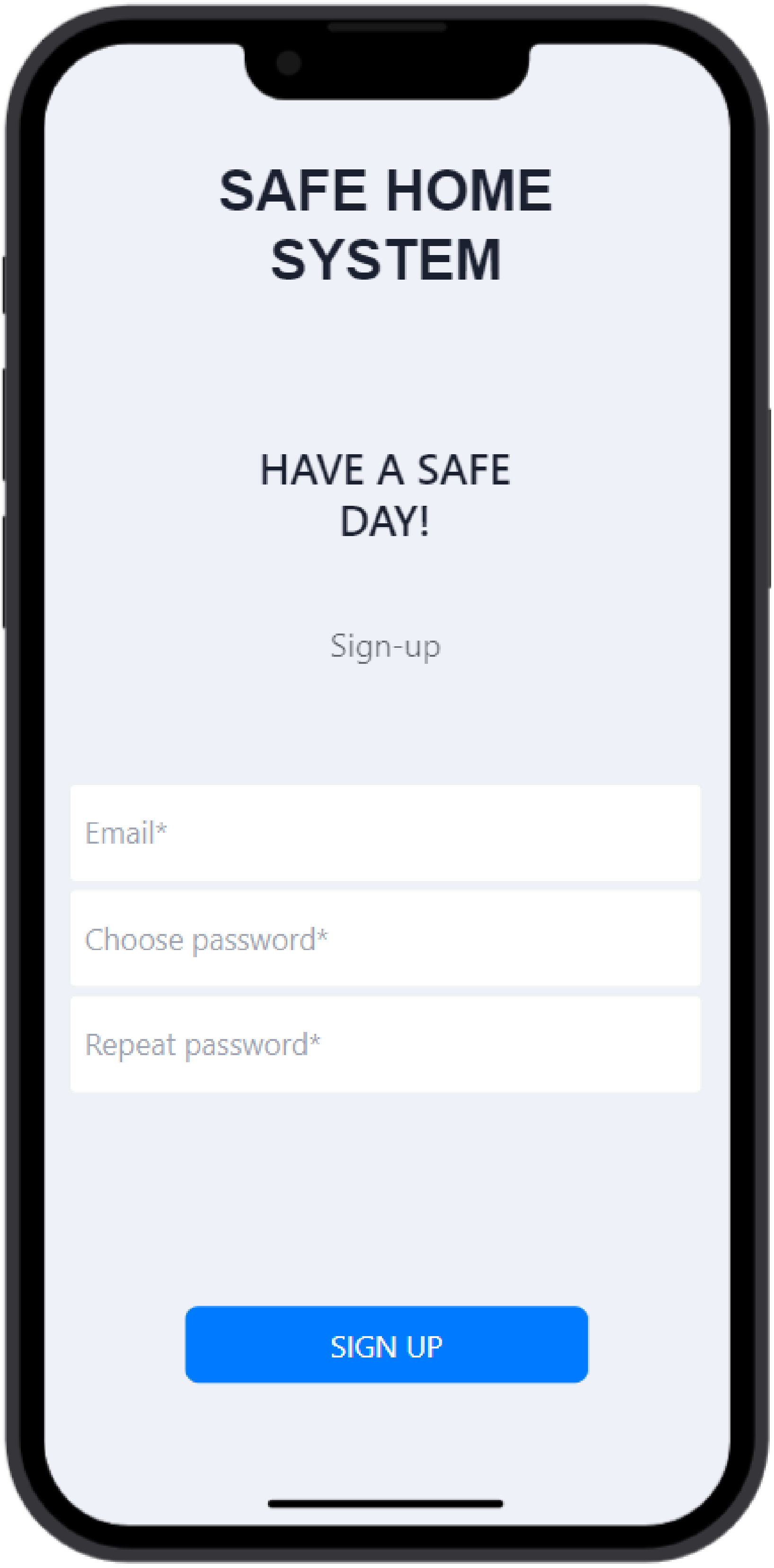
Detection of the burglary incident = After watching it from the camera, it detects the situation and learns the number of people.

Created Report = Creates an emergency notification and forwards it to the necessary places

Notify Security and Police = Upon the notification of the avil situation, they immediately go to the scene and catch the thieves

3.3.5 User Interface Mock-ups




SAFE HOME SYSTEM

HAVE A SAFE DAY!

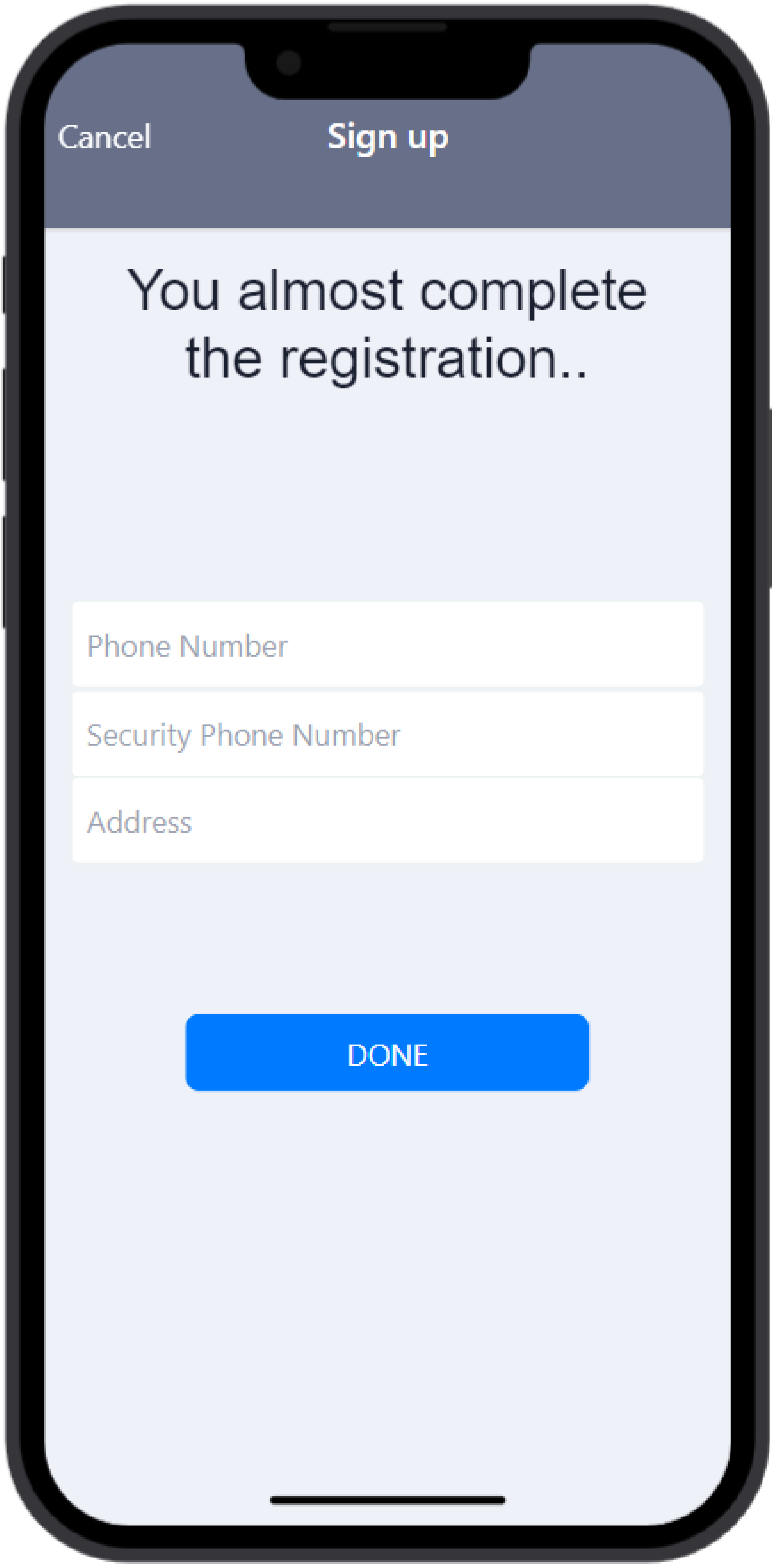
Sign-up

Email*

Choose password*

Repeat password*

Page 1 – Sign Up Page



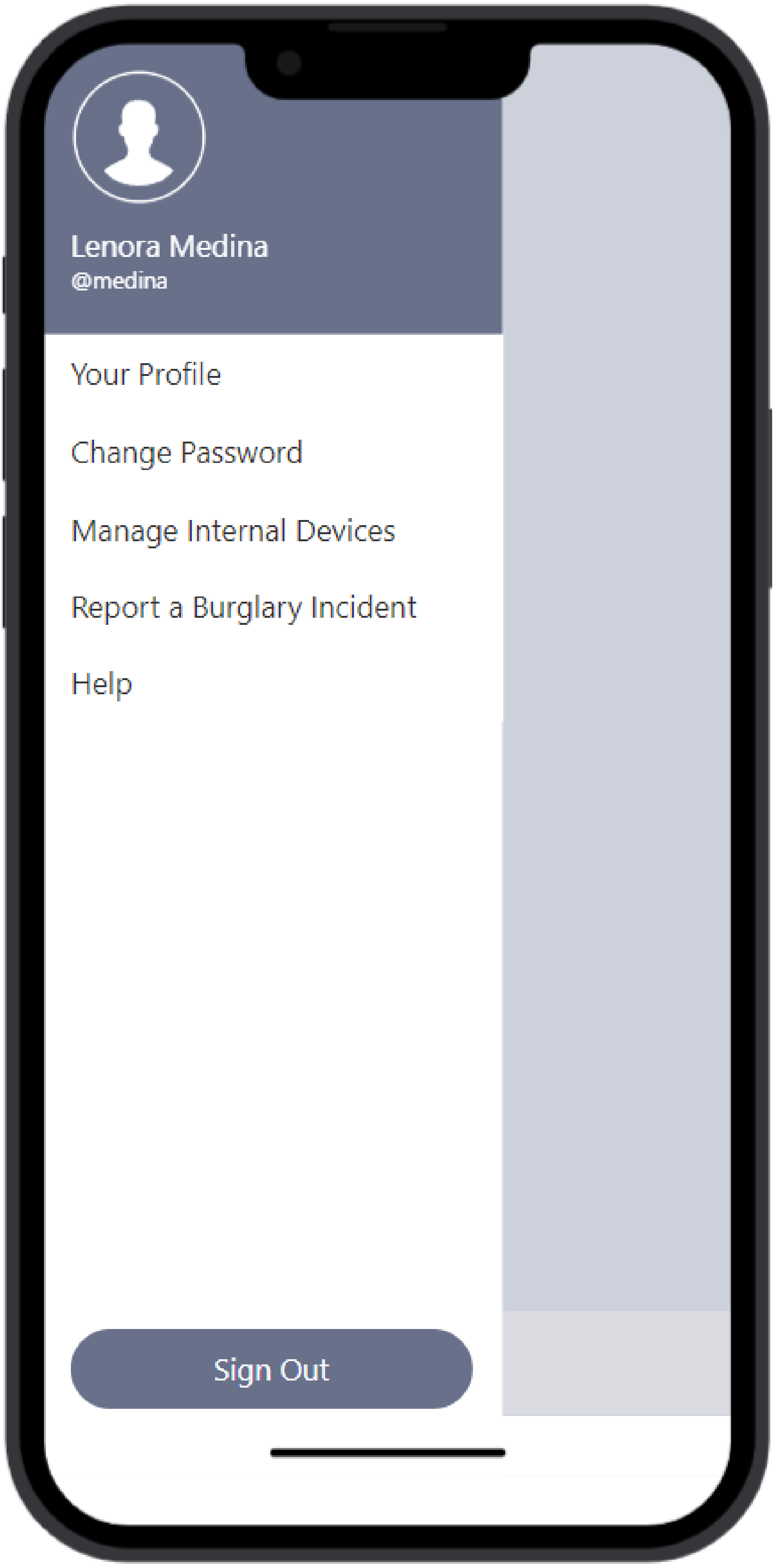
You almost complete the registration..

Phone Number

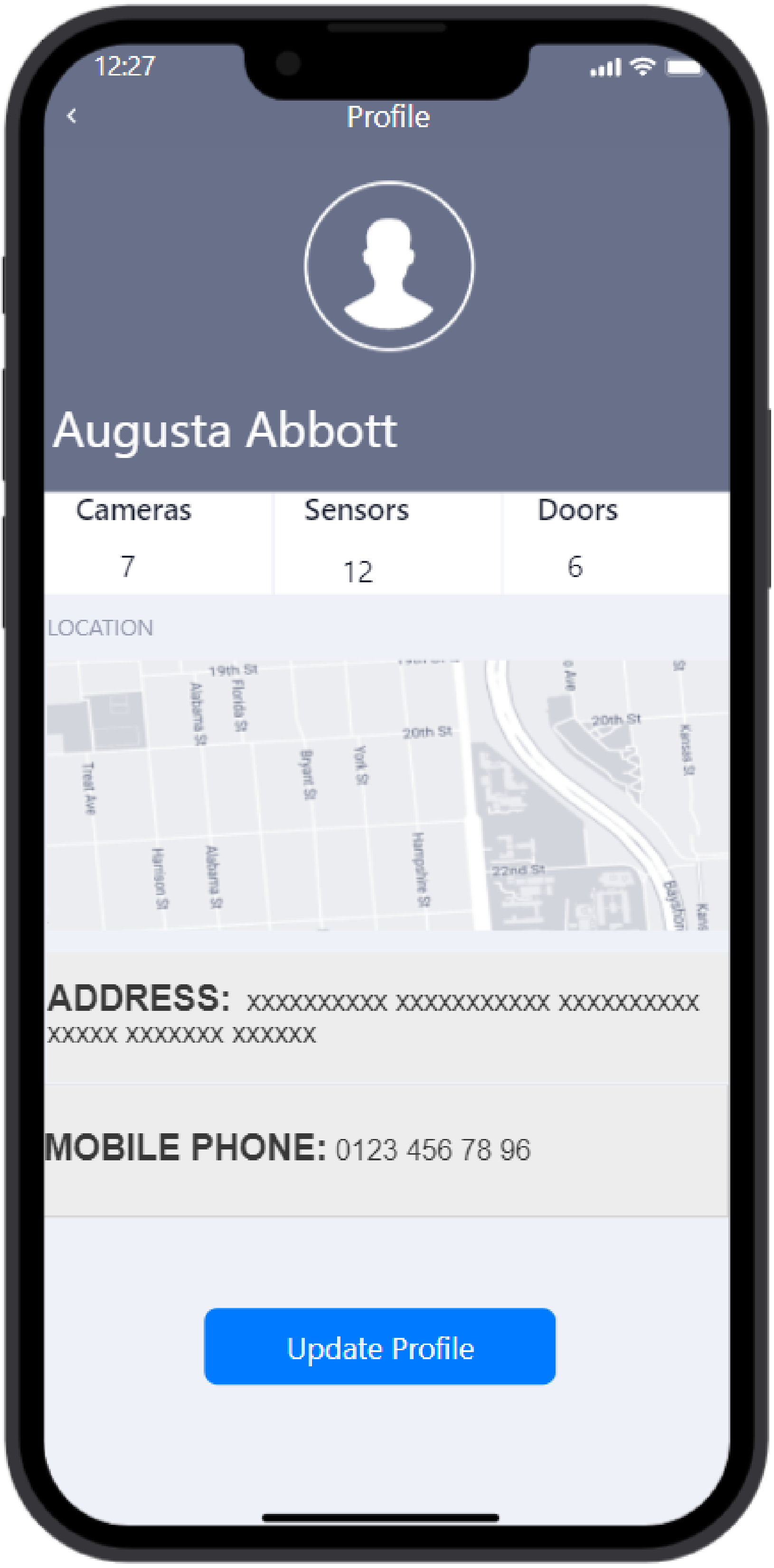
Security Phone Number

Address

Page 2 – Sign Up Page 2



Page 3 – Left Slide Menu



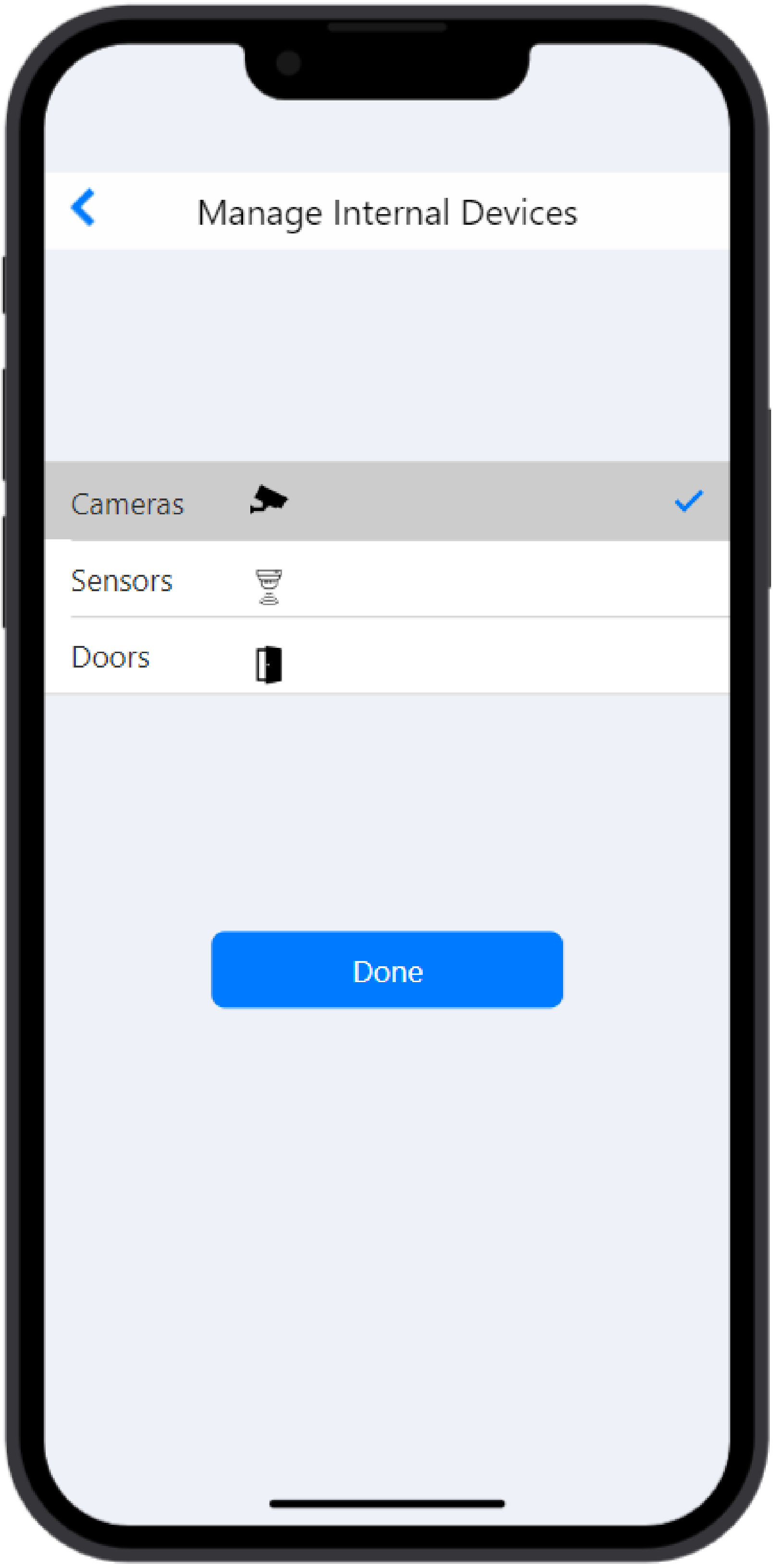
Augusta Abbott

Cameras	Sensors	Doors
7	12	6

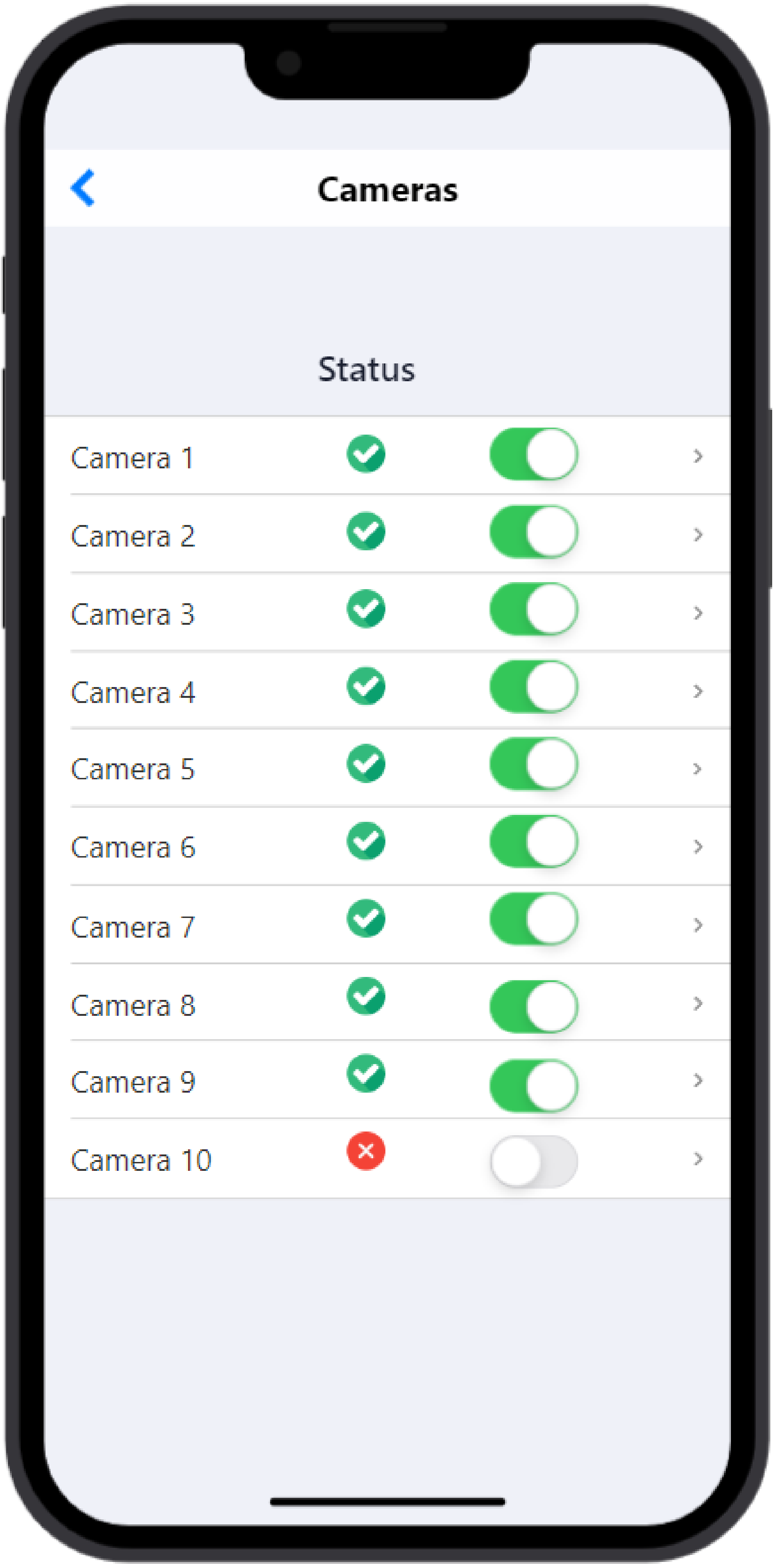
ADDRESS: xxxxxxxxxxx xxxxxxxxxxx xxxxxxxxxx xxxxx xxxxxxx xxxxxx

MOBILE PHONE: 0123 456 78 96

Page 4 – Profile Interface



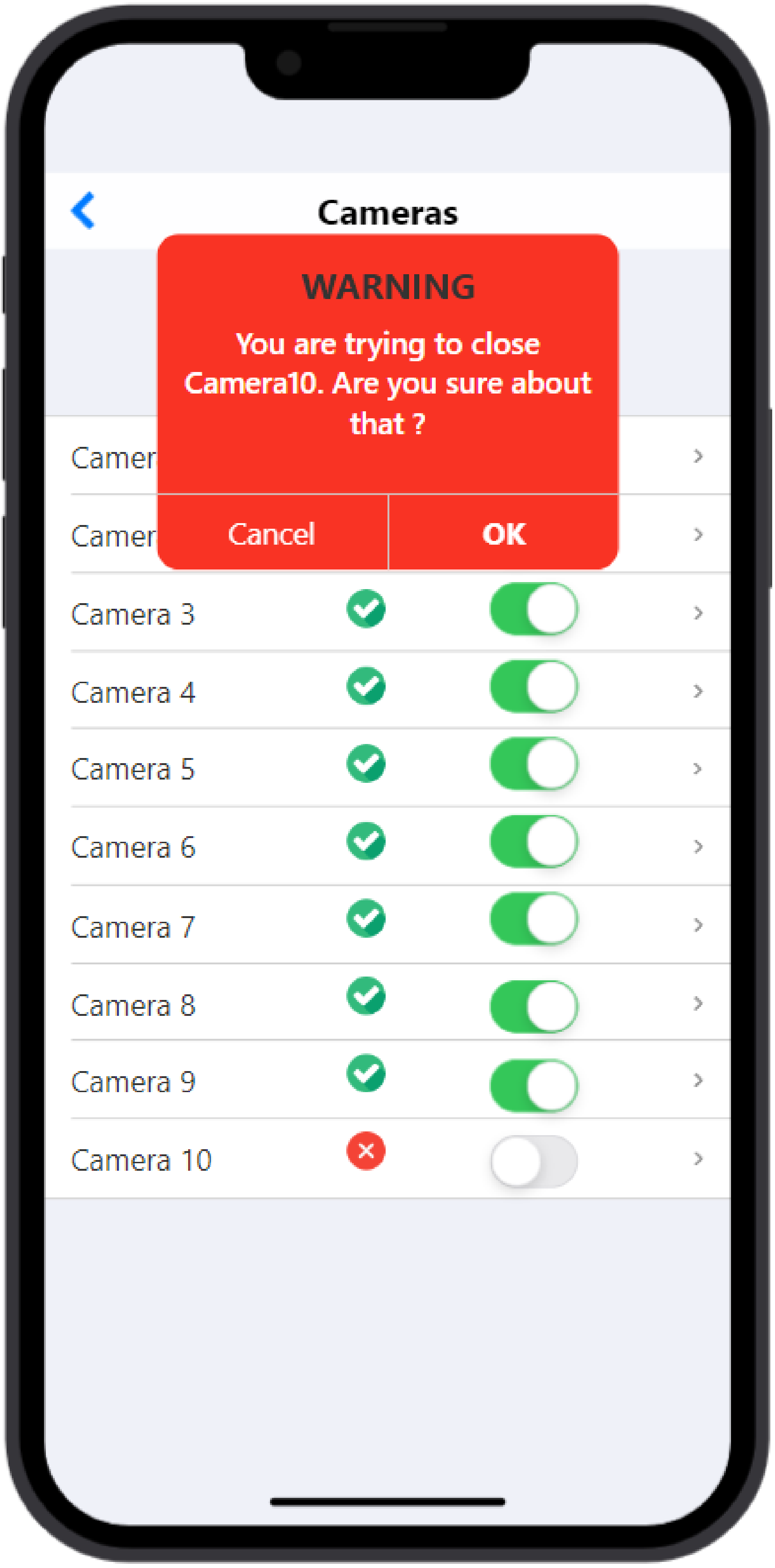
Page 5 – Manage Internal Devices Menu



Cameras

	Status		
Camera 1	✓	☑	›
Camera 2	✓	☑	›
Camera 3	✓	☑	›
Camera 4	✓	☑	›
Camera 5	✓	☑	›
Camera 6	✓	☑	›
Camera 7	✓	☑	›
Camera 8	✓	☑	›
Camera 9	✓	☑	›
Camera 10	✗	☐	›

Page 6 – Cameras Menu



Cameras

WARNING

You are trying to close Camera10. Are you sure about that ?

Cancel | OK

Camer ›

Camer ›

Camera 3 ✓ ● ›

Camera 4 ✓ ● ›

Camera 5 ✓ ● ›

Camera 6 ✓ ● ›

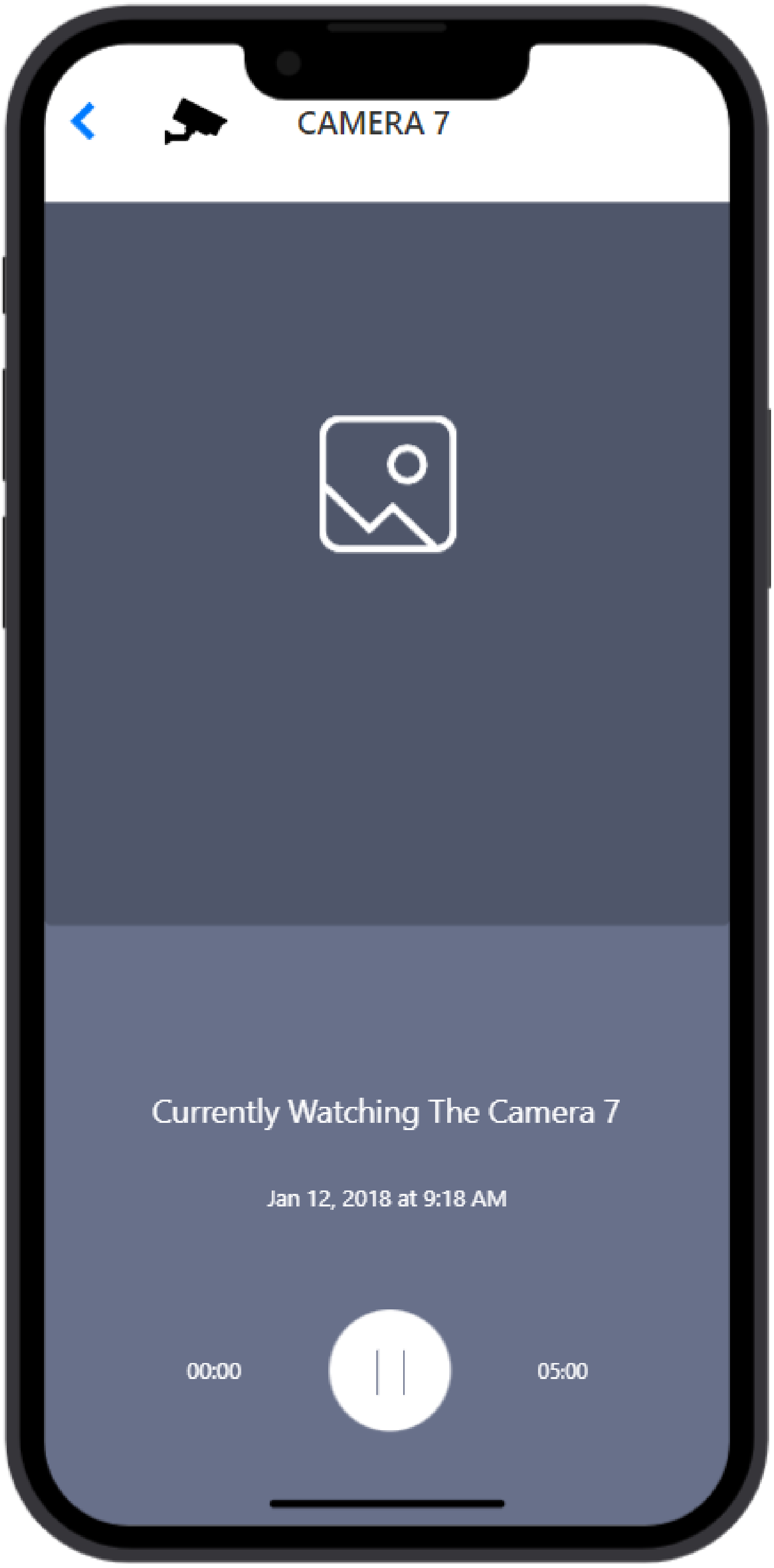
Camera 7 ✓ ● ›

Camera 8 ✓ ● ›

Camera 9 ✓ ● ›

Camera 10 ✗ ○ ›

Page 7 – Cameras Warning

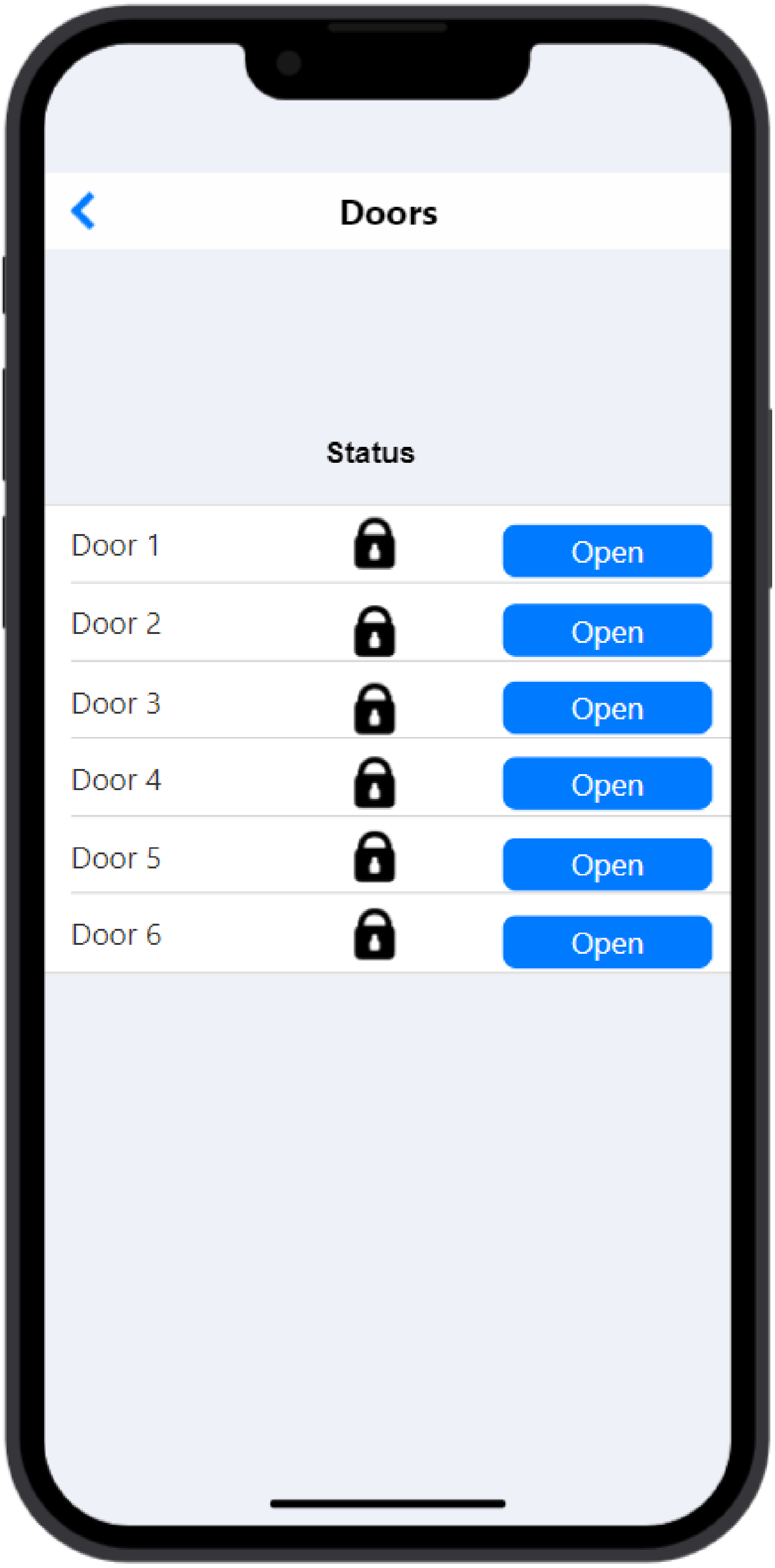


CAMERA 7

Currently Watching The Camera 7

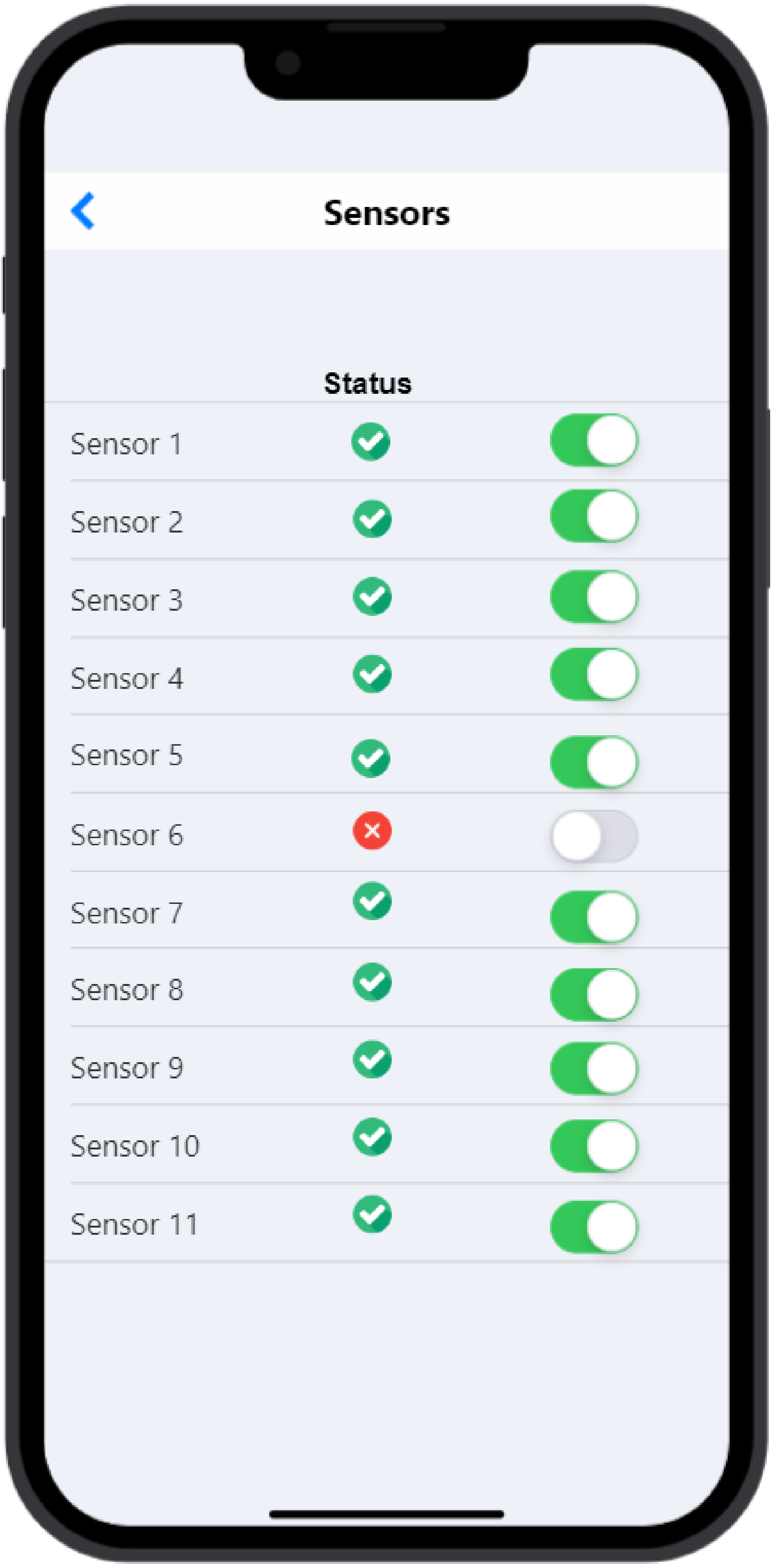
Jan 12, 2018 at 9:18 AM

Page 8 – Monitor Home Through Cameras



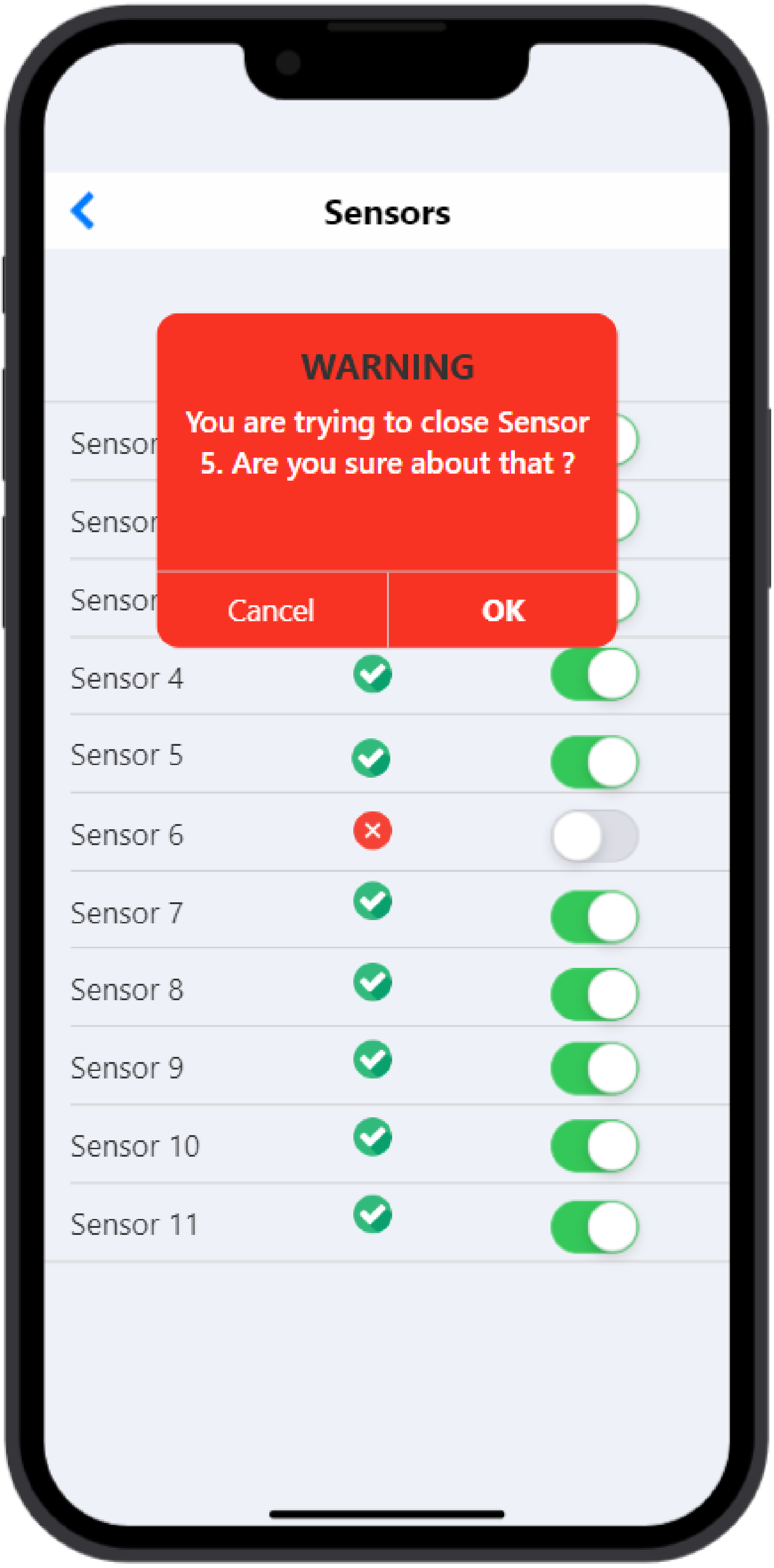
	Status	
Door 1	🔒	Open
Door 2	🔒	Open
Door 3	🔒	Open
Door 4	🔒	Open
Door 5	🔒	Open
Door 6	🔒	Open

Page 9 – Doors Menu Interface



	Status	
Sensor 1	✓	●
Sensor 2	✓	●
Sensor 3	✓	●
Sensor 4	✓	●
Sensor 5	✓	●
Sensor 6	✗	○
Sensor 7	✓	●
Sensor 8	✓	●
Sensor 9	✓	●
Sensor 10	✓	●
Sensor 11	✓	●

Page 10 – Sensors Menu Interface

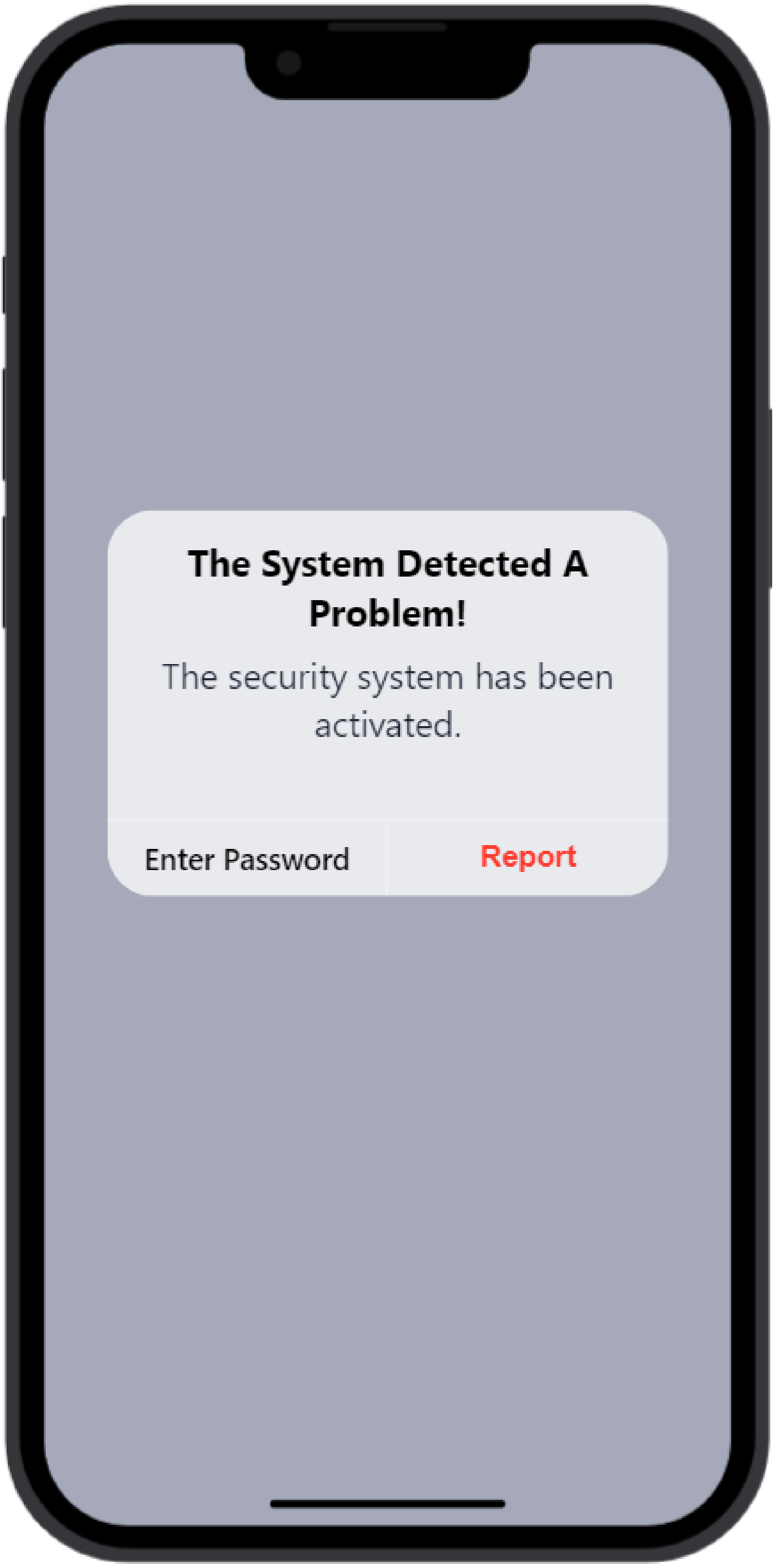


WARNING

You are trying to close Sensor 5. Are you sure about that ?

Sensor		
Sensor		
Sensor		
Sensor 4	✓	☑
Sensor 5	✓	☑
Sensor 6	✗	☐
Sensor 7	✓	☑
Sensor 8	✓	☑
Sensor 9	✓	☑
Sensor 10	✓	☑
Sensor 11	✓	☑

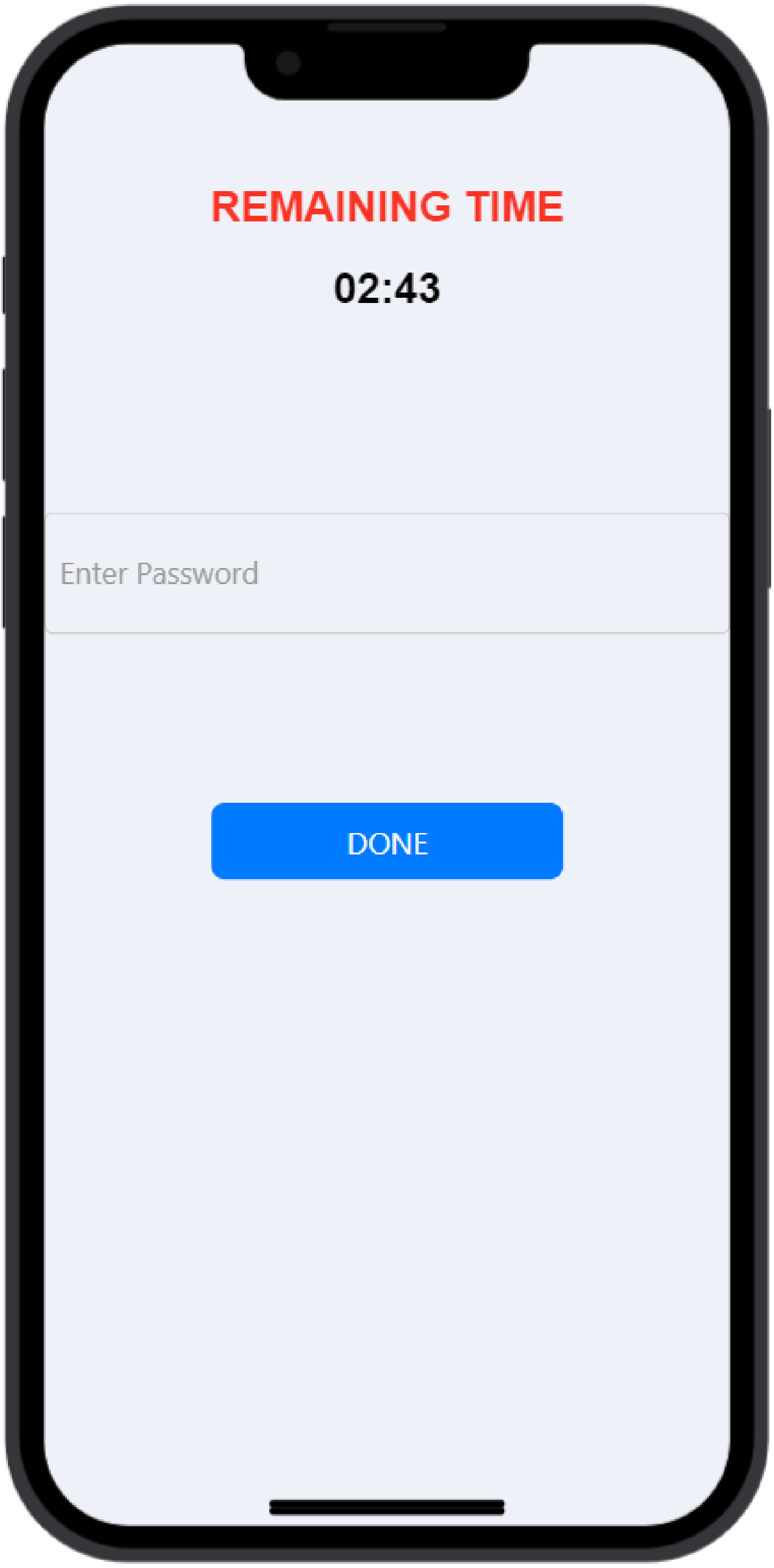
Page 11 – Sensors Menu Warning



The System Detected A Problem!

The security system has been activated.

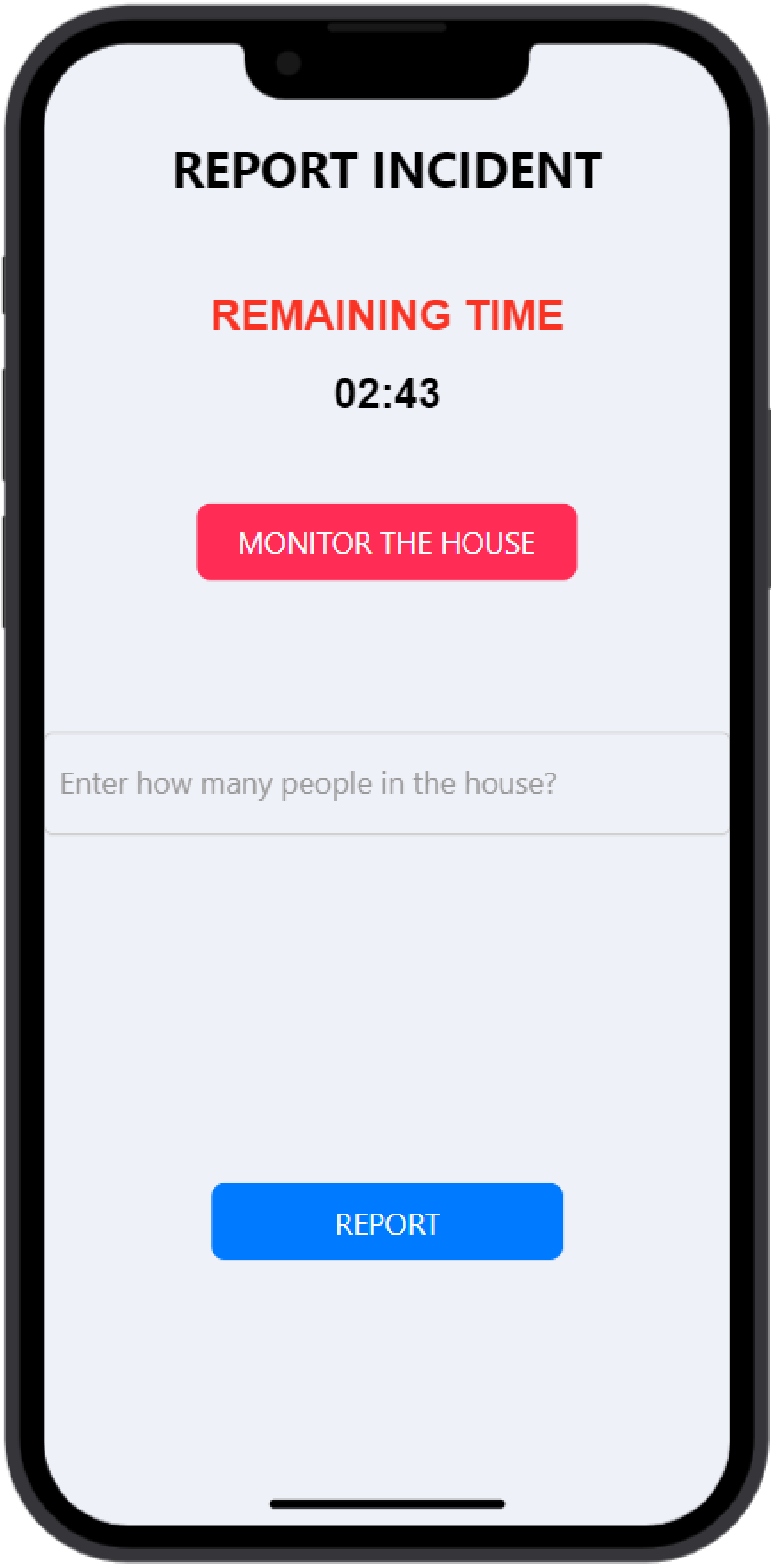
Page 12 – System Detect a Problem



REMAINING TIME

02:43

Page 13 – Entering Password Interface

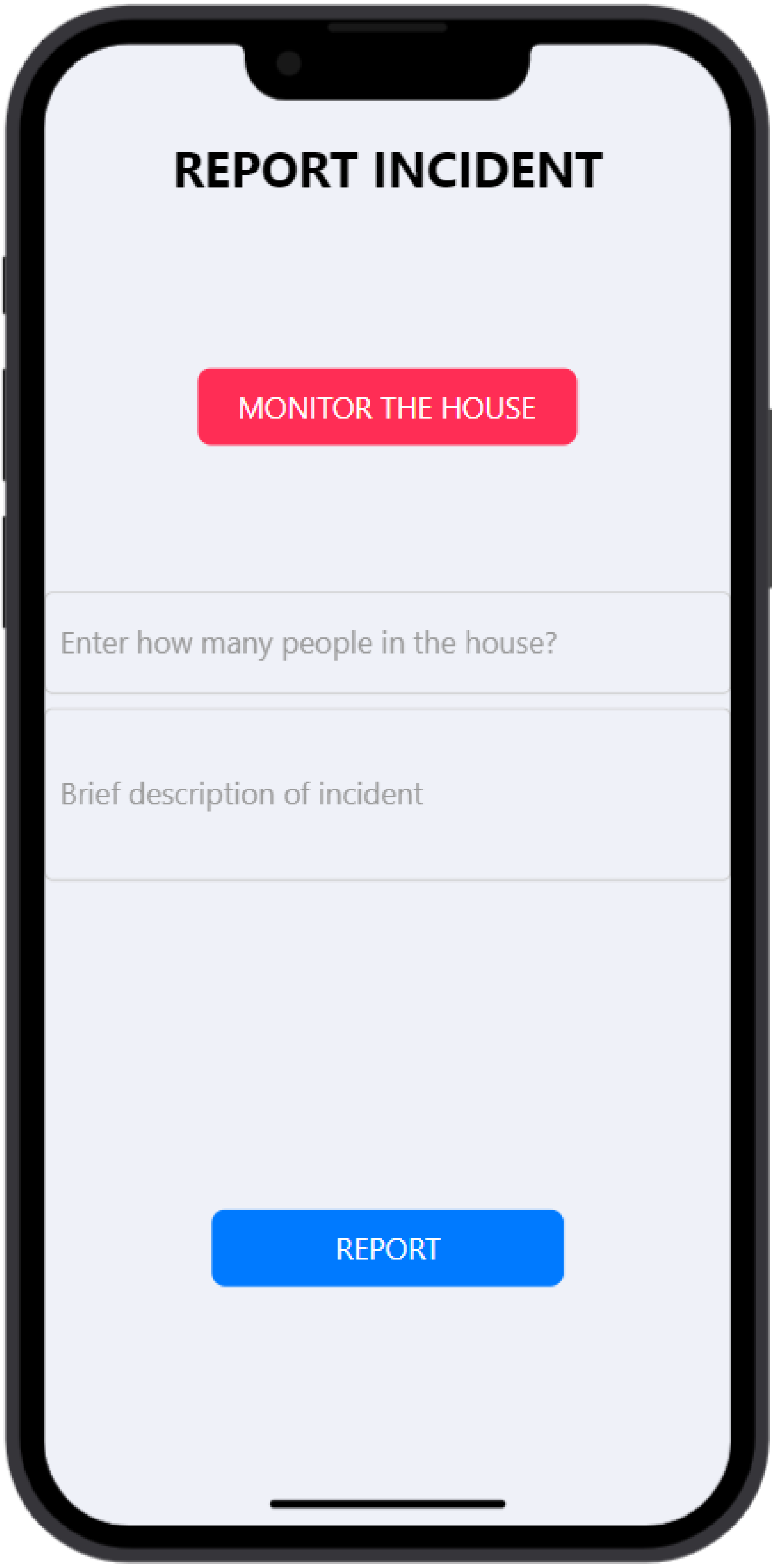


REPORT INCIDENT

REMAINING TIME

02:43

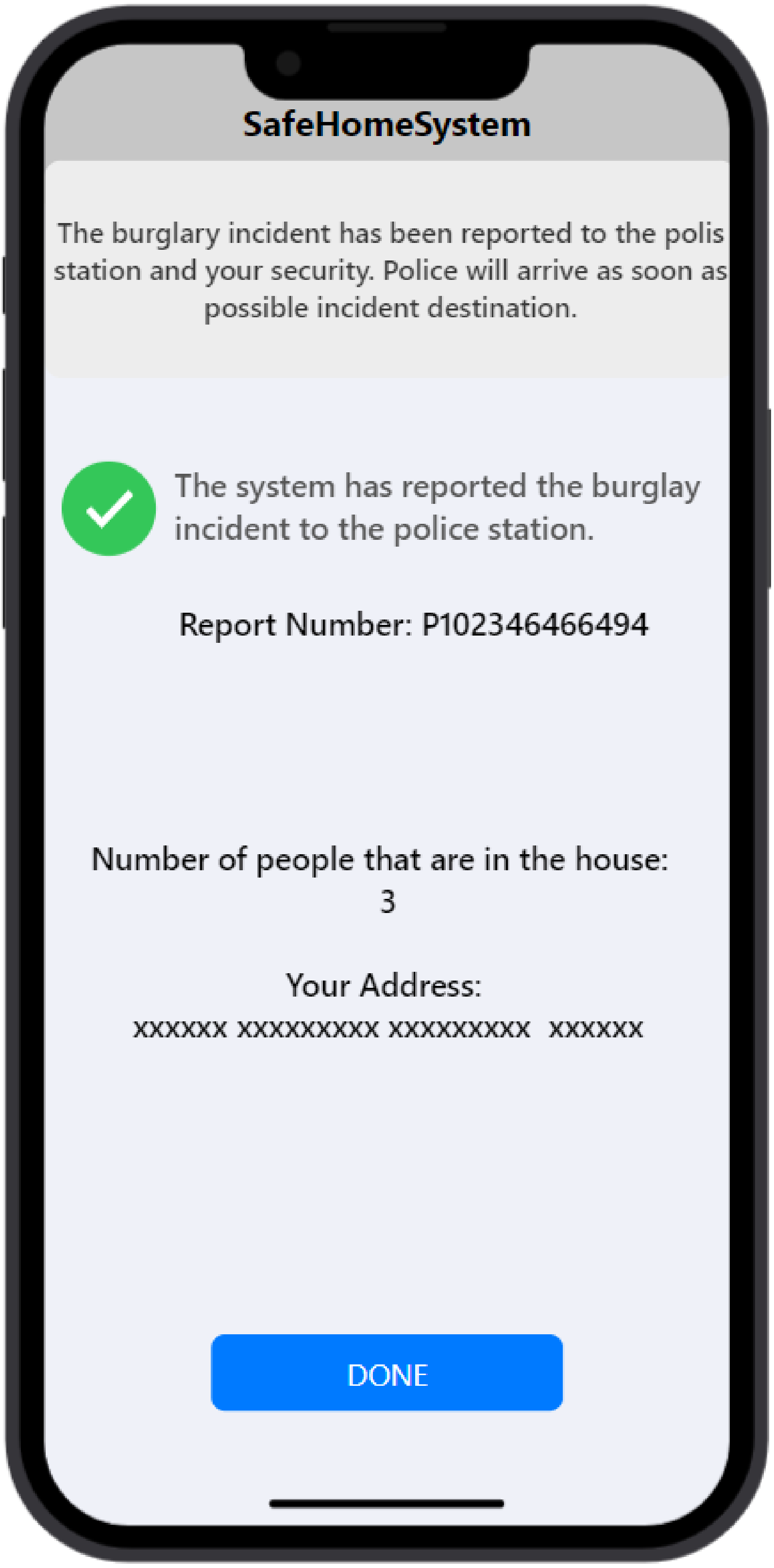
Page 14 – Report Incident Interface After Detection



REPORT INCIDENT

MONITOR THE HOUSE
Enter how many people in the house?
Brief description of incident
REPORT

Page 15 – Report Incident Interface from Left Menu



SafeHomeSystem

The burglary incident has been reported to the polis station and your security. Police will arrive as soon as possible incident destination.

✓ The system has reported the burglay incident to the police station.

Report Number: P102346466494

Number of people that are in the house: 3

Your Address: xxxxxx xxxxxxxxx xxxxxxxxx xxxxxx

Page 16 – Message Sent from System After Reporting A Incident

4 Glossary

- **EXTERNAL FACTORS IN THE SYSTEM**

Homeowner = Person or people living in the house where the system is located.

Nature = It's an external factor that can activate the system such as natural disasters, wind, storm.

Burglar = The person trying to break into the house to steal something.

Policeman = Person responsible for catching the burglar and guarding the house.

Security = Responsible for the site security.

- **SYSTEM FEATURES**

CreateAccount = To use the system through the application, it is necessary to register and open an account.

ChangePassword = Renewing the password in case of forgetting or expiration of the password.

ManageInternalServiceDevice = The user's use of the system through the application.

DetectPossibleIncident = Be active in a possible case.

DoesNotWorkApp = App crashes and doesn't work.

FailToReachTheHomeOwner = The situation where the host does not return the notification on the phone and cannot reached.

LockAllDoors = The system's protection of the house in case of danger.

getNotification = Controls and evaluates incoming notifications.

- **GENERAL**

Verification = Confirmation message sent by the system for security.

userID = The login information given to the user by the system.

Validate = Confirmation message sent by the system for security.

Sensors = Devices that send notifications to send system.

5 Appendix

- Annex – I: Distribution of Work
- Annex – II: Meeting Minutes

Distribution of Work

Yaren and Uğurcan created ManageInternalDevices sequence diagram

Hurşit and Sefa created LockAllDoors sequence diagram

Merthan created DetecPossibleIncident sequence diagram.

We did create and edit class and statechart diagram together.

Meeting Minutes

<Copy this sheet as much as you need (for each meeting)>

Date:	18/10/2022
Location:	CBU engineering faculty canteen
Duration:	1 hour
Participants:	Yaren Mamuk, Uğurcan Çırak, Merthan Erler, Hurşit İçke, Sefa Altınok, Elif Merve Arslan

Understand and interpret the given project, do research on the given project, write a scenario for the project

As group members, we gathered in the canteen to do the project homework. We opened the given project assignment and started talking about the assignment. We decided that research should be done for the project in the WhatsApp group we had previously established. We started the meeting by telling everyone about their ideas. The ideas found were saved in word file. Then we jointly identified an emergency: burglary. We started to think about scenarios related to the emergency we found. Ideas came from everyone. Finally, we decided on 3 scenarios. We wrote the details of the scenario requested in the assignment. The works were shared in the WhatsApp group. We decided to hold a meeting to talk about the missing places and ended the meeting.

Meeting Minutes

<Copy this sheet as much as you need (for each meeting)>

Date:	20/10/2022
Location:	CBU engineering faculty canteen
Duration:	1 hour
Participants:	Yaren Mamuk, Merthan Erler, Hurşit İçke, Sefa Altınok

Completing the missing parts in the project

We talked about the missing parts of the project. We arranged the deficiencies in the scenario and the places where there was meaninglessness. We wrote the use cases in detail. After writing the use cases, we drew the UML diagram.

Meeting Minutes

<Copy this sheet as much as you need (for each meeting)>

Date:	23/10/2022
Location:	meeting on Discord
Duration:	1 hour
Participants:	Yaren Mamuk, Merthan Erler, Uğurcan Çırak, Sefa Altınok

Completing the missing parts in the project

We've added a critical Use Case. We have completed the UML diagram. Talked about functional and non-functional requirements. Finally, we did the missing parts of the homework.

Meeting Minutes

<Copy this sheet as much as you need (for each meeting)>

Date:	24/10/2022
Location:	CBU engineering faculty canteen
Duration:	1 hour
Participants:	Merthan Erler, Uğurcan Çırak, Sefa Altınok

Completing the missing parts in the project

We talked to Zeynep teacher about homework. We learned the missing and wrong places in the homework. Then we did the homework arrangement. Functional and nonfuncitonal req. We have completed functional and nonfunctional req.

Meeting Minutes

<Copy this sheet as much as you need (for each meeting)>

Date:	28/10/2022
Location:	meeting on Discord
Duration:	1 hour
Participants:	Uğurcan Çırak, Yaren Mamuk

Completing the missing parts in the project

We have completed the last missing part of Glossary in homework. Then we completed the meeting reports.

Annex – II

Meeting Minutes

<Copy this sheet as much as you need (for each meeting)>

Date:	04/11/2022
Location:	CBU engineering faculty canteen
Duration:	2 hours
Participants:	Yaren Mamuk, Uğurcan Çırak, Merthan Erler, Hurşit İçke, Sefa Altınok, Elif Merve Arslan
Understand and interpret the given project, do research on the given project, write a scenario for the project	
We met to complete step 3. We have completed the User Interface Mock-ups section. We have designed a simple mobile application in a way that everyone can understand. After completing the User Interface Mock-ups, we finished our homework.	

Meeting Minutes

<Copy this sheet as much as you need (for each meeting)>

Date:	26/11/2022
Location:	Meeting on Discord
Duration:	1 hour
Participants:	Yaren Mamuk, Merthan Erler, Hurşit İçke, Sefa Altınok, Uğurcan Çırak, Elif Merve Arslan

A meeting was held for the parts that need to be done in the project.

We distributed the tasks for the departments requested to be done in the project. We talked about how to do the necessary parts in the homework and shared information.

Meeting Minutes

<Copy this sheet as much as you need (for each meeting)>

Date:	04/12/2022
Location:	meeting on Discord
Duration:	3 hours
Participants:	Yaren Mamuk, Merthan Erler, Hurşit İçke, Sefa Altınok, Uğurcan Çırak, Elif Merve Arslan

Completing the missing parts in the project

We have all completed the given sections. We edited the parts that we saw missing or wrong. We told each other things we didn't understand in homework. We have completed the assignment