

**Baby Monitor**

**Final Report**

Course: ECCE-336-Intro to Software Engineering

Semester: Fall 2022

Section No.: 1

Students Names: Khaled Alhefeiti, Ahmad Mohammad, Saeed Almarri, Falah Alfalahi

Student Ids: 100058849, 100053567, 100053463, 100059123

Instructor: Dr. Davor Svetinovic

Teaching Assistant: Ruba Nasser

Link: <https://github.com/ECCE336-100053463/ECCE-336-Project>

1. Introduction 4

1.1 Overview 5

1.2 Scope 5

1.3 Intended Audience 5

1.4 Intended Use 6

2. Feasibility Study 7

2.2 Costs and Benefits 8

2.3 Technical 8

2.4 Resources 9

2.5 Alternatives 9

3. Requirements 10

3.1 System Requirements 11

3.2 Functional Requirements 13

3.3 External Interface Requirements 15

3.4 Non-functional Requirements (NFRs) and Constraints 16

4. Modeling and Designing the Solution 19

4.1 Use Case 20

4.2. Class Diagram 27

4.3. Sequential Diagrams 31

4.4. State Diagrams 33

4.5. Data Dictionary 39

4.6. Tools 39

5. Implementation 40

5.1 Start Page 41

5.2 Login Page 42

5.3 Registration Page 43

5.4 Main Menu 44

5.5 Camera Feed 45

5.6 Recorded Feed 46

5.7 Live Feed 47

5.8 Notification Page 48

1. **INTRODUCTION**

**1.1 Overview**

The product is a Baby monitoring application with a live camera that shows live footage of the child’s room 24/7. Moreover, the system will send notifications of the wellbeing of the child and send alerts in the case of emergencies. Furthermore, the application can even contact the local authority in the case that the parents do not reply fast. Multiple devices will be able to use this application at once.

**1.2 Scope**

The system’s goal is to set up an application in which the parents Mouza and Ahmed are able to view their child Mohammad wherever they may be. We will first create a streaming live feed function in which the parents are able to link their smart devices to our application and proceed to view their household whenever they want.

Secondly, we will make sure to have a functionality in which multiple devices are able to view the same live feed at once so that anyone can access the application at the same time so that we don’t have a waiting time for any user. Finally, we will have a function that picks up voices from the surrounding cameras in which it notifies the parents phone if something happened by giving them a notification to their devices based on sounds such as baby crying.

**1.2** **Intended Audience**

This application is targeted directly to working parents that have to leave their child at home with a caretaker all day long until they get back home from work. It can be very helpful and ease the parents' worrying significantly and not have to call back home multiple times a day.

**1.3** **Intended Use**

The baby monitor is helpful in multiple ways. For instance, it can access a live feed of the baby’s room anywhere anytime. Furthermore, it can also access recorded footage up to a week prior. Most importantly, it sends alert in the case of any emergency that occurs in the child’s room

**2. FEASIBILITY STUDY**

**2.1 Costs and Benefits**

The development of the software will cost at least 15000 AED. The customer will have many benefits using the system.

For example:

* + Providing a live feed access.
  + Notifications about surroundings.
  + Private video feed for security.
  + Multiple access to feed.

**2.2 Technical**

To complete this project, we will require a programmable camera, a personal computer for each member to handle all the coding and prototyping. That is mostly all that will be required hardware wise. On the other end of the spectrum some special apps, websites and tools will be crucial to managing everything. To start we will use Github as our project manager. All our planning will be conducted on it. Visual Studio Code is the application where we will do our coding and prototyping. What sets it apart is its extensive library of extensions and tools of which we will be using many. The most important one is PlantUML, a tool to simply and easily create advanced diagrams using simple lines of code. Lastly, we will have to rent a virtual private server to have our camera and app connected to the web all of the time. With all these tools available at hand what we have in scope for the project would be achievable.

**2.3 Resources**

We are expecting to complete the movie system project within a period of six weeks. It will be done by a group of 1-3 members, and each member will have an experience in application design implementation and programming since it is going to need an application to run.

The programming language that we want to use is python as it is the language that most of the members are experienced at.

Furthermore, the project should represent all the requirements that the client needs, and it must be done as requested.

**2.4 Alternatives**

The issue of remotely monitoring babies can have a long list of solutions. Not all of them must implement a surveillance system. It can be solved with a fraction of a cost using a simple sound-based device that sends a warning when the sound is higher than 100 decibels which means that the baby is crying so the parents can check on him. Another solution could be making a smart bracelet that could check and monitor all aspects of a child’s wellbeing. But that could prove costly and timely. So changing the approach of solving this problem ever so slightly can have the biggest influence on the entire cost of the project.

**3. REQUIREMENTS**

**3.1 System Requirements:**

The system must be accessible by downloading the app through Google Play store for Android devices, or Apple’s App Store for iPhone devices.

Upon opening the app, the user should be requested to either log-in or register for an account, with inputs for an e-mail and password to enter to their account, as well as a phone number for emergency notifications.

If the e-mail and password combination is incorrect, a response message should appear to ask the user to try again.

If the e-mail has already been registered, it cannot be used to register a new account and a response message informing the user shall appear.

The system must be responsive through-out each action taken by the user, and most crucially, should be private and secure without any compromise on the safety of the user’s data.

Once a user has successfully logged in, they should be presented with a comprehensive monitoring feed that shows the overall system of monitoring devices connected.

The application should have a list allowing the user to navigate to different parts of the application, such as “Connected Devices”, “Settings”, “Monitor history”, among other things.

The “Connected Devices” tab should show the user all the devices that are currently linked to their account, as well as the option to remove or add other devices. Also, tapping on a specific device should take you to a screen showing more details pertaining to that monitoring device.

The “Settings” tab should contain options to customize the application and monitoring devices to a certain extent, as well as make changes to any personal information such as the e-mail, password, or phone number.

The “Monitor History” tab should present the logged history of each monitoring device, to allow the user to view past recordings.

Previous recordings are to be stored in a private server that is to be retrieved upon the request by the user to view past recordings through the Monitor History tab.

The VPS must be thoroughly secured from any unauthorized or malicious access, as it poses a very big threat to the privacy of the user.

When viewing a camera feed, the user should be able to zoom into the display if needed.

The system should maintain continuous contact with the VPS to monitor for emergencies, while the system’s database should always be ready to issue a notification to the user’s personal device in case of any emergency.

**3.2 Functional Requirements:**

1- The system must allow a user to register and login to the application.

2- The system must display and send notifications and alerts to user

3- The system must have a comprehensive camera system with recorded and live feeds

|  |  |
| --- | --- |
| **Requirement** | **Description** |
| **AccessRegistrationSystem()** | **Access the system’s registration system** |
| **EnterPersonalInfo()** | **Ask Client for his/her email and phone number and store the email as a public string and the number as private Integer** |
| **EnterUsername()** | **Ask the user for his/her username of choice and store it as public string** |
| **EnterPassword()** | **Ask the user for his/her password of choice and store it as private string** |
| **ValidatePassword()** | **Check if the passwords entered match and if it meets the requirements required** |
| **RegisterUser()** | **Register the user and all his info in the system’s database** |
| **AccessLoginSystem()** | **Access the system’s login system** |
| **CheckUser()** | **Check if the user exists in the system’s database** |
| **CheckPassword()** | **Check if the password matches the one linked with the user in the system’s database** |
| **MainMenu()** | **Access the application main menu** |
| **AccessCameraSystem()** | **Access the system’s camera system** |
| **AccessCameraFeed()** | **Access the camera’s feed** |
| **AccessLiveFeed()** | **Access the live came feed of the child’s room** |
| **AccessRecordedFeed** | **Access the recorded feed of the last 7 days** |
| **RecordFeed()** | **Record the camera feed continuously and delete the recordings older than week automatically** |
| **AccessNotificationSystem()** | **Access the system’s notification system** |
| **UpdateFeed()** | **Update the notification feed accordingly** |
| **SendAlert()** | **Send an alarming notification to the parents phone** |
| **ContactParents(PhoneNumber)** | **Contact the parents using the stored phone number in the database** |
| **ContactAuthority(Local\_Authority\_Num)** | **Contact the local authority using the stored local authority phone number in the database** |
| **ExitApp()** | **Exit the system** |

**3.3 External Interface Requirements**

The Application will have a Graphical User Interface (GIU) which will be designed in a way that makes it easy to use and simple to understand. The navigation should also be effortless for the user and there should be an assistance feature which helps the user learn how to use the Application. Additionally, it should support a variety of devices. The priority is to have support for smartphones such as android and apple phones because it’s compact and can be used anywhere. Other platforms such as Windows and OSX will be supported later.

The application will use a database in the VPS to store users' account information. For communication, there will be a local authority phone number available in case of an emergency as well as an email for user support. The product in function should depend on external hardware such as a camera system that is installed around the house. There will be a way to control the camera itself and use functions such as zoom in and zoom out.

**3.4 Non-functional requirements (NFRs) and Constraints**

**3.4.1 Non- Functional Requirements:**

**3.4.1.1 Security**

The system will be designed with a level of security that is appropriate for the sensitive information. Therefore, the virtual private server (VPS) will be more than enough to ensure that everything is secured, and the customers’ privacy desirable is achieved. Since the system does not handle high security level information such as credit card and credentials, password protecting the database. So only the users should have access to it should suffice.

**3.4.1.2 Reliability**

Regular back-ups will be made. So, there will be minimal data loss if the system stops working somehow. Thorough testing and truly challenging the system’s limits by all four team members will also ensure its reliability.

**3.4.1.3 Maintainability**

The system will be regularly maintained by a group member each month to ensure that the system is running smoothly.

**3.4.1.4 Portability**

This system will be compatible with both android and IOS with time, and it will be designed with more than those two operating systems in mind. In the case that time does not allow the system will shift focus to whichever operating system the customer prioritizes

**3.4.1.5 Capacity**

The VPS can store high amounts of data and the system will only store surveillance camera footage up to a week back. That data is what will occupy most of the system capacity. Whereas its functionality and everything else in the system falls shy in comparison and would not need a lot of space to operate

**3.4.1.6 Usability**

The system will be designed in a way that makes it intuitive and very easy to use without filling it with extra and aimless functionality. So, it will be designed to be direct, clear, and straight to the point.

**3.4.1.7 Reusability**

The system should be designed in a way that allows the database to be re-used regularly for the various silent auctions that the organization shall hold.

**3.4.2 Constraints**

**3.4.2.1 Budget**

Since physical items must be purchased. Such as the surveillance cameras and the private virtual server. Furthermore, the wages per hour for each group member is also considered a constraint

**3.4.2.2 Lack of expert knowledge**

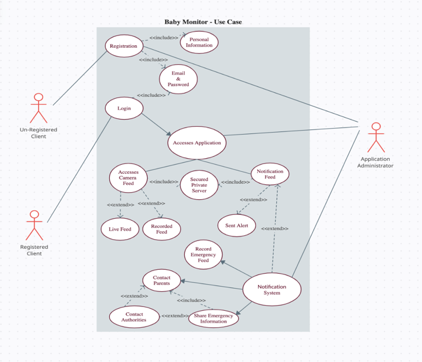
is a limitation, for the reason that some functionalities of the system require skills and expertise in topic that have high learning curves

**3.4.3.3 Time**

Because urgency is required, and some parts of development have to be rushed. Which means some compromises are inevitable

**4. MODELING AND DESIGNING THE SOLUTION**

**4.1 Use Case:**

**4.1.1 Use Cases Diagram:**

**4.1.2 Use Cases:**

**4.1.2.1 Register**

**Title: Register**

**Description:** The client registers a new user in the system.

**Actors:**

1. Client

2. Application Administrator

**Pre-conditions:**

1. The client must be unregistered.

**Post-conditions:**

1. Client is registered in the system.

**Basic flow:**

1. The system asks the client to enter his personal information.
2. The system asks the client to enter his email and password.
3. The system checks if the user is registered before.
4. The system registers the client in the system.

**Alternative flow:**

3a) The client is already registered in the system.

4) The system prompts an error message.

**4.1.2.2 Login**

**Title:** Login

**Description:** The client logs in to the system.

**Actors:**

1. Client

2. Application Administrator

**Pre-conditions:**

1. If unregistered, the client must register.
2. If registered, the client must log in.

**Post-conditions:**

1. Client is logged in.

**Basic flow:**

1. The system validates the client’s username and password and logs him into the system
2. The system displays the main user interface

**Alternative flow:**

1a) The name does not match the password

2) The system prompts an error message

**4.1.2.3 Access Camera Feed**

**Title:** Access Camera Feed

**Description:** The client can access either the live feed or pre-recorded feed in case of an emergency

**Actors:**

1. Client

2. Application Administrator

**Pre-conditions:**

1. The client must be logged in.

**Post-conditions:**

1. System updates the new information and works normally.

**Basic flow:**

1) The client accesses camera feed.

2) The system asks which feed to access live or recorded

3) The client accesses the live feed.

4) The client leaves.

5) The system updates the new information and works normally.

**Alternative flow:**

3a) The client accesses recorded camera feed

4) The client leaves.

5) The system updates the new information and works normally.

**4.1.2.4 Send Alert**

**Title:** Send Alert

**Description:** The system detects an emergency, so it sends an alert to the parents**.**

**Actor:** Child

**Preconditions:**

1. Client is Logged in and system
2. Surveillance camera is monitoring the child.

**Post conditions:**

1. The parents have been notified about the emergency.

**Basic flow:**

1)The system detects an emergency

2) The system notifies the parents of the emergency.

3) The parents confirm to the system that they received the notification.

4) The system updates the new information and keeps monitoring the emergency.

**Alternative flow:**

3a) The parents do not confirm to the system that they received the notification.

4) The system notifies the local authority.

5) The system notifies the parent that the local authority was notified.

6) The system updates the new information and keeps monitoring the emergency.

**4.1.2.5 Contact Parents**

**Title:** Contact Parents

**Description:** System contacts both parents on the state of the child

**Actor:**

1. Client

2. Application Administrator

**Preconditions:**

1. Client is logged in
2. System has the client’s contact information.

**Post conditions:**

1. System has contacted the parents.

**Basic flow:**

1) The system detects the state of the child.

2) The system contacts the parents on the state of their child.

3) System updates the new information and works normally.

**Alternative flow:**

1a) The system detects an emergency

2) The system alerts the parents of the emergency

3) The Parents confirm receiving the notification

4) System updates the new information and works normally.

**4.1.2.6 Contact Authority**

**Title:** Contact Authority

**Description:** System contacts the local authority

**Actors:**

1. Client

2. Application Administrator

**Preconditions:**

1. Client is logged in
2. System has the local authority contact information.

**Post conditions:**

1. System has contacted the local authority.

**Basic flow:**

1) The system detects an emergency.

2) The system contacts the parents on the state of their child.

3) The parents do not confirm receiving the message

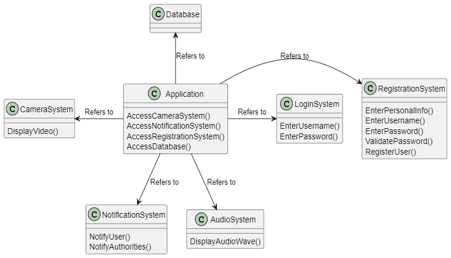
4) The system contacts the local authority.

5) The system informs the parent that the local authority was notified.

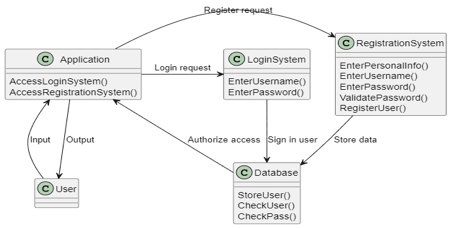
6) The system updates the new information and keeps monitoring the emergency.

**4.2 Class Diagrams**

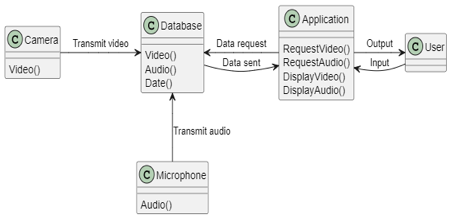
**4.2.1 Application Class Diagrams**

****

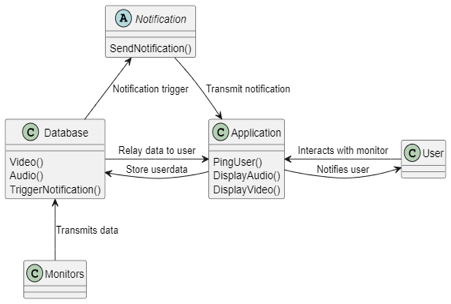
**4.2.2 Registration & Login Systems Class Diagrams**

****

**4.2.3 Camera System Class Diagram**

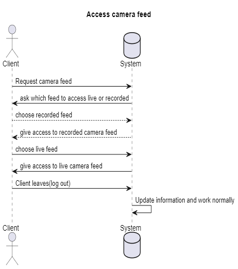
****

**4.2.4 Notification Class Diagram**

****

**4.3 Sequential Diagrams**

Diagram

Description automatically generated

UC3: Access Camera System

Client Requests Access to Camera Feed

System asks which feed to access

Client Chooses Live Feed

System gives access

Client Logs Out

System Updates Information and work normally

Alternatives:

3A. Client Chooses Recorded Feed

System gives access

Client Logs Out

System Updates Information and work normally

UC2: Access Registration and Login Systems

Unregistered Client Accesses Registration System

System asks for personal Information

System asks for Username and Password

System checks if user exists in Database

User does not exist

Register new user in Database

Client Access Login System

System asks for email and password

System verifies with Database

User is found in Database

System Logs the client in

Alternatives:

1A. Registered Client Accesses Login System

2. System asks for email and password

3. System verifies with Database

4. User is not found in Database

5. System asks the client to try again

6. System verifies with Database

7. User is found in Database

8. System Logs the Client in

Graphical user interface, application

Description automatically generated

**4.4 State Diagrams**

UC1: Access Application

Client Accesses Application

System asks to choose Registration or Login

Client Chooses to Login

System accesses login system

Client accesses the app main menu

System asks to choose camera or notification system

Client chooses camera system

System accesses camera system

Exit System

Alternatives:

A)

3A. Client chooses Registration

4. System accesses registration system

5.Exit System

B)

7A. Client chooses notification system

8. System accesses notification system

9. Exit system

**4.4.1 Application State Diagram**

Diagram

Description automatically generated

UC1: Access Application

Client Accesses Application

System asks to choose Registration or Login

Client Chooses to Login

System accesses login system

Client accesses the app main menu

System asks to choose camera or notification system

Client chooses camera system

System accesses camera system

Exit System

Alternatives:

A)

3A. Client chooses Registration

4. System accesses registration system

5.Exit System

B)

7A. Client chooses notification system

8. System accesses notification system

9. Exit system

**4.4.2 Registration System State Diagram**

Diagram

Description automatically generated

Alternatives:

A)

7A. Password not validated

8. System ask client to retry or exit system

9. Client chooses to retry

10. System asks for password

11. Password is validated

12. Register new user in Database

13. Access Login System

B)

7A. Password not validated

8. System ask client to retry or exit system

9. Client chooses exit

10. System exits

UC2: Access Registration System

Client Accesses Registration System

System asks for personal Information

System asks for Username

System asks for password

System asks for password again

System validates password

Password is validated

Register new user in Database

Access Login System

**4.4.3 Login System State Diagram**Diagram

Description automatically generated

UC2: Access Login System

Client Accesses Login System

System asks for Username and Password

Client Enters Username and Password

System checks if user in database

User in database

System checks if the password entered match

Password match

Client accesses application main menu

Alternatives:

A)

4A. User is not in database

5. System ask client to retry, register or exit system

6. Client chooses to exit

7. System exits

B)

6A. Password does not match

8. System ask client to retry, register or exit system

9. Client chooses to retry

10. System ask for Username and password

11. Client Enters Username and Password

12. System checks if user in database

13. User in database

14. System checks if Passwords match

15. Passwords match

16. Client accesses application main menu

**4.4.4 Camera System State Diagram**

Diagram

Description automatically generated

UC3: Access Camera System

Client Accesses Camera Feed

System asks which feed to access

Client Chooses Live Feed

System gives access to live feed

Client accesses main menu

Alternatives:

A)

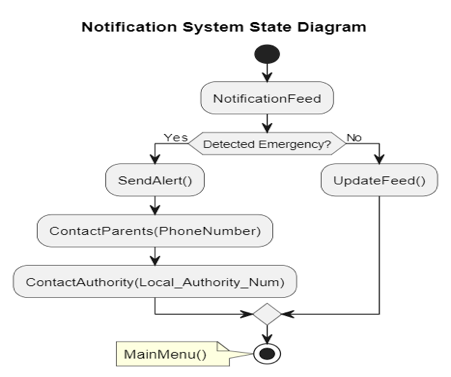
3A. Client Chooses Recorded Feed

System gives access to recorded feed

Client accesses main menu

System Updates Information and work normally

**4.4.4 Notification System State Diagram**

****

UC4: Access Notification System

Client accesses notification system

System displays notification feed

System detects state of child

System Detected emergency

System Contact Clients

Clients do not respond

System Contacts local authority

Local Authority confirm notification

Client accesses main menu

Alternatives:

A)

3A. System detects nothing

4. System Update feed and work normally

5. Client accesses main menu

B)

6A. Clients confirm notification

7. System updates feed and work normally

8. Client accesses main menu

**4.5 Data Dictionary**

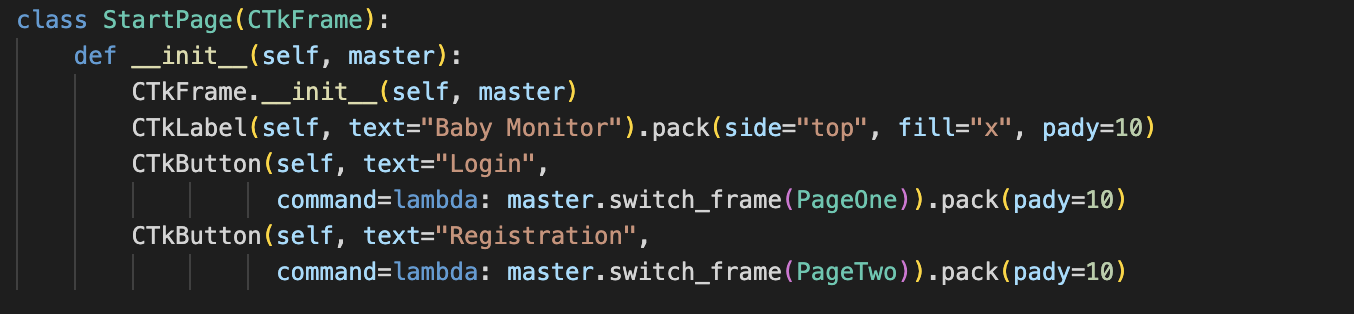
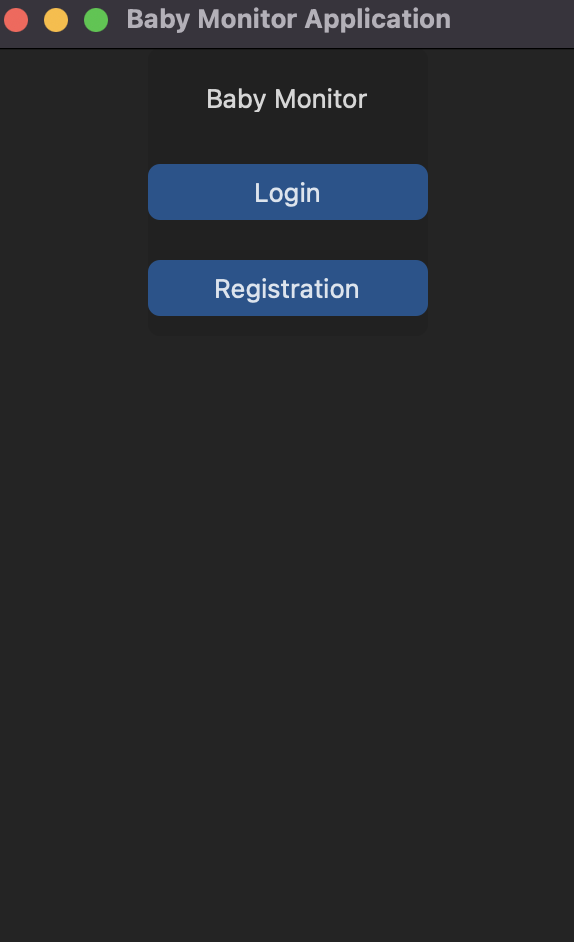
|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Field Length** | **Description** |
| Email | Public String | 30 characters | Client email |
| PhoneNumber | Private Integer | 15 characters | Client phone number |
| Username | Public String | 15 characters | Client’s username |
| Password | Private String | 20 characters | Client’s login password |
| Local\_Authority\_Num | Public Constant Integer | 15 characters | Phone number to contact local authority |

**4.6. Tools**

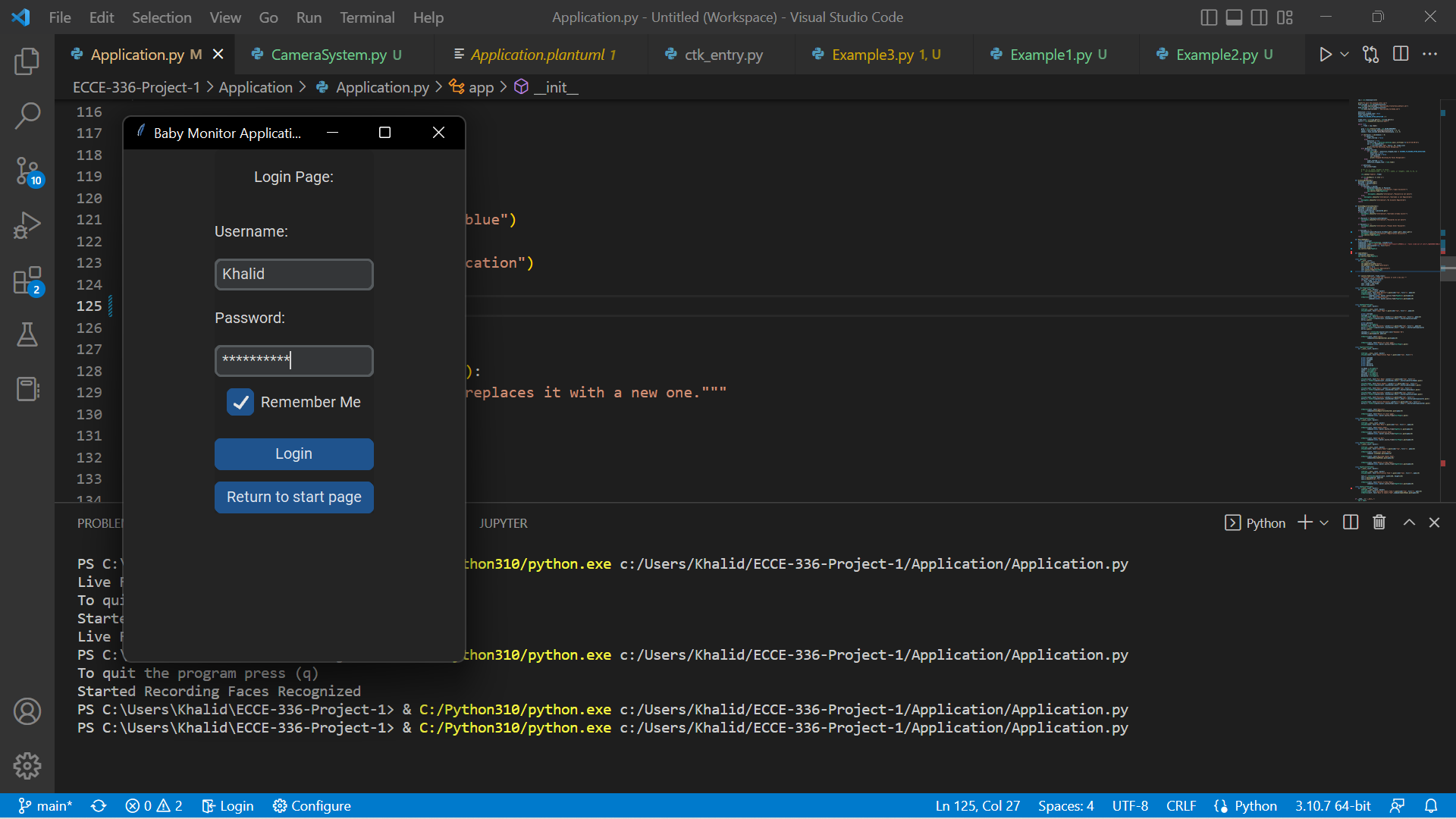
|  |  |
| --- | --- |
| **Tools** | **Purpose** |
| Plant UML | Drawing Diagrams |
| Microsoft Word | Processing Words |
| Visual Studio Code | Prototyping and Development |
| Github | Project Management |

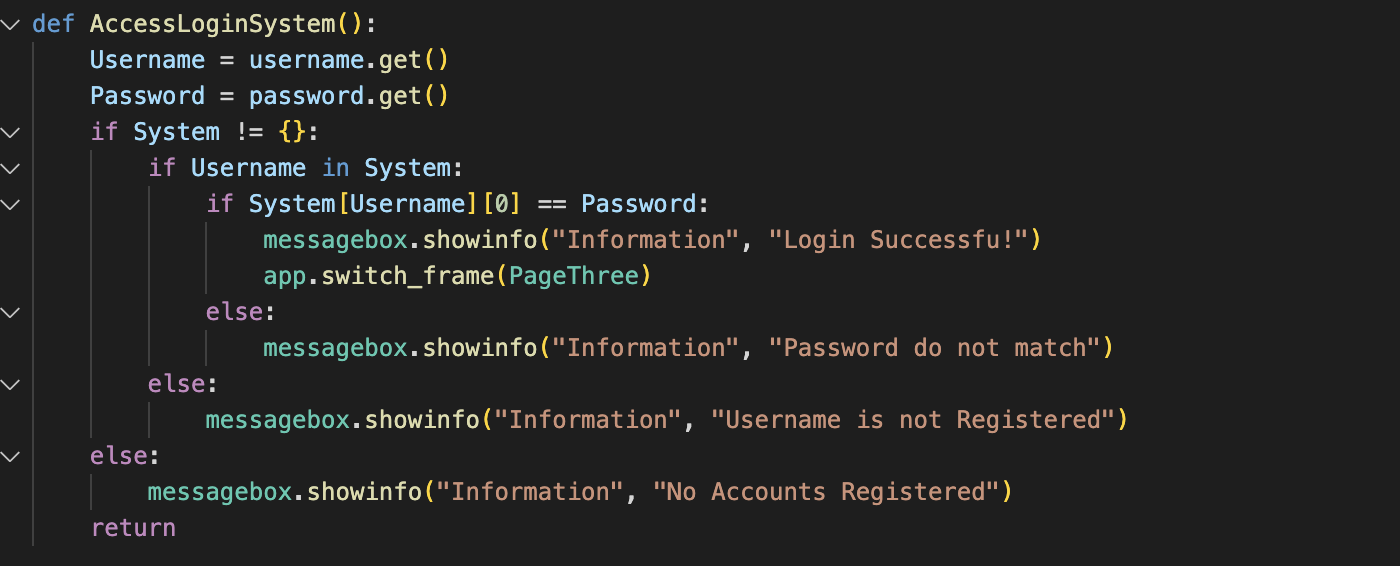
**5. IMPLEMENTATION**

**5.1 Start Page**

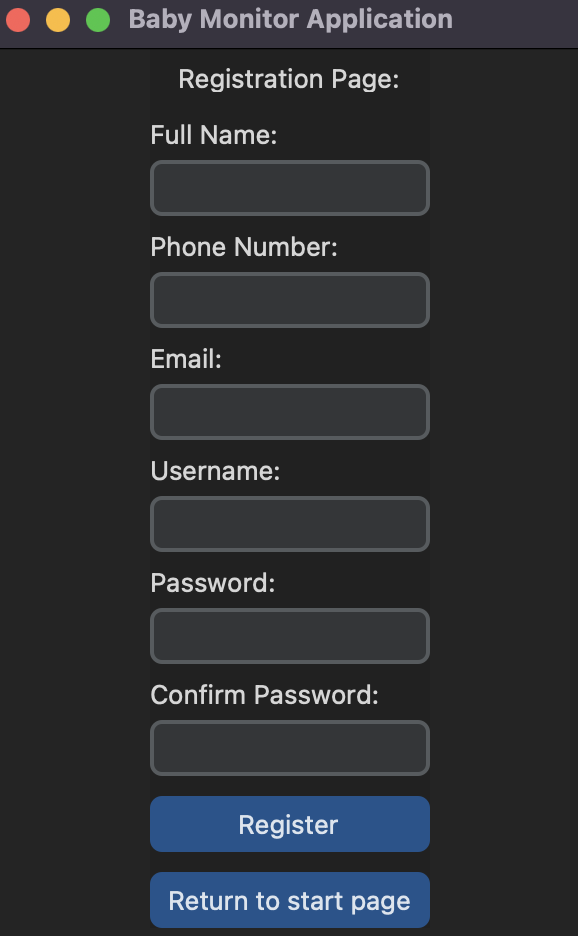
****

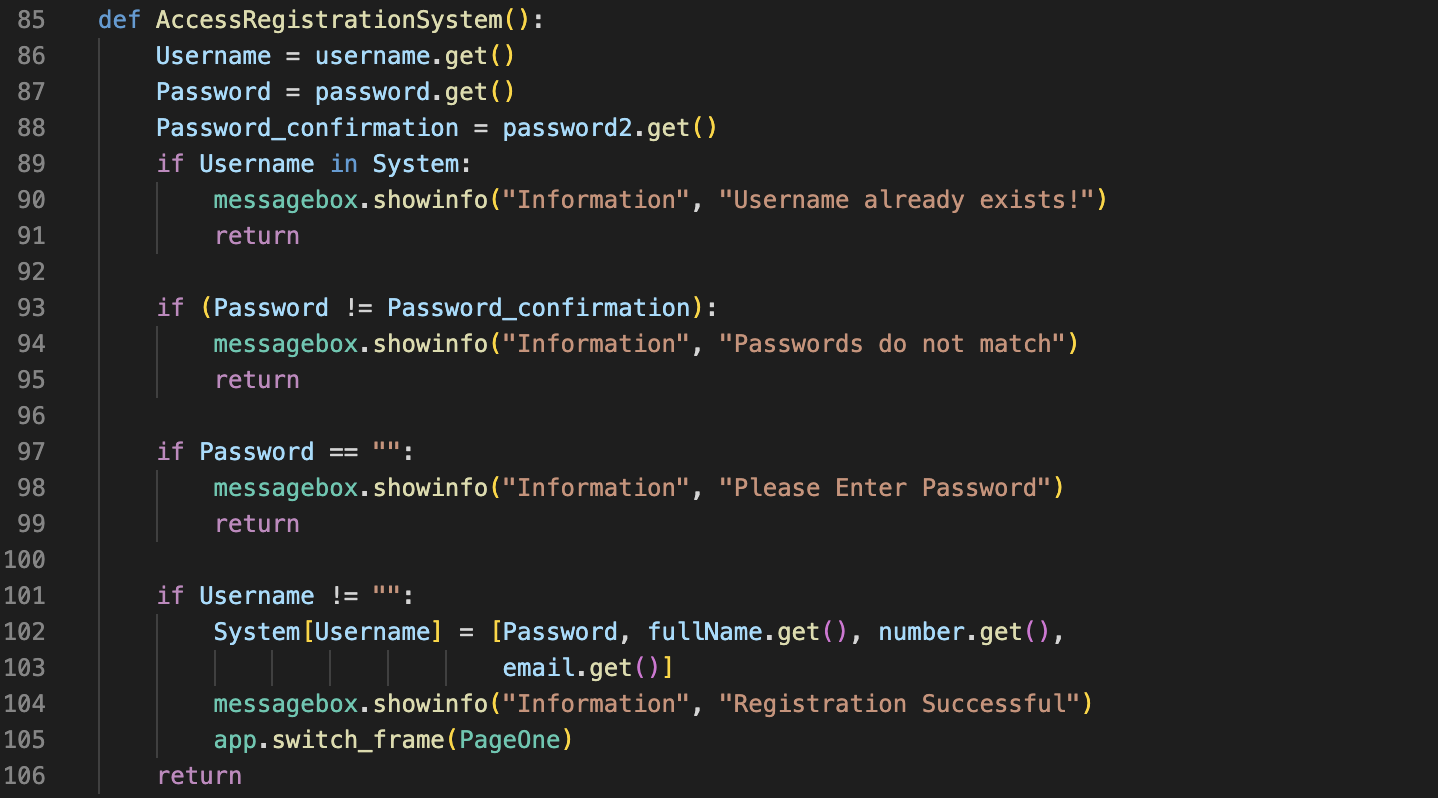
**5.2 Login Page**

****

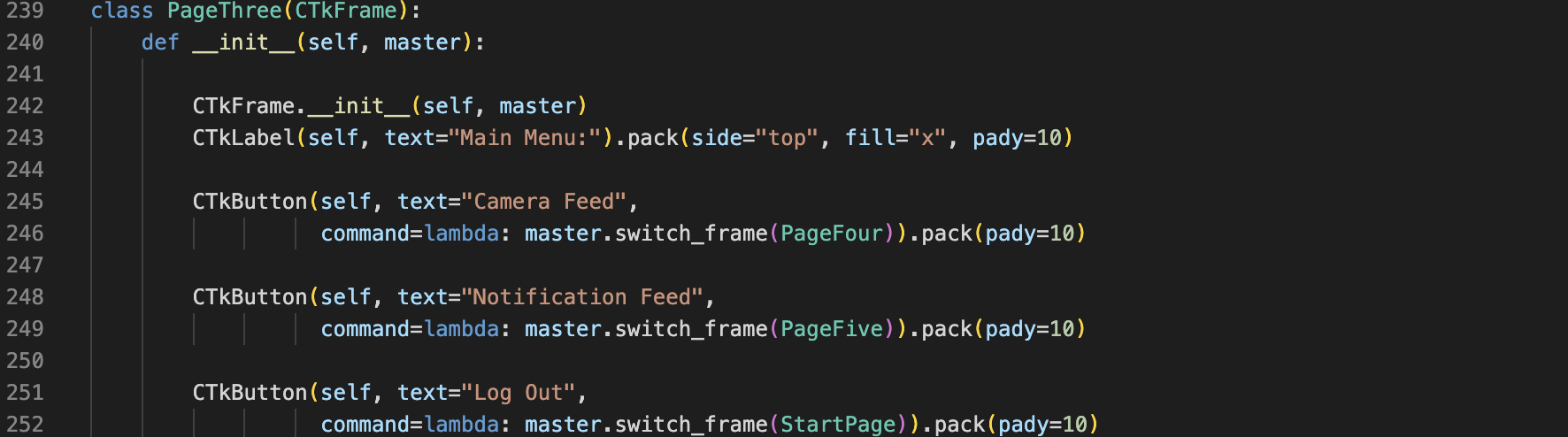
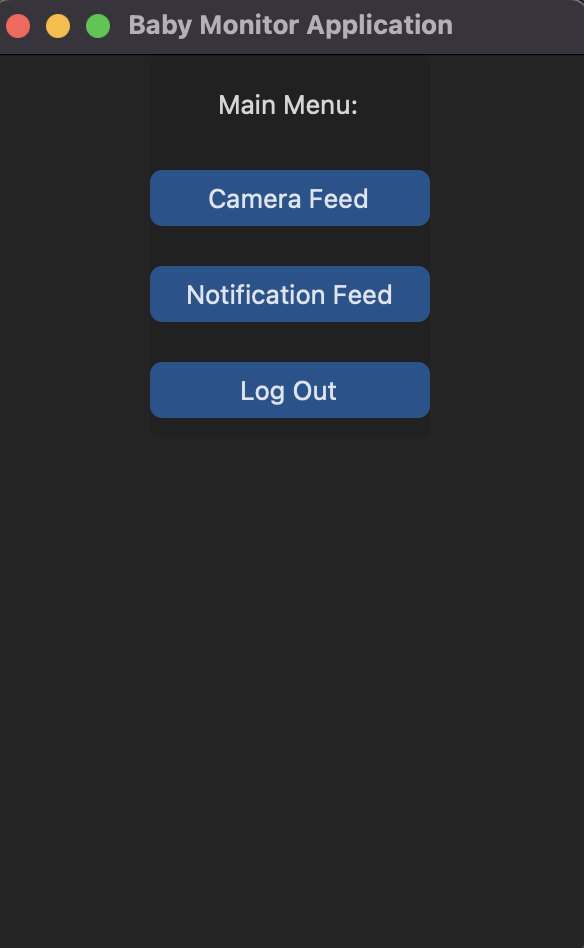
****

**5.3 Registration Page**

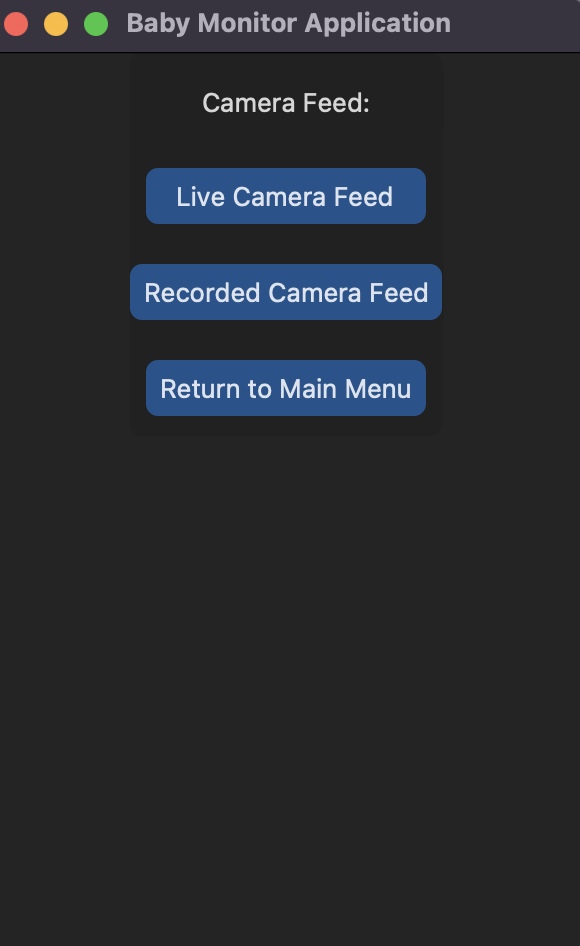
****

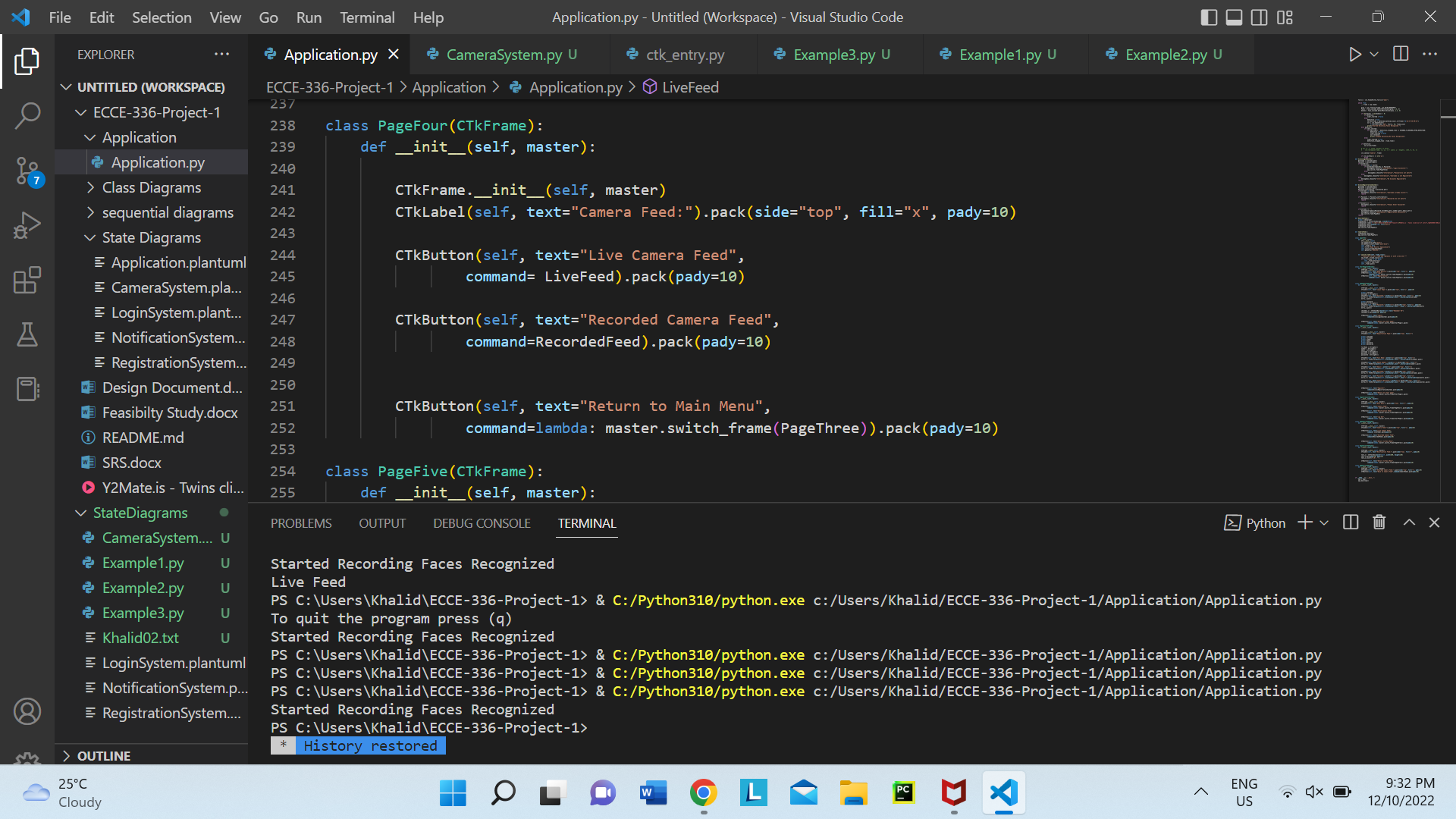
****

**5.4 Main Menu**

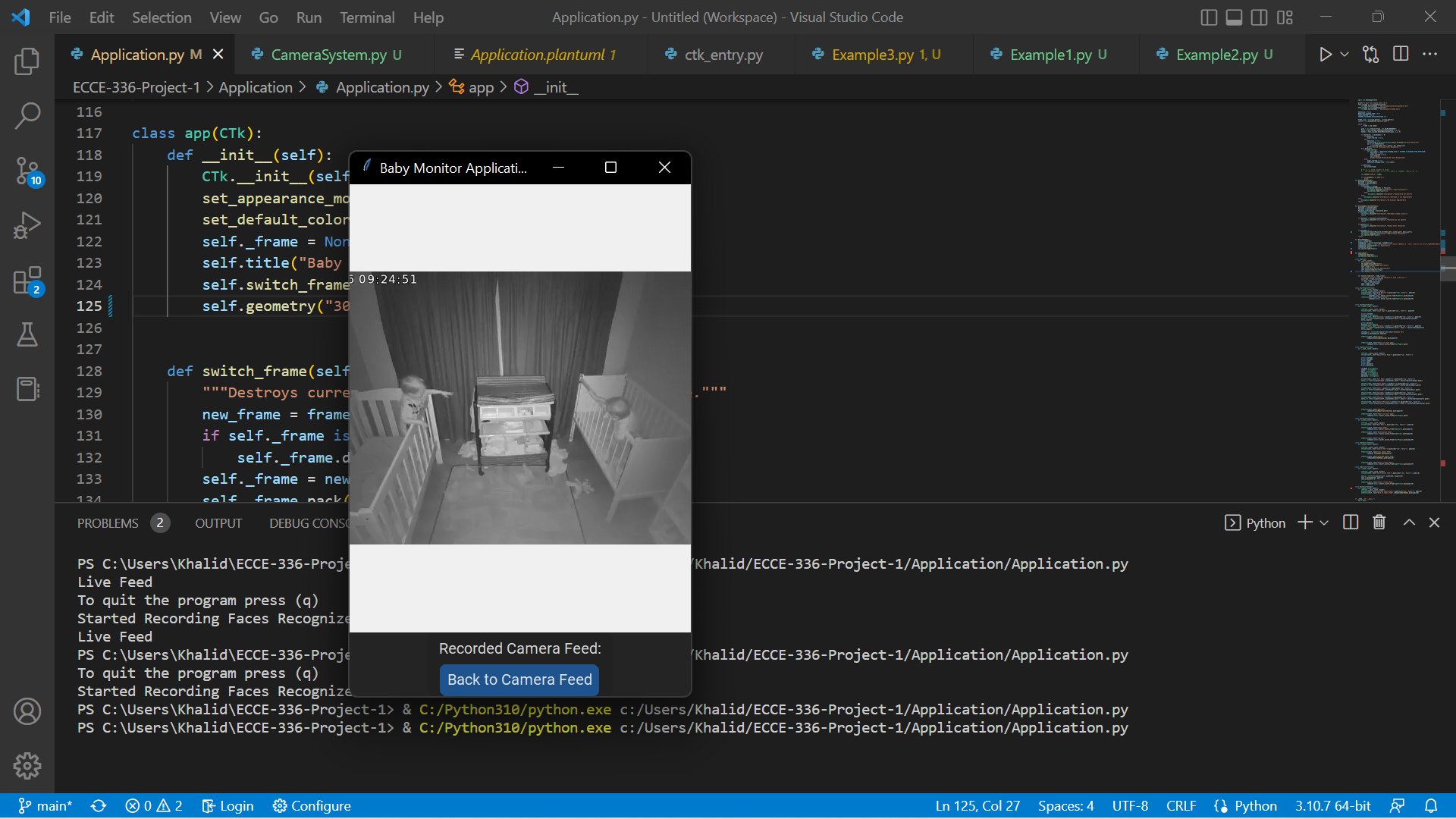
****

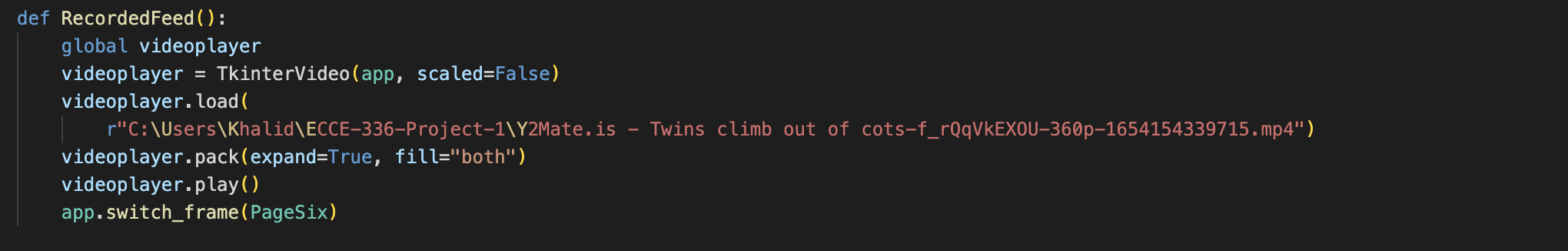
**5.5 Camera Feed Menu**

****

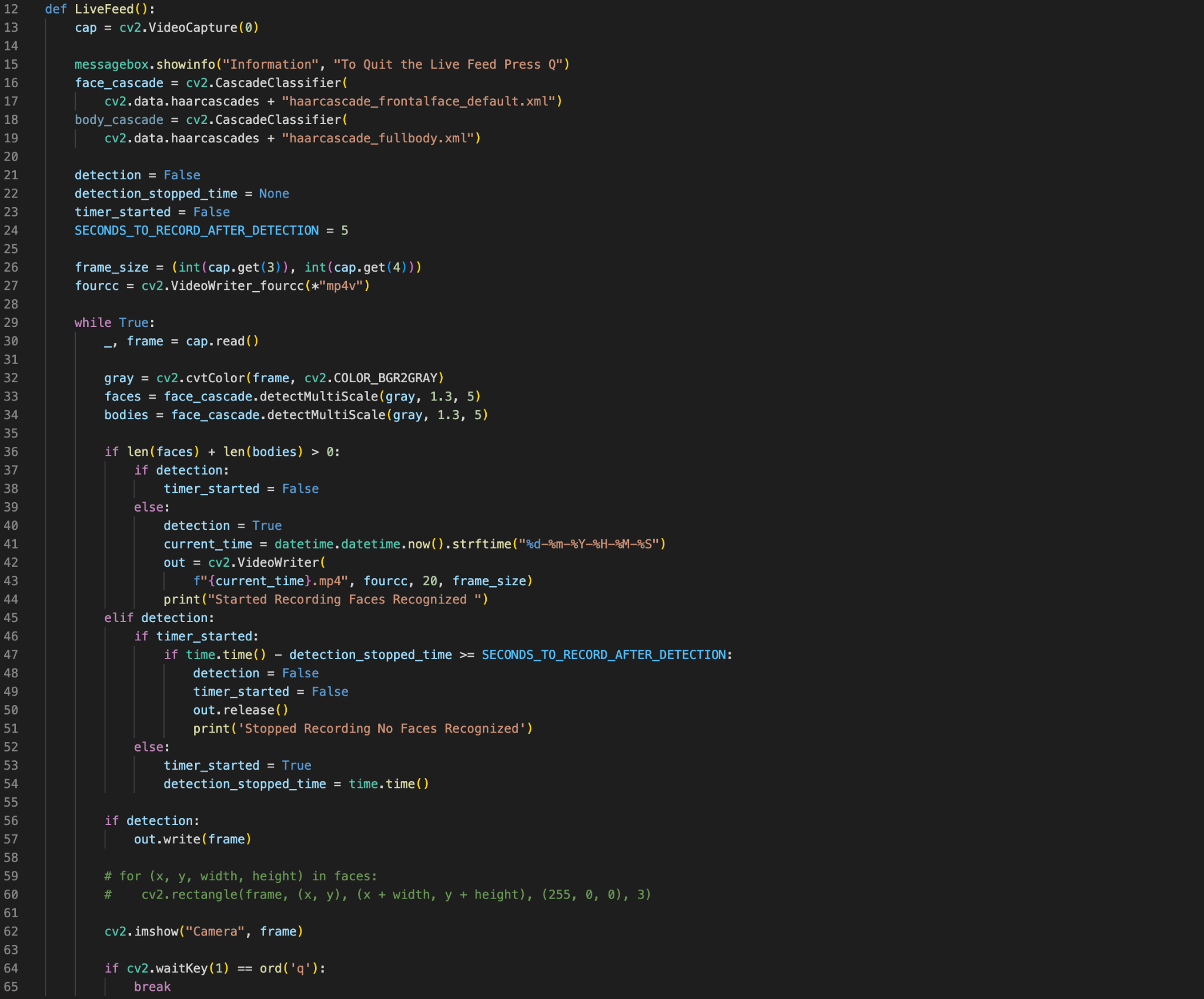
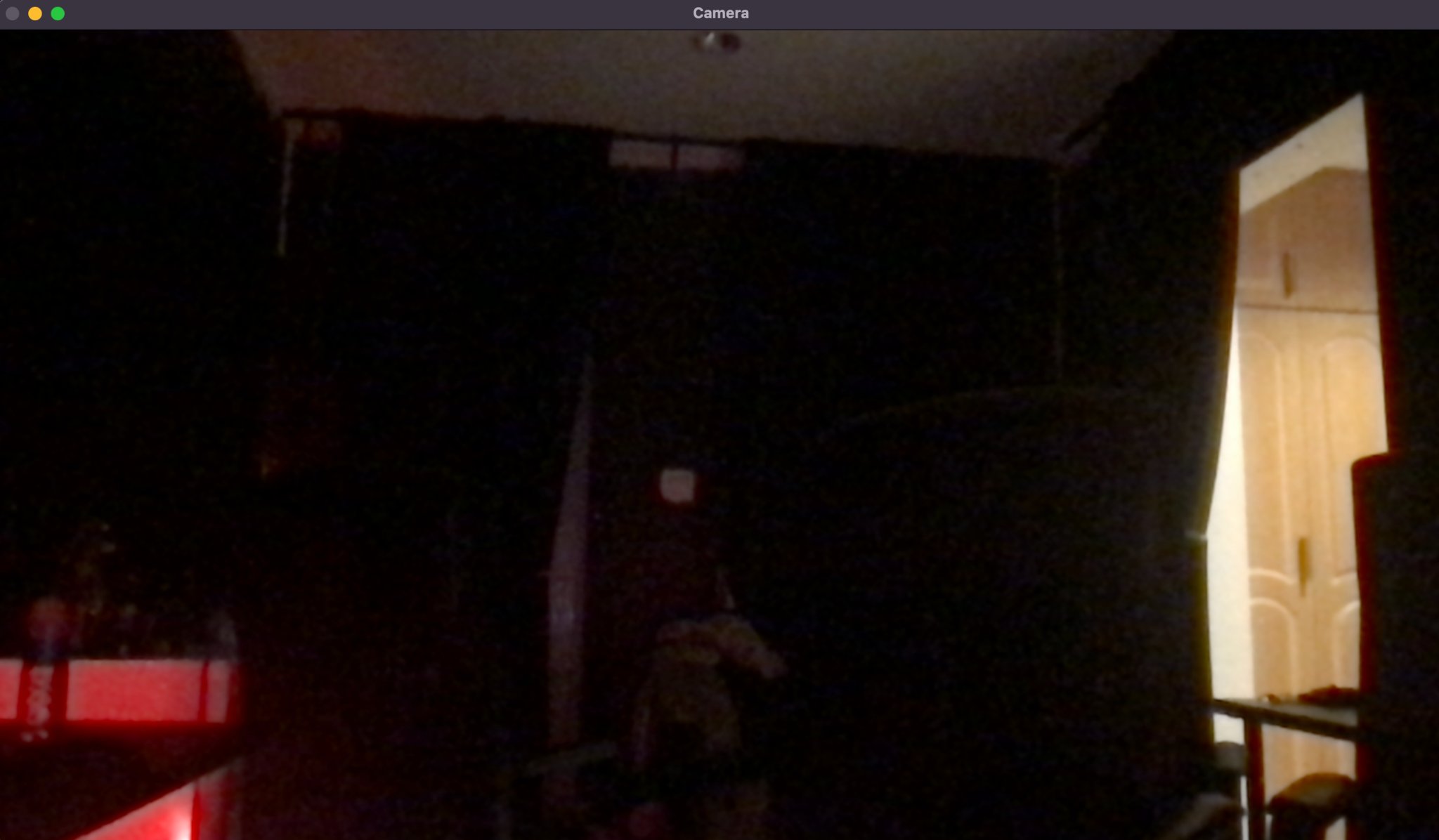
****

**5.6 Recorded Feed**

****

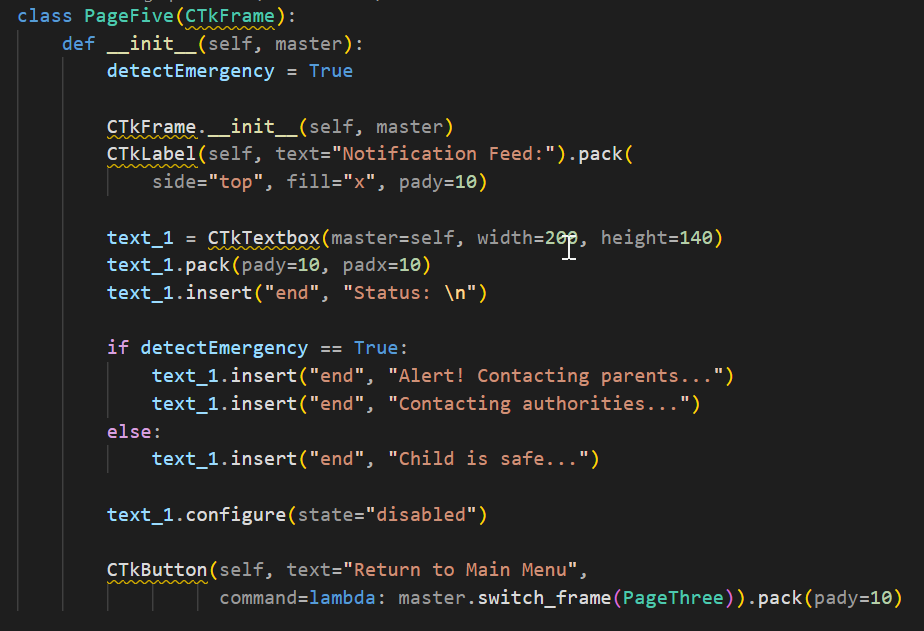
****

**5.7 Live Feed**

****

**5.8 Notification Page**

****

****