The Automatic Alarm System using Motion Detection

Sep 2022

Acknowledgement

I would like to thank my friends and families for their encouragement and support during the writing of this report. Without them, I might not be able to finish this report on time.

Table of Contents

Abstract

The Automatic Alarm System uses OpenCV to detect motion, and alarm the user to prevent potential risks in his residence or office. With the improvement of life quality, more and more people concern about residence and office security. Most monitoring systems only provide monitoring functions for further checking and tracing, but don’t have real-time alarming functions. This system focuses on the Gmail user and allows the user to receive alarm messages when the system detects any motion in their residence. The Automatic Alarm system uses OpenCV library for image processing, video analysis, and object detection, and uses the smtplib library for sending alarm message and captured frame. The combination of OpenCV and smtplib provide an efficient and convenient way to detect motion and send messages.

Introduction

OpenCV (Open Source computer vision) is a library of programming functions mainly aimed at real-time computer vision (Baksheev, A., et al, 2012). OpenCV is written in C++ and supports Python, Java, and MATLAB. In this project, OpenCV-Python is used for image processing, video analysis, and object detection. Home and office security have been a serious issue, and the monitoring system is popular. Most monitoring systems only provide monitoring functions for further checking and tracing, but don't have real-time alarming functions. This project is an automatic alarm system that uses OpenCV to detect motion and send alarm messages via the smtplib library to the user's Gmail in real time. Adrian Rosebroke built a motion detection and tracking system in 2015 using Python and OpenCV (Rosebroke,2019). He deployed the system to a Raspberry Pi with a camera connected. His system only achieved motion detection and didn't have the alarm part (Rosebroke,2019). This project also uses Python and OpenCV. I use the PC camera and add the function to alarm the user when the system is triggered by the motion, which can prevent further loss.

The main objective of the project is building an automatic alarm system to help people detect any potential risk when they leave their residences or offices. The system is supposed to detect any motion with the help of a camera and motion detection functions, and inform the user in time by sending a message to the user's Gmail by image processing technique.

The upcoming sections include the system requirement specification and the system design requirement, and the prototyping of the project will be provided, followed with the system evaluation, discussion, and future work.

System Design and Implementation

Requirement Specification

1. Introduction

This section gives a scope description and overview of everything included in this Software Requirements Specifications (SRS) document. The purpose of this document is to describe and make a list of the abbreviations, and the definitions appear in the whole document.

1.1 Purpose

The purpose of this document is to present a Software Requirements Specification of an automatic alarm system which includes a motion detection camera and control software. It will illustrate the purpose and feature of the system, the interface of the system. Explain system constraints and complete declarations for the development of the system. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 Scope

This software “Automatic Alarm System” has the main object as building an automatic alarm system which detected motion in residences or offices to help people get rid of potential risks from thieves when they leave their residences or offices. The system is supposed to detect motion in the monitored area with a PC camera or an external camera; then the image processing technique will analyze the motion to check whether the motion is harmful. If so, the system will inform the user in time by sending a message to Gmail.

By receiving this message, people will catch the potential risk in the rooms immediately to achieve the goal as an automatic alarm system.

The whole alarm system is based on the OpenCV library and smtplib library running on the python environment and is designed for normal people even though they do not know electronics and computer coding skills. To use this system, users can initialize the system as easy as other software.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
User	Someone who interacts with the program
IDE	Integrated Development Environment
OpenCV	Open Source Computer Vision Library
Python	An interpreted, high-level, general-purpose programming language.
PPF	Prominent Processing Feature
PAF	Prominent Accuracy Feature
DESC	Description
RAT	Rational
DEP	Dependency

Table 1: Definitions, Acronyms, and Abbreviations

1.4 Overview

This document includes three chapters. The second chapter, the Overall Description section, provides an overview of the system functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the second chapter. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter, External Interface Requirement section of this document is written primarily for the description of different system interfaces. It also clarifies the details about the functionality of the product and the features.

2. The Overall Description

This section will give an overview of the whole system. The functionality and the interactions among all basic systems will be introduced. It will also describe what type of users that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 Product Perspective

This system mainly contains three parts: hardware input part, web server processor part and mobile software output part. For the first part, there will be a PC camera or an external camera put in the places to be protected, which will record every frame. When the motion analysis control center detects any motion, the current frame and an Alarm message will be sent to the user's Gmail. The web server processor is for user to enter email address and password to log in Gmail, and the mobile software is for the user to receive the captured motion frame and the alarm message. By finishing this process, the workflow of the alarm system is completed. In Figure 1 below, the camera manages the input of the recording frame, the processor running in a python environment manages the motion analysis, and the webserver and the mobile software manage the login and message receive.

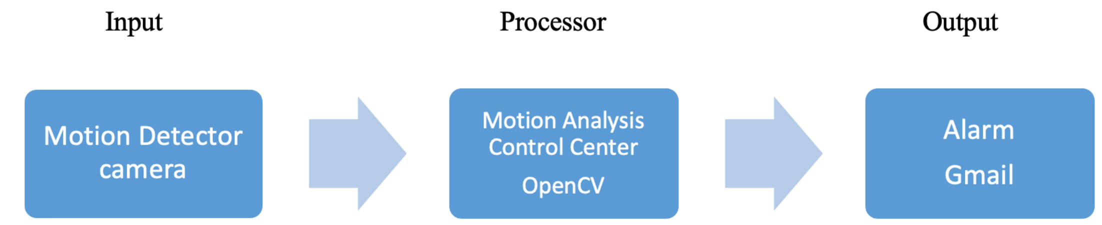


Figure 1. Block Diagram

2.1.1 Interfaces

Considering that the software is designed for normal people, the user interfaces of software should have clear guidance and function revealing. There are two labels in Figure 2: Gmail and Password, and two entries for the user to input. There are Two buttons: start and quit. The start button is for the user to start the system with the correct Gmail and password; the quit is for users to quit the system.

After the user enters the correct Gmail and password as showing in Figure 3, he will be able to start the system for motion detection. When the user wants to stop the system, he can click the quit button to stop the system and close all windows.

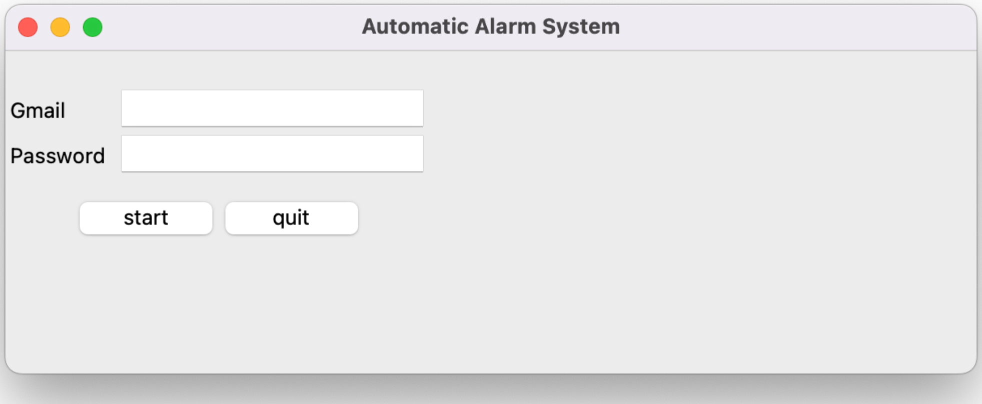
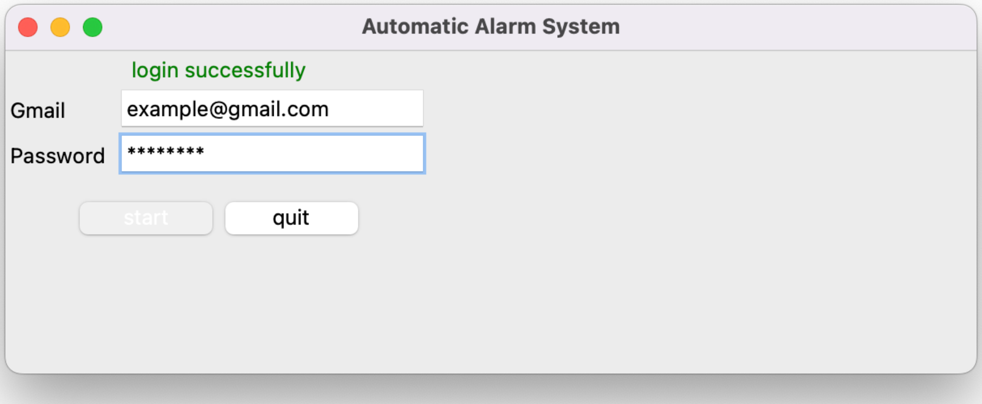


Figure 2: Login window.



login successfully

Gmail: example@gmail.com

Password: ********

Figure 3: login success window.

2.1.2 Hardware Interfaces

Because this software requires a camera, the proper interface to access the camera is important. The video from the camera should be sent to the computer through the interface. Besides, the processor is supposed to deal with the video data.

To achieve this goal and get access to the camera, a 1.0 GHz or higher Processor, 1 GB memory (32 bits) Or 2GB memory (64 bits), 1 GB free hard disk space, DirectX 9 graphics device with WDDM 1.0 or later driver, and at least 128MB of graphic memory is needed for this system.

2.1.3 Software Interfaces

After the camera transfers the video signal to the processor, the software in the processor will use OpenCV to achieve motion detection. OpenCV is a library of programming functions mainly aimed at real-time computer vision; it can analyze the motion happens in the video and unscramble the information in the movement. After the processor processes the data, the email transmission will be achieved by the smtplib library.

2.1.4 Communications Interfaces

Since the system needs the video recorded by the camera, so the video from the camera should be translated into the proper format that is suitable for OpenCV. Also, the video will be parsed into appropriate signals and transmitted to email information.

2.1.5 Memory Constraints

To store the video in the camera, a 1 GB memory (32 bits) Or 2GB memory (64 bits) is needed. Also, for dealing with the graphic information, the DirectX 9 graphics device with WDDM 1.0 or later driver, 128MB of graphic memory is needed in the processor server.

2.2 Product Functions

The system will first use a camera in the monitored area to record the motion happens in that area and then analyze the motion in the video using OpenCV. OpenCV is a library of programming functions mainly aimed at real-time computer vision, it can analyze the motion happens in the video and unscramble the information in the movement. After the processor processes the data, Python will be used to access Gmail through the smtplib library. Then the user will get an alarm email in his own Gmail.

2.3 User Characteristics

There is only one kind of user in this system. They are the owner of the automatic alarm system, who are also the Gmail receiver. People who use this system in their residential or office will be the only user of the system. It could be the manager of a company or a family member of a family. Users can use this system to get rid of the risks of danger.

2.4 Constraints

Because the system is based on the video recorded in the places, so the camera chosen will be a big problem. OpenCV can process motion only if the motion is clear and convincing to recognize. The clearer the video is, the more specific the result will be. In this case, the recording camera and the transfer process will be essential. If the analysis system is not accurate enough, then the whole system will not achieve the objective to forecast risk not to prevent the risk.

2.5 Assumptions and Dependencies

The alarm server is based on sending messages to the users' Gmail account. So, the problem is that the processor needs the network while working. If there is no network, the message will not be sent out. Similarly, if the user does not login his Gmail account, he will not be able to receive the email.

2.6 Apportioning of Requirements

In the case that the project is delayed, some requirements could be transferred to the next version of the application. Those requirements are to be developed in the second release. See Appendix A.

3. Specific Requirements

This section contains all the software requirements at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that system meets those requirements.

3.1 Functional requirements

3.1.1 Functional requirement 1

Feature ID: FR-1

Feature Name: User system login and Gmail login

Description: The system shall allow the user to log in the system with the correct Gmail and password.

3.1.2 Functional requirement 2

Feature ID: FR-2

Feature Name: System start

Description: A “start” button is needed. After the user successfully login, the system shall allow the user to start the system.

3.1.3 Functional requirement 3

Feature ID: FR-3

Feature Name: System quit

Description: A “quit” button is needed. The system shall allow the user to quit and close all the windows.

3.1.4 Functional requirement 4

Feature ID: FR-4

Feature Name: Video Capture

Description: The system shall be connected to a PC camera or an external camera to capture and save video stream.

3.1.5 Functional requirement 5

Feature ID: FR-5

Feature Name: Motion Detection

Description: In the motion detection class, the system shall provide functions to detect motion from the video captured by the camera.

3.1.6 Functional requirement 6

Feature ID: FR-6

Feature Name: Send alarm message and captured frame

Description: After detecting the motion, the system shall send an alarm email and the captured frame to the user's Gmail account.

3.2 Performance Requirements

3.2.1 Performance requirement 1

Feature ID: PR-1

Feature Name: Start Button response time

Description: The "start" button shall take at most 1 second to start the system and begin motion detection.

3.2.2 Performance requirement 2

Feature ID: PR-2

Feature Name: Quit Button response time

Description: The "quit" button shall take 2 s to quit the system and close all windows.

3.2.3 Performance requirement 3

Feature ID: PR-3

Feature Name: Prominent Accuracy Feature

Description: The accuracy of the alarm message should be higher than an ideal standard (if possibly, 95 out of 100 times alarm should be accurate). Besides, the systems should tolerate 1 out of 100 times theft which is not detected by the system.

3.2.4 Performance requirement 4

Feature ID: PR-4

Feature Name: Motion Detection interval

Description: Motion detection shall detect 5 times/second.

3.3 Hardware Interfaces

PC with camera and environment that can run python environment.

Need a proper interface to access the camera. The video from the camera should be sent to the computer through the interface.

3.4 Logical Database Requirements

There is no need to use a database in our project. Because the user's data storage and processing are entirely done locally, and the storage of the videos in the form of local storage does not require uploading data to the server. It is a standalone desktop program. If we want to upgrade the program, we would consider recording the relevant clip and upload it to the server. The user name and password will be saved to a local file.

The data descriptions of each of the data entities are as follows:

User Entity

Data Item	Type	Description	Comment
Gmail	String	User’s Gmail account	
Password	Text	Password user typed in	Maybe several

Table 2: User entity

Camera Entity

Data Item	Type	Description	Comment
ID	String	ID of the Camera	Maybe recorded as a hex
Definition	Int	The clarity of the camera	Maybe several modes
Time	int	The system time of the camera	

Table 3: Camera entity

3.5 Design Constraints

3.5.1 Standards Compliance

If the user reports an error when using the program, a report will be sent, for example, the storage fails, and the message sending failure.

3.5.1.1 Report format

The report needs to be full of the following content:

-1. User's account number name and its corresponding Gmail ID

-2. Type of Error

-3. System Version

-4. Report Generation Time

3.5.1.2 Audit Tracing

All error reports and run logs are stored under a file named UserLog. The longest retention period for these files is approximately one week.

3.6 Software System Attributes

3.6.1 Reliability

The program relies on Motion Detection from OpenCV, so the overall reliability of the program is high. The program should have no crashes at execution time, and no errors should occur during the execution period. Motion Detection technology requires a success rate of more than 85%, allowing a certain degree of misjudgment.

3.6.2 Availability

The system allows the user to restart the program when the program crashes, and the program requires an individual history record. Restore the user's history if conditions permit. The system shall allow users to restart the application after failure with the loss of at most 30 seconds' camera's videos.

3.6.3 Security

The security of the user password is protected by LFSR algorithm. To prevent the local video storage failure, the system shall use an RSA cytological algorithm to upload the suspect video clip to the server. The program does not have permission to record normal videos and upload them, so that this program will divulge no personal information. Keep specific log or history data sets

3.6.4 Maintainability

The program is highly maintainable because the organization of the program is clear and straightforward. The OpenCV and Python API are all well written and tested.

3.7 Use Case Diagram

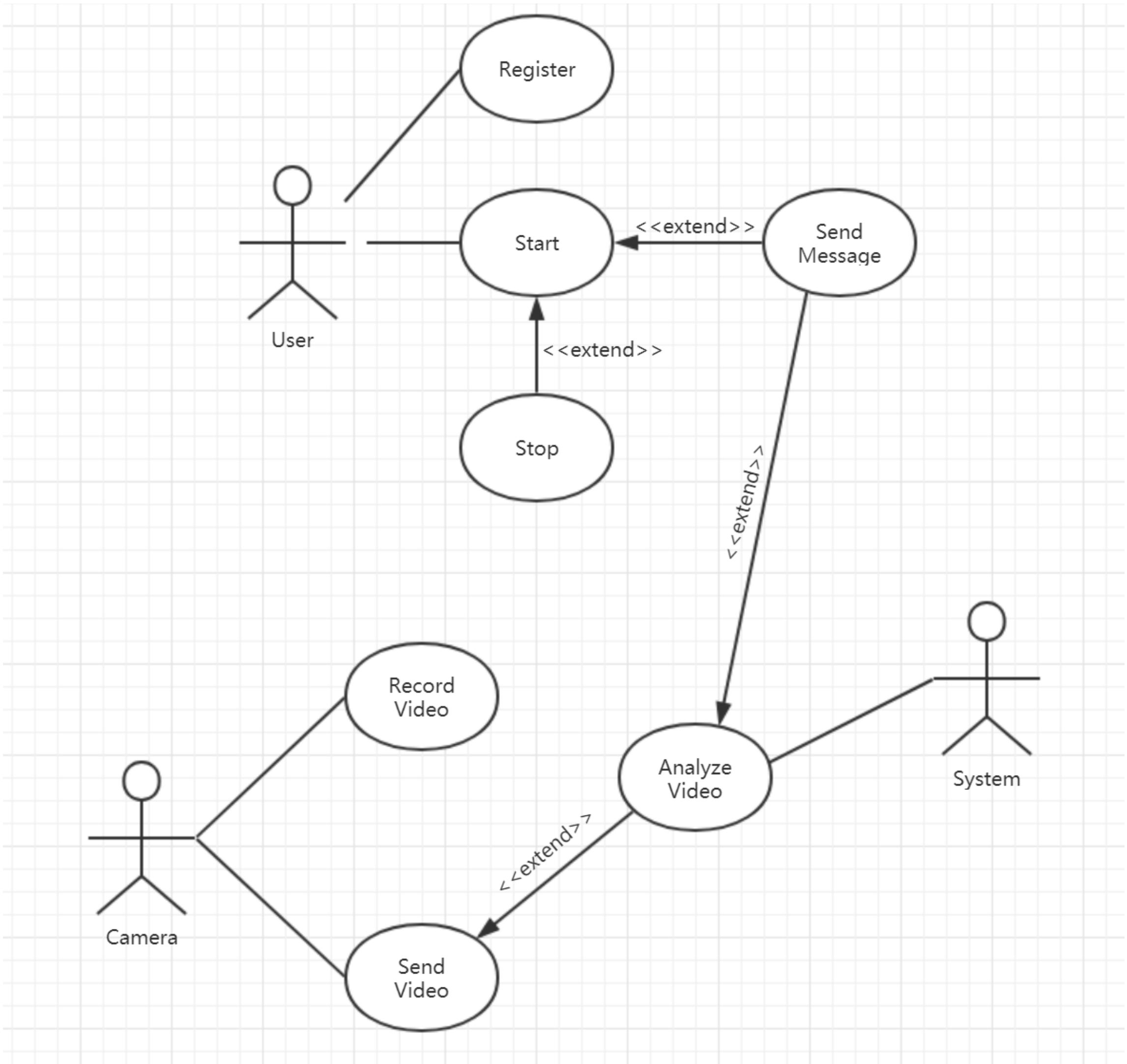


Figure 4: Use case diagram.

3.8 Class Diagram

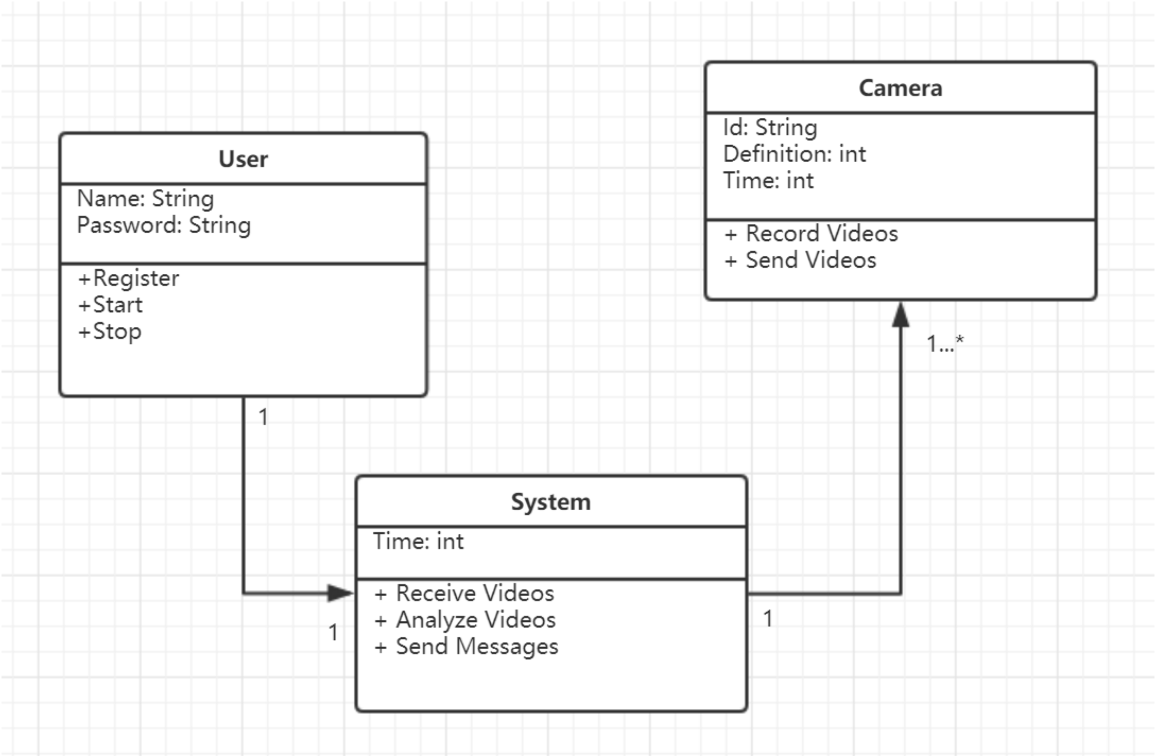


Figure 5: Class diagram.

3.9 State Diagram

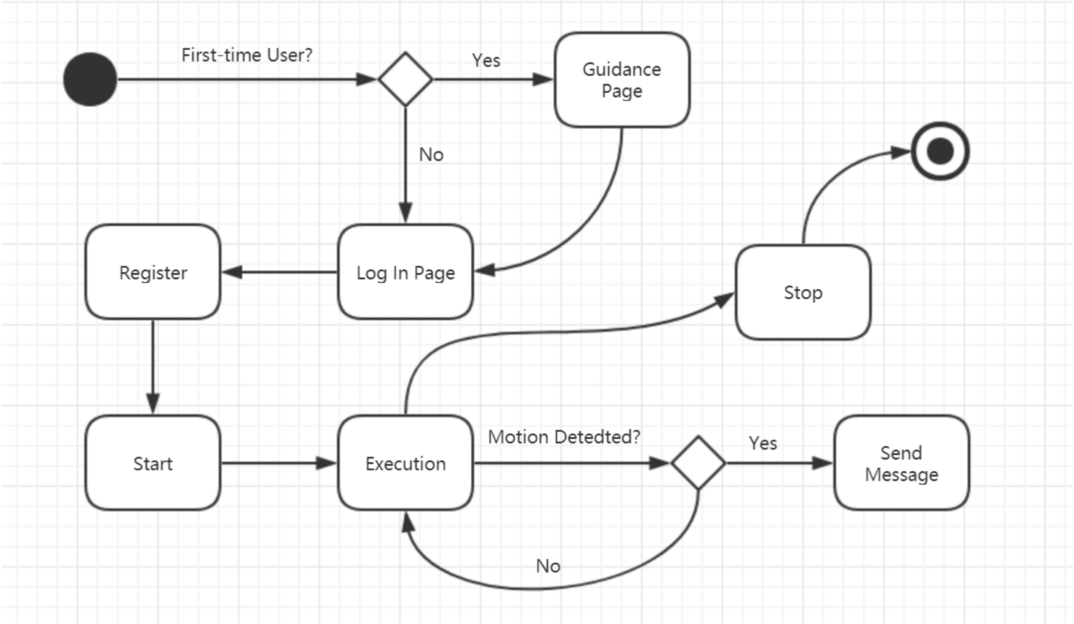


Figure 6: State diagram.

1. Change Management Process

After identifying the requirements received from the customer either by formally in writing, email, phone call or other ways, our team needs to specify the requirement definition. Then, more detailed implementation, design, coding, and testing shall be implemented. At last, acceptance testing is necessary to deploy the functions. Note that these steps are round-trip until acceptance testing is passed.

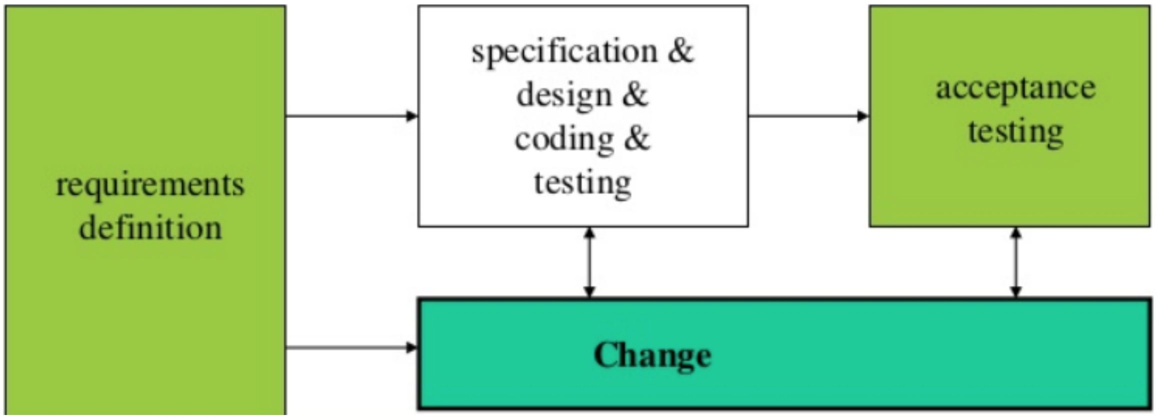


Figure 7. Change Management Process Flow Chart

Design Specification

1. Introduction

1.1 **Goals and Objectives**

"Automatic Alarm System with Motion Detection" is established to detect any unusual motion with the help of a camera and image processing technique and inform the user by sending a message in Gmail immediately if necessary.

Users can use it to monitor their house for 24 hours every day and secure their house. Customers can adjust the settings of all the cameras, such as positions, rotation, image quality, and on-off status. In the process of monitoring, customers will have to go through a few steps, such as login, select cameras, and video recording.

The administrator of "Automatic Alarm System with Motion Detection" is given tools for adding and removing functions to and from the system. The tools make the entire process of managing the system automated and do not place any restrictions on the knowledge needed to use this system, which make it user-friendly.

1.1 System Statement of Scope

The scope of the system is to allow users to come and monitor their house on our system. Automatic Alarm System with Motion Detection expects to have two kinds of user: an administrator and a customer.

CUSTOMER

From the customer point of view, there are the following inputs:

- User login elements (password, log in);
- User info;
- Cameras selection;

The functionality of the system on the customer side consists of:

- User authentication;
- Image processing;
- Searching for cameras;

The output for the customer has some elements, such as:

- Confirmation page;
- Image analysis result;
- Alarm message

ADMINISTRATOR

The inputs on the Administrator part are:

- Administrator log in;
- Administrator info;
- Take inputs;

-

Functionality for Administrator is the following:

- Adding and removing functions;
- Editing cameras functions;
- Editing system functions
- Creating reports;

The Administrator part has the outputs as follow:

- All kinds of reports;
- Functional changes;

1.2 Software Context

"Automatic Alarm System with Motion Detection" is developed to be a complete solution for the people trying to monitor and secure their house. Compared to most of the monitoring systems, "Automatic Alarm System with Motion Detection" provides image analyzing and sending messages functions for users. The administrator has some dynamic tools that allow him/her to add and remove functions from the system quickly.

1.3 Major Constraints

The major constraints are as following:

- Motion analysis accuracy

- Network condition

2 Data design

2.1 Internal Software Data Structure

Majority of data passed among the components will be simple data types like string or integer. During components communication, dataset structure might be used. This structure is the same as described in section 2.3 Temporary Data Structure.

2.2 Global Data Structure

The only global constructs are the data structures provided by ASP.NET, such as system settings and camera settings.

3 Architectural and Component-Level Design

3.1 System and Program Structure, Architecture Diagram Showing System Components.

3.1.1 Architecture Diagram

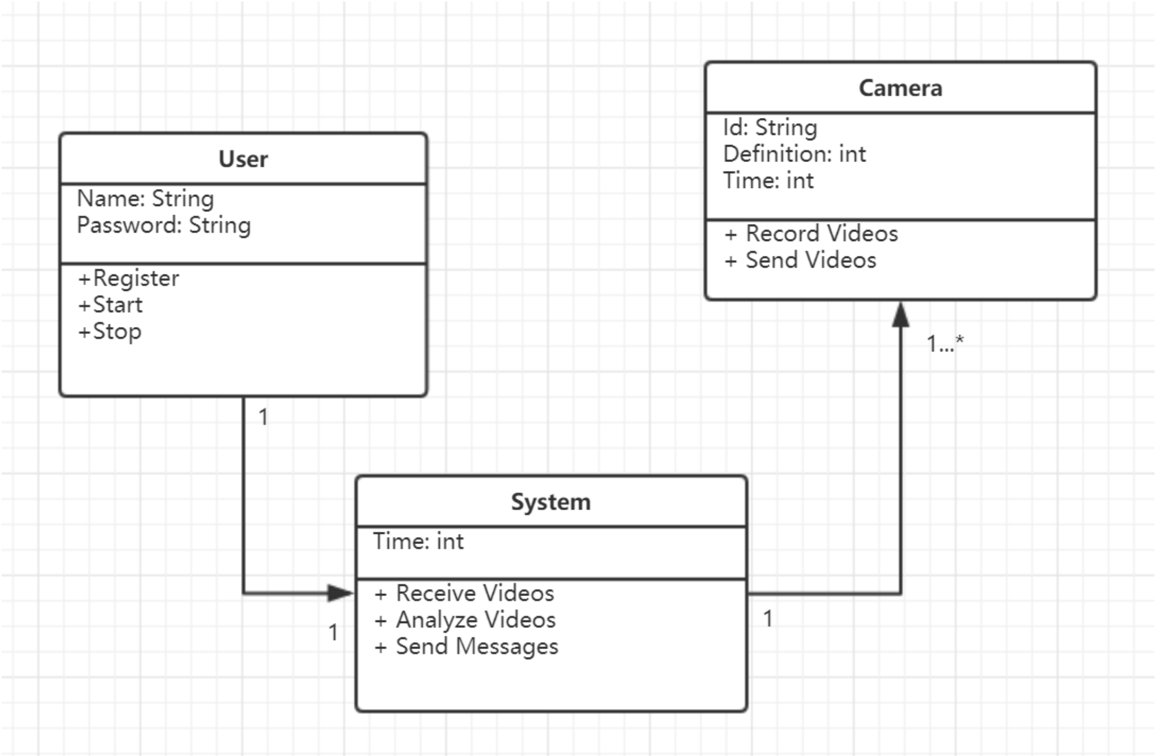


Figure 8: Top-Level Architecture diagram

3.1.2 Alternatives

Figure 8 shows the top-level architecture of the automatic alarm system, which contains three classes: the user class, the camera class, and the system class. Figure 1 directly shows the relationship between the user, the system, and the camera. The relationship between the user and the system is one to one, and the relationship between the system and the camera is one to more.

Figure 9 is the block diagram of input, processor, and output, which shows the way that hardware works on the processor, and the way that the processor works on mobile software.

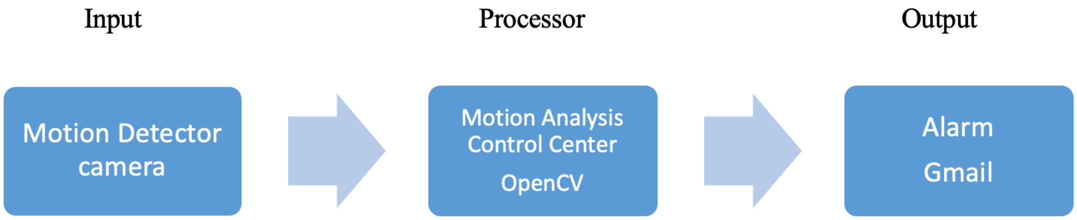


Figure 9: Block diagram of input, processor, and output.

3.2 Component Descriptions

3.2.1 Component 1: Motion Detector (Camera)

3.2.1.1 Component Narrative

The camera is the motion detector, which can be a computer camera or the external camera connected to the computer. The camera records the video in real time and uploads the video to the motion analysis control center.

3.2.1.2 Interface Description

This system uses a computer camera since most computers are equipped with cameras. Though the external camera may have a better view than the computer camera, they have almost the same capability, and the computer camera is easier to be controlled.

The camera is connected to the motion analysis control center. It gets the real-time video as input and output the pixel value for the motion analysis control center to analysis.

3.2.2 Component 2: Motion Analysis Control Center

3.2.2.1 Component Narrative

The motion analysis control center uses OpenCV to achieve motion detection. OpenCV is the Open Source Computer Vision Library, it focuses on real-time applications and is used in this system to detect real-time motion.

3.2.2.2 Interface Description

OpenCV provides many algorithms for separating the dynamic foreground from the static background to detect motion. This system uses GMM's cv2.ackgroundSubtractorMOG(). The motion analysis control center takes the video recorded by the camera as the input, and analyzes each new image's pixel value, uses GMM to assess the pixel value, separates the foreground and background, and output the motion detection result through the Gmail.

3.2.3 Component 3: Alarm (Gmail message)

3.2.3.1 Component Narrative

The alarm is a Gmail message and can be self-defined. For example, "Someone is in your home."

3.2.3.2 Interface Description

Python has an API called smtplib, which can be used to invoke Gmail. Before the motion analysis control center connecting to the Gmail, the user needs to log in Gmail by entering email address and password. If the motion analysis control center detects the motion, smtplib API is called, and a message will be sent to the user's Gmail.

3.3 Interface Descriptions

3.3.1 External Machine Interfaces

This system uses computer camera as the machine interface; it is internal and can be controlled by the motion analysis control center easily. If the user wants to use an external camera, the system does not require a complex machine interface, because most external cameras use the USB port and don't need to install the driver.

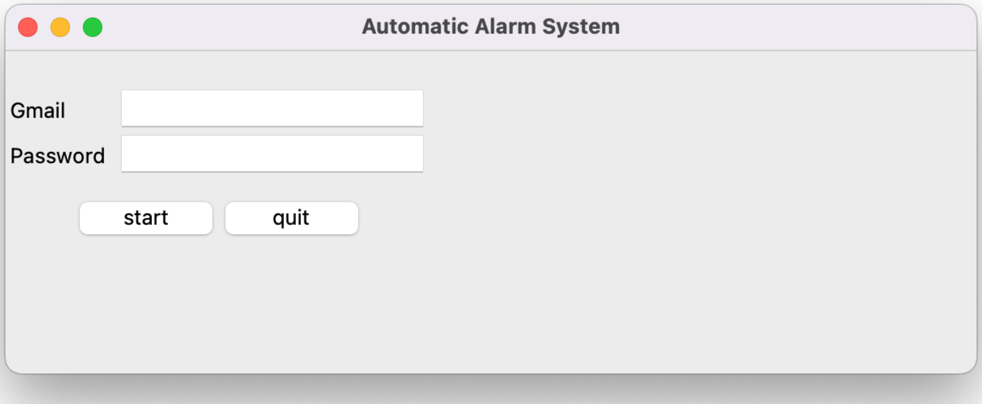
3.3.2 External System Interfaces

The system interface of this system can be any Python IDE that can run python, invoke OpenCV and import smtplib.

4 User interface design

4.1 Description of The User Interface

4.1.1 Screen Images


Automatic Alarm System

Gmail

Password

Figure 10: Main window.

4.1.2 Objects and Actions

Figure 10 shows the main window of the Automatic Alarm System. It has two labels: Gmail and Password; two entries for the user to enter the email address and password, two selection bars for the user to start and quit the system.

4.2 GUI Components

GUI component	description

Frame	frame control; displays a rectangular area on the screen, mostly used as a container
Canvas	canvas control; display graphic elements such as lines or text
Entry	input control; used to display simple text content
Button	button control; display the button in the program
Label	Label control; used to display text content

Table 4: GUI components

4.3 UIDS Description

Python provides a library of multiple graphical development interfaces. Several common Python GUI libraries are as follows:

Tkinter: The Tkinter module (Tk interface) is the interface to Python's standard Tk GUI toolkit. Tk and Tkinter can be used on most Unix platforms, as well as Windows and Macintosh systems. Subsequent versions of Tk8.0 can implement native window styles and work well on most platforms.

wxPython: wxPython is an open source software, an excellent GUI graphics library for the Python language that allows Python programmers to create complete, full-featured GUI user interfaces easily.

Jython: Jython programs integrate seamlessly with Java. In addition to some standard modules, Jython uses Java's modules. Jython has almost all the modules in the standard Python that don't depend on the C language. For example, Jython's user interface will use

Swing, AWT or SWT. Jython can be compiled dynamically or statically into Java bytecode.

5 Restrictions, Limitations, and Constraints

The accuracy of motion analysis may constrain the system, and the system may send messages mistakenly because of users' normal motions and disturb the users.

The system is also likely to be constrained by the network situation; it may not send the alarm information in time if the network connection condition is terrible.

6. Requirements Traceability Matrix

Project	Requirements List	Name	Description	Status	Notes
Automatic Alarm System with Motion Detection	1	Login	Log in to the system; Identify invalid name or password	N/A	
	2	Register	Register an account of the system	N/A	

	3	Video Recording	Record the videos	N/A	
	4	Video Uploading	Upload the video to be analyzed	N/A	
	5	Image Analysis	Analysis of the motion of the images	N/A	
	6	Send Message	Send alarm message if necessary	N/A	
	7	User-defined Message	The message should be self-defined	N/A	

Table 5: Requirements Traceability Matrix

Prototyping

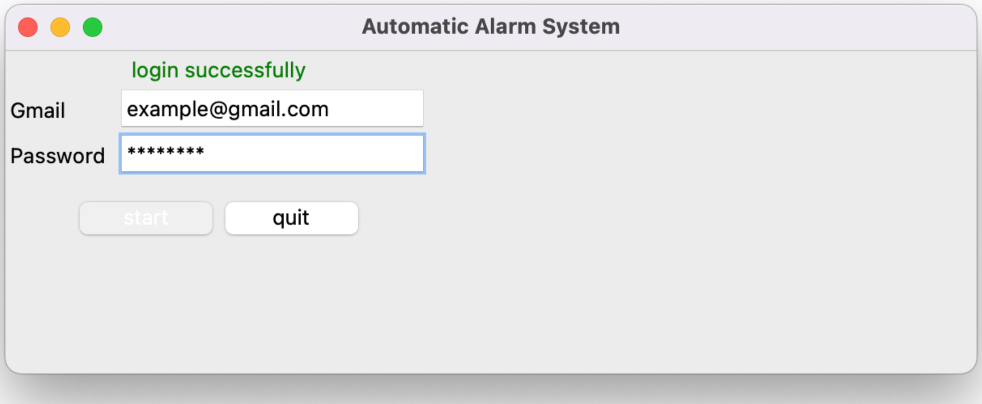
1. User Interface

1.1 User Start

Input: correct Gmail and password

Action: Click “start” button

Output: “login successfully,” password is hidden, the camera starts work.



login successfully

Gmail example@gmail.com

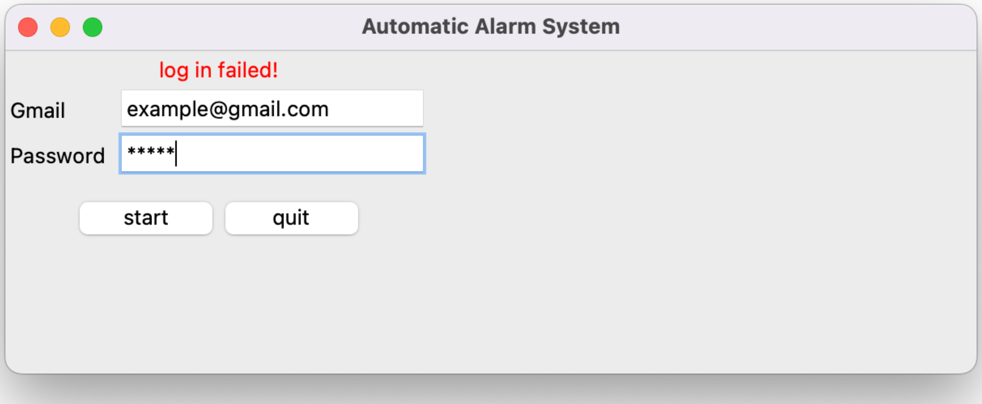
Password ********

Figure 11: User starts successfully.

Input: Wrong name or password

Action: Click “start” button

Output: “login failed,” password is hidden, the camera doesn’t work.



Automatic Alarm System

log in failed!

Gmail example@gmail.com

Password *****

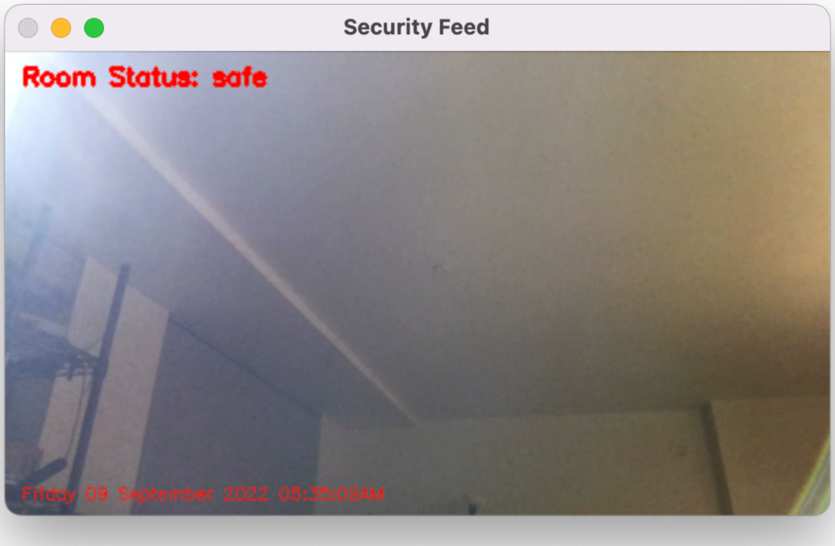
Figure 12: User starts failed.

1.2 Motion Detection

Input: current frame captured by the camera

Action: No movement

Output: Room Status: safe; No alarm message.



Security Feed

Room Status: safe

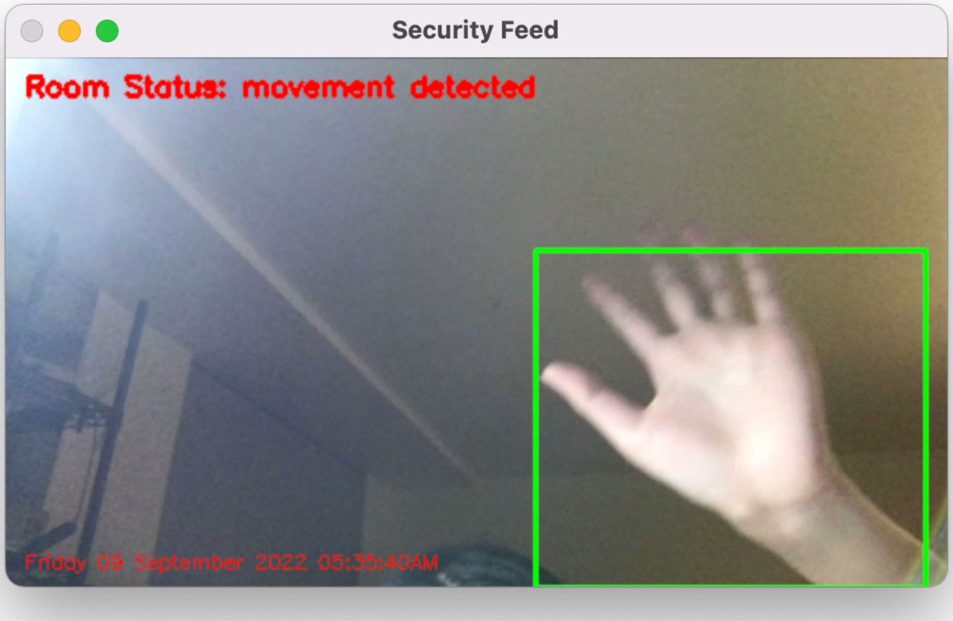
Friday 09 September 2022 05:35:09AM

Figure 13: Main frame: safe.

Input: current frame captured by the camera

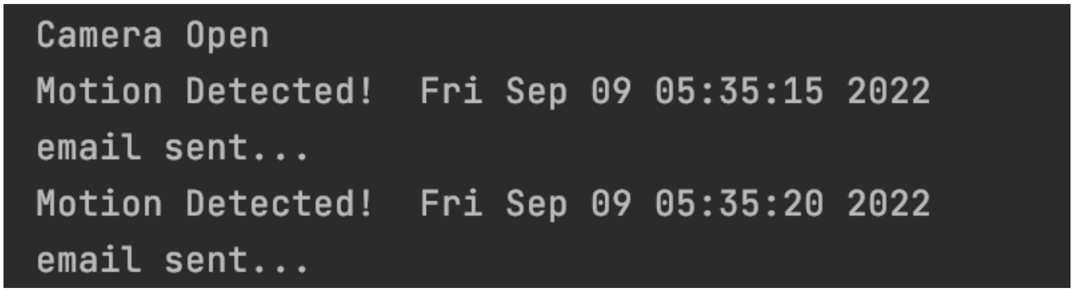
Action: Wave hand

Output: Room Status: movement detected; Alarm message is sent to Gmail.



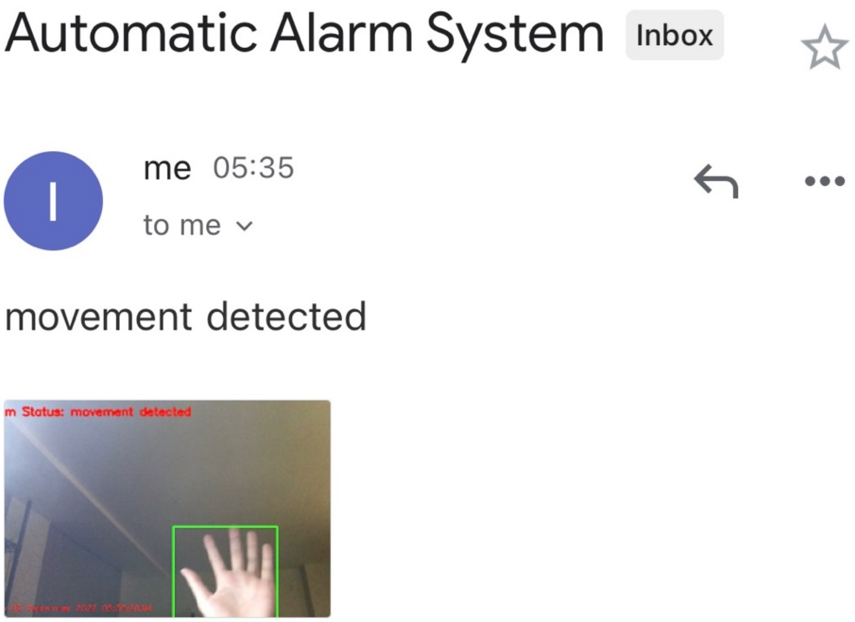
Security Feed
Room Status: movement detected
Friday 09 September 2022 05:35:40AM

Figure 14: Main frame: movement detected.



```
Camera Open
Motion Detected!  Fri Sep 09 05:35:15 2022
email sent...
Motion Detected!  Fri Sep 09 05:35:20 2022
email sent...
```

Figure 15: Output: Motion Detected.



Automatic Alarm System

me 05:35

to me

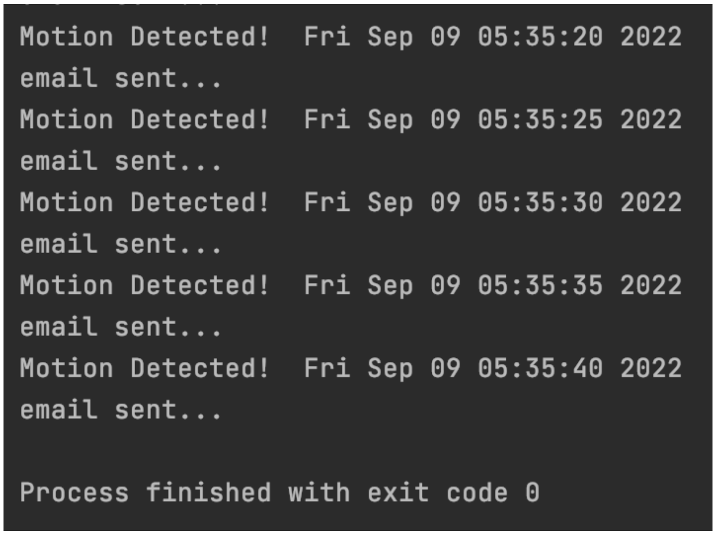
movement detected

Figure 16: Gmail.

1.3 User quit

Action: Click the “quit” button

Output: all windows closed, program finished.



```
Motion Detected!  Fri Sep 09 05:35:20 2022
email sent...
Motion Detected!  Fri Sep 09 05:35:25 2022
email sent...
Motion Detected!  Fri Sep 09 05:35:30 2022
email sent...
Motion Detected!  Fri Sep 09 05:35:35 2022
email sent...
Motion Detected!  Fri Sep 09 05:35:40 2022
email sent...

Process finished with exit code 0
```

Figure 17: Program finished.

System Evaluation

This evaluation form is given by Han Jie.

(scoring: 1: “very poor”; 2: “poor, but acceptable”; 3: “fine, but not impressive”; 4: “well-done”; 5: “excellent work”)

Criteria	Consideration	Score
Completeness/Requirement	Has the software satisfied all important requirement? Does it function properly according to your request?	5
Quality/Security	Are there any unresolved software bugs? Are there any security problems such as unencrypted sensitive data?	5
User Interactions	Does the GUI (if any) work smoothly? Is it interesting/attractive? Is the setup/installation of the system troublesome?	4
Code (maintenance/clarity)	Is the code written well to follow common practice? Does the name of variables/classes/methods make sense? Did the developers put appropriate comments inside the code?	4
Extensible (versioning)	Can other developers easily extend/reuse partial of the code in the future? Can it be migrated to other systems? What if we want to change the database or other components in the future?	4

Table 6: System evaluation

Han Jie’s comments:

Their project is useful and easy to know what is needed to implement and they did a good job. Their code is not hard to understand. However, if they add some comments will be better. The system has lots of interactions with the users that even can make you feel annoying. The GUI is concise but enough. It satisfies the users. The future work is mainly about beautify the system and fix some small problems.

Self-evaluation:

This project satisfies all important requirements: rlogin, start motion detection, send alarm message, and captured frame to Gmail and quit. All the user information will be saved to a local file, which is safe and easy for retrieval. The GUI is simple and not attractive due to the limitation of Python's Tkinter. The system has clear and appropriate comments inside the code, and the class name and the variable name is well defined. This system can run in any python environment.

Discussion

The system detects motion five times / second, and send the captured frame and alarm message every five seconds. This time interval is tested for several times and identified to be the most appropriate for the system response time and the user experience. Compared to other projects using OpenCV to detect motion, this project combines OpenCV and smtplib library and allows the system to send alarm messages to the user's Gmail in real time. Gmail has 1.8 million users by 2022. Since Gmail has so many users and more and more people concern about residence security, this automatic alarm system has a good foreground. In this project, I use a PC camera, which is not convenient to monitor a big area, and it's better to use an external camera with small size and high pixel.

The GUI is simple and easy for users to operate. The user information is saved in a local file, so the register and login part seems not necessary, but for security consideration and future development of the database, the registration and login are required.

Future Work

This system needs an external camera for monitoring a larger area that satisfies the protection of residence and office security. The motion detection technology will be developed more accurately and sensitively. This system needs a database to store the user information, and the captured frame of the potential risk, so the data can be organized for retrieval. Design a more standard and beautiful user interface.

Reference:

Baksheev, A., et al. "Realtime computer vision with opencv." (2012).

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

Rosebrock, Adrian. “Basic motion detection and tracking with Python and OpenCV.” (2015).

Rui, Chen & Zhuohui, Li. “Senior Project proposal.” (2019).

Xudong Lou & Yuwei, Zhou. “Senior Project System Requirement Specification.” (2019).