Albergo supervision

Final Year Project

Session 2019-2023

A project submitted in partial fulfillment of the degree of

BS in Information Technology

Department of Information Technology

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2023

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R**&**D**
Area of specialization	
FYP ID	FYP-BITM-F22-011

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BITM-F19-063	Adil Tahir	bitm-f19-063@superior.edu.pk	
(ii)	BITM-F19-062	Abdul Moeed	bitm-f19-062@superior.edu.pk	
(iii)	BITM-F19-074	Awais Saleem	bitm-f19-074@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others.

Plagiarism Free Certificate

This is to certify that, I _____ S/D of Muhammad Idrees, group leader of FYP under registration no _____at Information Technology Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: Name of Group Leader: _____ Signature: _____

Supervisor: Mr. Naseer Ahmad
Designation: Lecturer

Signature: _____

Co-Supervisor: Mr. Fahad Sabha
Designation: Lecturer

Signature: _____

HoD: Dr. Asad Naqvi

Signature: _____

Dedication

We dedicate this project to Allah Almighty who gave us the strength to do this software development project and to our advisor (Prof.Naseer Ahmad) who gave this idea to do this project. He has been a source of strength throughout this journey of the Last Year project. With his help we decided to do this project and divided the whole work into three different parts after 6 to 7 months of consecutive work, thanks to Allah almighty we did it. Last but not the least (Abdul Moeed) I appreciate my group members (Adil Tahir Awais Saleem) who helped me in this project with full support and courage.

Acknowledgements

I am thankful to my supervisor whose valuable guidance and kind supervision he provided us during the final year project which shaped this work as its show. Finally, our parents are also an important inspiration for us, so we thank them with respect and reverence.

Executive Summary

Albergo supervision basically is uniform detect system. We use artificial intelligence to make this model. Uniform detection AI is a technology that can identify various uniform designs in images and videos. It is helpful to identify those who are dressed uniforms, such as police officers, security guards, and others. This innovation analyses photos and identifies the type of uniform being worn using computer vision and machine learning. To make this work, data on various uniform types is gathered and used to train the AI system. Once taught, the AI can quickly identify uniforms in real-time and make judgements based on what it observes. Accurately identifying people who are sporting particular uniforms is the aim of uniform identification AI.

Chapter 1
Introduction

Chapter 1: Introduction

- **Image Acquisition**

  The image is collected using a digital camera from several locations in this method.

- **Image Preprocessing**

  We use a preprocessing technique to remove the noise and obtain an informative Image.

- **Image Segmentation**

  The method of dividing a picture into distinct parts is known as image segmentation. It is used to elicit some sensitive into from a segmented image.

- **Feature Extraction**

  We will be working with RCNN model used to classify the image for the dress and state-of-the-art object.

- **Dress Code Violation**

  Finally, by applying all these processes, the dress image will be used to detect whether the staff person has followed the dress code. If not, then an alert message will inform this violation.

1.1. Background

We will use the process of image segmentation and pose estimation to segment the image into super pixels and then analyze accordingly. A CCTV camera or comparable technology will be used to record images of clothes to detect that person's attire. The dress image will be used to detect whether the staff person has followed the dress code if not then an alert message will inform this violation.

1.2. Motivations and Challenges

- Working in Machine Learning
- Integration between Desktop and Web Application

1.3. Goals and Objectives

2. Good graphics
3. Staff uniform improvement.
4. To provide quality user experience.
5. To provide knowledge about the product.

1.4. Existing Solutions

Paper	Year	Dataset	Total classes	Total images	Model	Accuracy	Segmentation/Classification	Conference/Journal
Clothing identification via deep learning: forensic applications	2018	Deep Fashion logo dataset	5412 categories	278461	NVIDIA's Deep Learning, CNN	75%	Classification	Journal
Classification Of Dress Codes Using Convolution Neural Networks	2020	Collected from random web pages and surveillance cameras	NULL	270	CNN	Shirt=81.5% Dress Code=84.2%	Classification	Conference

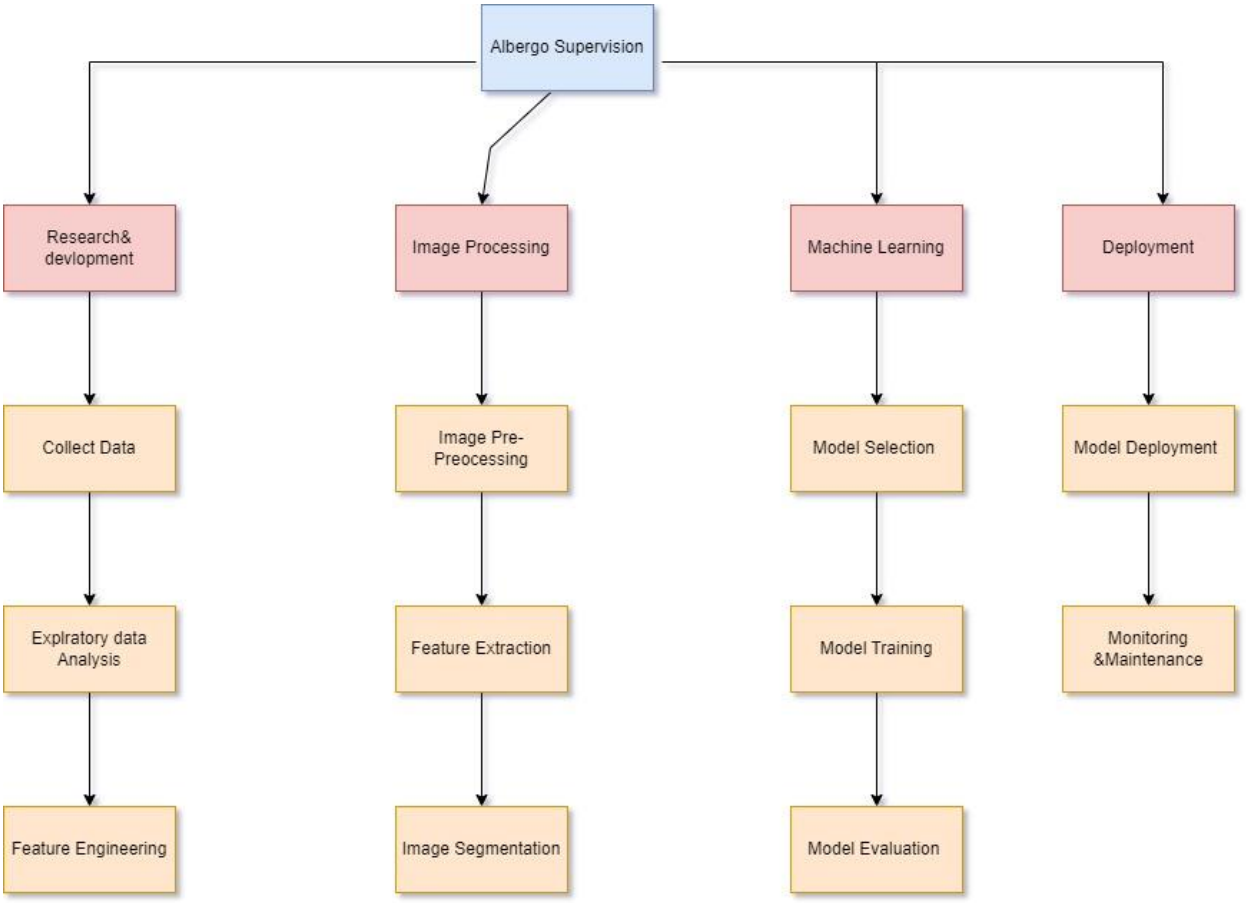
1.5. Gap Analysis

The gab analysis of our project is:
Accuracy: Compare the system's ability to accurately identify people wearing the target uniform to the desired level of accuracy.
Speed: Measure the time it takes for the system to process an image and identify the uniform and compare it to the desired processing speed.
False positives and false negatives: Evaluate the rate of false positives (identifying a non-target uniform as a target uniform) and false negatives (failing to identify a target uniform) and compare it to the desired rates.
Interoperability: Evaluate the system's ability to integrate with other systems, such as facial recognition or tracking systems and compare it with the desired levels of interoperability.

1.6. Proposed Solution

The proposed solution for a uniform detector through Artificial Intelligence is to use an AI-based facial recognition system. This system would be able to detect people wearing uniforms and differentiate between authorized personnel and people who are not wearing uniforms. It could be used to monitor the entrances and exits of buildings, or to keep track of who is on duty in a particular area. The system would use a database of images to match against the faces of people entering the premises. It could be supplemented with other features such as voice recognition or biometrics to further increase accuracy. Additionally, the system could be used to alert security personnel in case of any suspicious activity.

1.7.1. Work Breakdown Structure



1.7.2. Roles & Responsibility Matrix

Task	Assigning	Starting date	Ending Date	Duration
Phase1 (Starting)				
Meeting	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	27-10-2022	30-10-2022	3
Research on Project	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	2-11-2022	11-11-2022	11
Requirements	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	15-12-2022	20-12-2022	5
SRS 1	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	18-11-2022	25-11-2022	7
Phase 2(Planning)				
Project proposal	Muhammad Adil, A.Moeed,	13-12-2022	21-12-2022	8

	Muhammad Awais Mughal			
SRS 2	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	23-12-2022	1-01-2023	6

Phase 3(Execution)

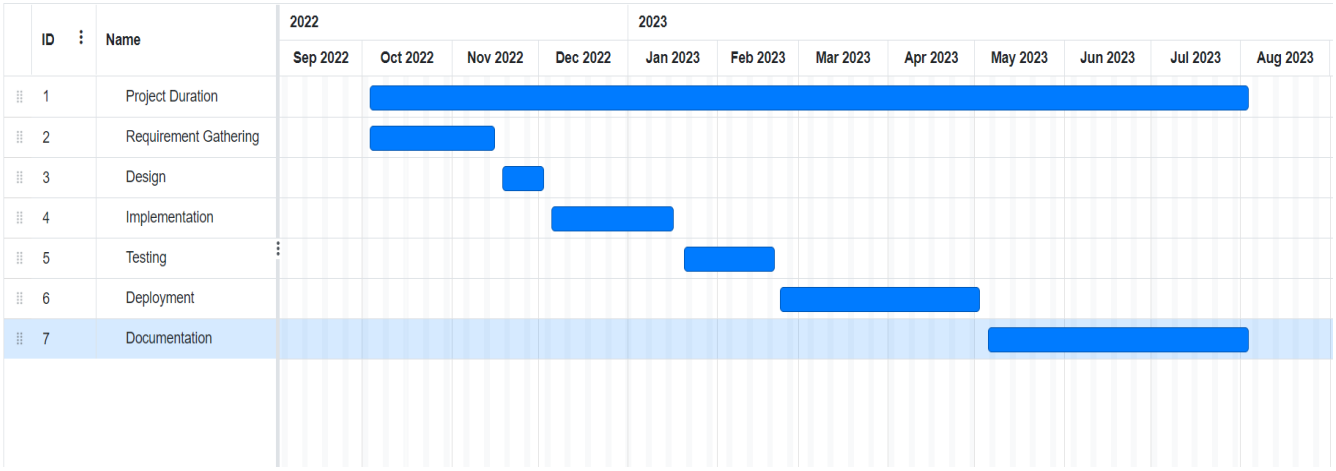
DataSet	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	5-03-2022	13-03-2023	32
RCNN	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	14-03-2022	20-03-2023	6
Image segmentation	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	22-03-2022	10-04-2023	18
Coding	Muhammad Adil, A.Moeed,	12-04-2022	30-4-2023	18

	Muhammad Awais Mughal			
SRS 3	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	1-5-2022	10-5-2023	10

Phase 4 (testing and concluding)

linking	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	12-05-2022	25-05-2023	17
testing	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	28-05-2022	15-06-2023	16
Closing of project	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	18-06-2022	20-06-2023	7
SRS 4	Muhammad Adil, A.Moeed, Muhammad Awais Mughal	22-06-2022	29-06-2023	4

1.7.3. Gantt Chart



Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.7. Introduction

2.7.1. Purpose

The purpose of our project to identify and classify objects that have a uniform appearance or texture within an image or video. This can be used in a variety of applications, such as image segmentation, object recognition, and surveillance. For example, uniform detection can be used to identify and track individuals wearing the same type of clothing or to segment an image of a field of crops into regions of uniform color.

2.7.2. Document Conventions

Our Project refers to the standard format and layout that should be followed when creating documents or images for use in uniform detection systems. This can include guidelines for image resolution, file format, and annotation conventions. The specific conventions may vary depending on the uniform detection system being used, but they are generally designed to ensure that the system can accurately and efficiently detect and recognize uniforms in the images. Some common conventions include:

- **Image resolution:** The images should be high resolution and taken in good lighting to ensure that the uniforms are clearly visible.
- **File format:** The images should be saved in a format that is compatible with the uniform detection system, such as JPEG or PNG.
- **Annotation conventions:** The images should be annotated with labels indicating the type of uniform present in the image, such as "police uniform" or "military uniform". The annotation should be in a format that the system can understand, such as XML or JSON.

2.7.3. Product Scope

The product scope of our Project typically includes the ability to identify and classify uniforms worn by individuals in an image or video stream. This may include identifying the specific uniform (police, military, security, etc.) as well as any insignia or markings on the uniform. The technology may be used in a variety of applications such as security and surveillance, crowd management, and identifying authorized personnel in restricted areas.

2.8. Overall Description

2.8.1. Product Perspective

Albergo System is a useful tool in a variety of settings, including security, retail, and manufacturing. For example, in security settings, uniform detection can be used to identify authorized personnel and ensure that only authorized individuals have access to restricted areas. In retail, uniform detection can be used to track employee performance and ensure that employees are following company dress code policies. In manufacturing, uniform detection can be used to monitor compliance with safety regulations and ensure that employees are wearing the proper protective gear. Overall, the product perspective for uniform detection is that it can be a valuable tool for organizations looking to improve security, efficiency, and compliance.

2.8.2. Operating Environment

An operating environment for uniform detection refers to the specific conditions and settings in which a uniform detection system is designed to function. This can include factors such as the types of uniforms being detected, the lighting conditions, the distance at which the system is able to detect uniforms, and any other relevant factors that can affect the performance of the system. For a uniform detection system to be effective, it must be optimized for the specific operating environment in which it will be used.

2.8.3. Design and Implementation Constraints

The User interference design must be simple and attractive.
It is a User-Friendly web base interface.
Code must be clean and readable.

To Run this web must use:

- *Camera system will install.*
- *Latest python version*
- Operating System**:** Windows

- *CPU: core i7 6th gen*
- *Ram: 8gb*
- *Storage: 20Gb*
- *Gpu: 2Gb Graphic Card*

2.9. External Interface Requirements

2.9.1. User Interfaces

In User interface the computer vision applications to detect and identify individuals who are wearing uniforms. The interface typically includes an input section for capturing images or video, such as a camera feed, and an output section for displaying the results of the detection, such as highlighting the regions of the image where a uniform is present or displaying the name or identification number of the person wearing the uniform. Some uniform detection interfaces also include options for adjusting the detection parameters, such as the types of uniforms to look for or the threshold for determining a positive detection.

User interfaces include:

Login Page

Logout Page

Warning/Alert Page

2.9.2. Hardware Interfaces

In Hardware interface Cameras can be used to capture images of individuals, while sensors can be used to detect the presence of an individual and collect additional data such as body temperature. Image processing software is then used to analyze the images and sensor data, to identify the uniform worn by the individual.

Additionally, the hardware can include a processing unit and a communication interface, to receive the processed image and send the result to a control unit or to a higher-level software.

Uniform detection can be done by multiple methods such as:

- Machine learning
- Pattern recognition

- Image processing
- Computer vision

2.9.3. Software Interfaces

We make an AI-based Project for uniform detection that uses computer vision techniques to analyze digital images or video streams and detect the presence of uniforms in them. The specific functionality of the interface will depend on the specific application and the type of uniform being detected. A software interface for uniform detection will include a set of algorithms and parameters that can be adjusted to improve the performance of the system.

2.9.4. Communications Interfaces

In Communication Interface for uniform detection, we refer to a system or device that facilitates communication between different devices or systems to detect uniforms worn by individuals. This system or device could use various technologies such as image recognition, RFID, or NFC to detect and identify uniforms worn by individuals, and then communicate that information to other systems or devices for further processing or analysis.

2.10. System Features

2.10.1. Registration

2.10.1.1. Description and Priority

Before using the system, users must first create an account, and admins have their own access to the system too.

2.10.1.2. Functional Requirements

Image Acquisition: The first step is to capture a picture or a video of the uniform. Several tools, including cameras, mobile devices, and other image-capture equipment, can be used for this.

Image preprocessing: To make it simpler for the algorithm to recognize patterns, the image is then treated to remove any noise, shrink its size, and improve its features.

Feature Extraction: The next step is to extract important features from the image, such color, texture, and shape, to assist the algorithm in finding patterns.

Pattern Recognition: The system works algorithms using the collected features to find any similar patterns in the image. To identify a match, this procedure compares the image's properties to a database of well-known uniform patterns.

Classification: Finally, the algorithm classifies the discovered uniform pattern and returns the relevant information, including the kind of uniform, the company or organization.

Output: Based on the uniform pattern identified, the algorithm's output is then utilized in various applications, such security monitoring, retail and fashion, and brand protection.

2.10.2. System Feature 2

Logout

Identifier	Admin Logout	
Purpose	logout from Account	
Priority	High	
Pre-conditions	Admin Must be login	
Post-conditions	Session is destroyed	
Typical Course of Action		
S#	***Actor Action***	***System Response***
1	User clicks on the menu	Different options appear
2	User chooses logout option from the menu	Session destroyed and after appear login page
Alternate Course of Action		
S#	***Actor Action***	***System Response***

2.10.3. System Feature 3

Dress Code Checking

Identifier	*Administrator Checking*
Purpose	*To check and balance over the system results and response to the security regulators.*
Priority	*High*
Pre-conditions	*Software Running with integrated cameras.*

Post-conditions	*Detected people data will be loaded to a file.*	
Typical Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin wants to see the people recognized by the system Staff Dresses*	*System will show the captured and processed results of Stuff*
2		
Alternate Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin wants to update the log list generated by the system of perpetrators*	*System will update the log list for more results concluded in real time video feed*
2	*Admin wants to delete any faulty or mistaken result generated by the system*	*System will delete the specific wrong result miscalculated by the recognition feature.*

2.11. Nonfunctional Requirements

2.11.1. Performance Requirements

In Our Project the Uniform detection refers to the ability of a computer vision system to accurately identify and classify the uniforms worn by individuals in an image or video. The performance requirements for a uniform detection system will depend on the specific use case and the level of accuracy required.

In general, a high level of accuracy is required for uniform detection in security or surveillance applications, as false positives or false negatives could have serious consequences. In these cases, a high true positive rate and low false positive rate are desired. The system must also be able to handle variations in lighting, occlusion, and camera angle.

2.11.2. Safety Requirements

There are several safety requirements that must be considered when implementing uniform detection:

Data privacy: Uniform detection systems should be designed to protect the privacy of individuals by minimizing the collection and use of personal information and ensuring that any personal information that is collected is properly secured and only used for authorized purposes.

Accuracy and bias: Uniform detection systems should be designed to minimize errors and biases, and to provide accurate results even when detecting individuals from diverse backgrounds. This

includes using appropriate training data and testing the system against a diverse set of images to ensure it works well for all individuals.

Transparency: The system should be transparent in terms of what it is detecting and why it is making certain decisions. The company should also provide a way for people to request or understand the data stored on them.

Auditing: The company should have a plan for auditing and monitoring the system for errors, biases, and other issues, and for taking corrective action as needed.

Compliance: The company should ensure that the system complies with all relevant laws, regulations, and industry standards regarding data protection, privacy, and security.

2.11.3. Security Requirements

Uniform detection, also known as person re-identification, is a computer vision task that involves identifying a person based on their clothing or uniform. Some security requirements for uniform detection systems may include:

High accuracy: The system should be able to accurately identify individuals even if they are wearing similar uniforms.

Scalability: The system should be able to handle large numbers of individuals and handle varying levels of crowd density.

Robustness: The system should be able to handle variations in lighting and camera angles.

Privacy: The system should be designed to protect individuals' privacy, for example by blurring faces or not storing images.

Usability: The system should be easy to use and integrated into existing security systems.

Redundancy: The system should have a backup in case of failure or emergency.

2.11.4. Usability Requirements

Usability requirements for uniform detection systems can include factors such as the accuracy and reliability of the system, the ease of use for the operators, and the ability of the system to integrate with other systems or equipment. Other requirements may include the ability to operate in various lighting conditions, the ability to detect a wide range of uniform styles or colors, and the ability to function in different environments or temperatures. Additionally, the system should have robust security features to protect data and confidentiality.

2.12. Domain Requirements

The specific requirements for uniform detection will depend on:

A camera or imaging device: This is used to capture images or video of the individuals being scanned for uniforms.

Image processing software: This is used to analyze the images or video captured by the camera and detect any uniformed individuals.

A database of uniform patterns: This is used to compare the images captured by the camera with known uniform patterns to identify individuals in uniform.

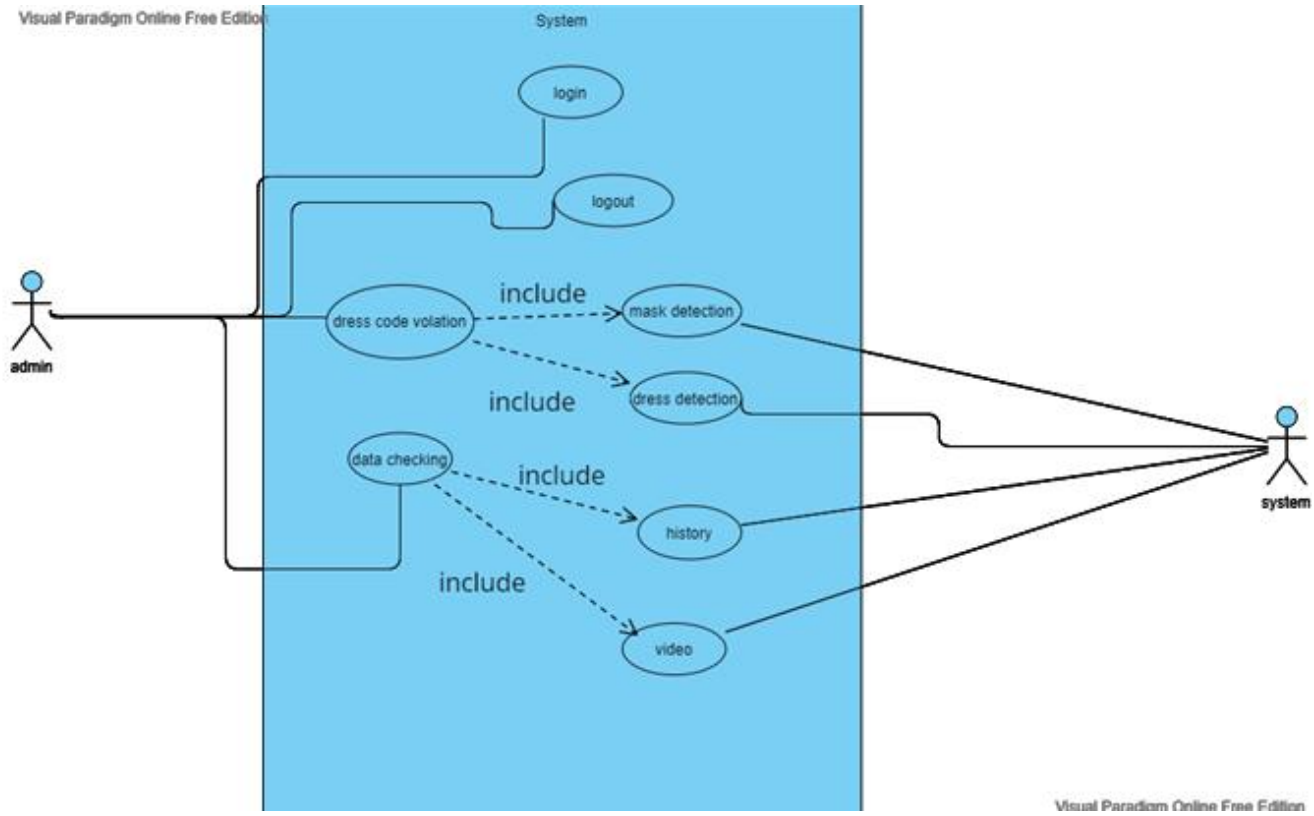
Hardware equipment: depends on the application and environment, will need related hardware for the system.

Chapter 3
Use Case Analysis

Chapter 3: Use Case Analysis

In this chapter, we create a use case model and create a description for every case to analysis the Albergo supervision.

3.1. Use Case Model



3.2. Use Cases Description

3.2.1Name of Use-Case 1: Login

Identifier	*Admin Login*
Purpose	*The access of admin to login to the system*
Priority	*High*
Pre-conditions	*Admin already have an account*

Post-conditions	*Admin can access portal or application and can update login password information.*

Typical Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin enters the username and password to login the system*	*Portal will be log in with the registered account.*
2	*Admin can change username and password for their dire need.*	*Database of the user account will be updated with the new username and password.*

Alternate Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin clicks on forget Password*	*A new password will be created by the admin and admin login again*

3.2.2 Name of Use-Case 2: Logout

Identifier	Admin Logout
Purpose	logout from Account
Priority	High
Pre-conditions	Admin Must be login
Post-conditions	Session is destroyed

Typical Course of Action		
S#	***Actor Action***	***System Response***
1	User clicks on the menu	Different options appear
2	User chooses logout option from the menu	Session destroyed and after appear login page

Alternate Course of Action		
S#	***Actor Action***	***System Response***
1	User does not choose logout option	Session remains

3.2.3Name of Use-Case 3: Dress Code Checking

Identifier	*Administrator Checking*

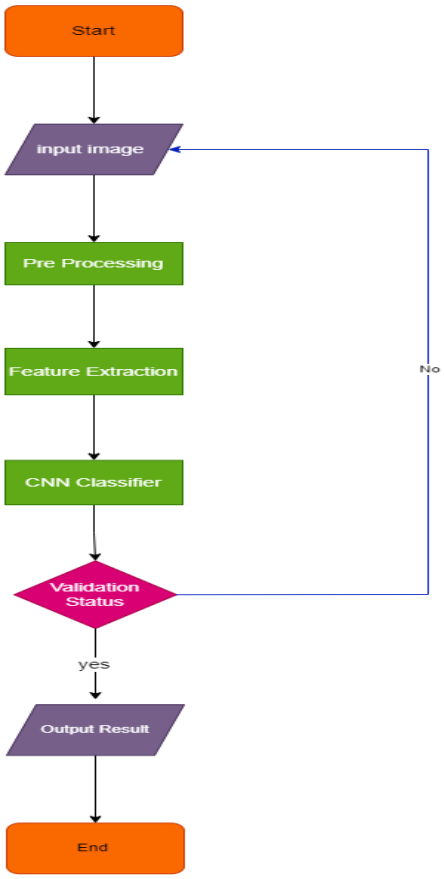
Purpose	*To check and balance over the system results and response to the security regulators.*	
Priority	*High*	
Pre-conditions	*Software Running with integrated cameras.*	
Post-conditions	*Detected people data will be loaded to a file.*	
Typical Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin wants to see the people recognized by the system Staff Dresses*	*System will show the captured and processed results of Stuff*
2		
Alternate Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin wants to update the log list generated by the system of perpetrators*	*System will update the log list for more results concluded in real time video feed*
2	*Admin wants to delete any faulty or mistaken result generated by the system*	*System will delete the specific wrong result miscalculated by the recognition feature.*

Chapter 4

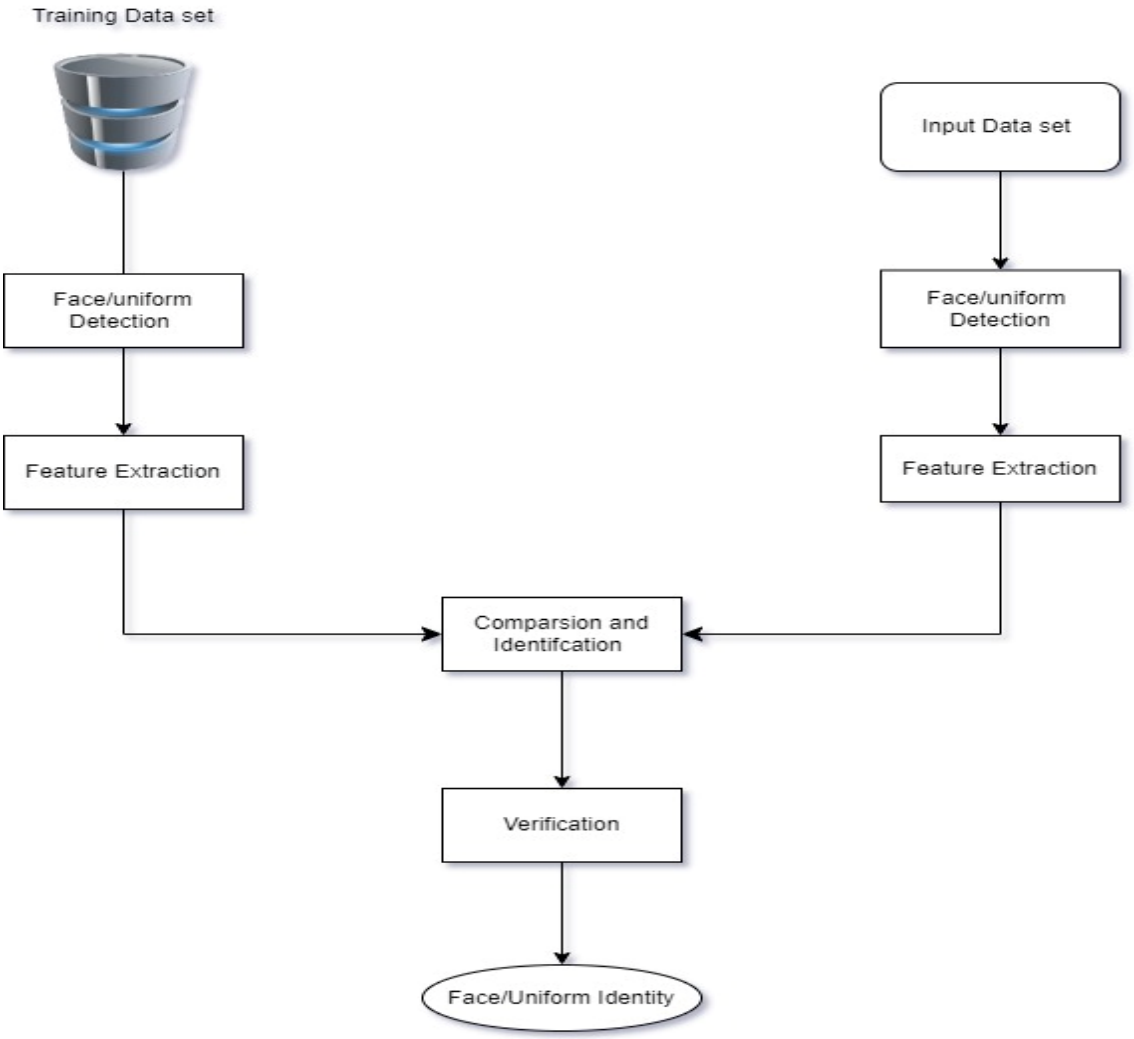
System Design

Chapter 4:

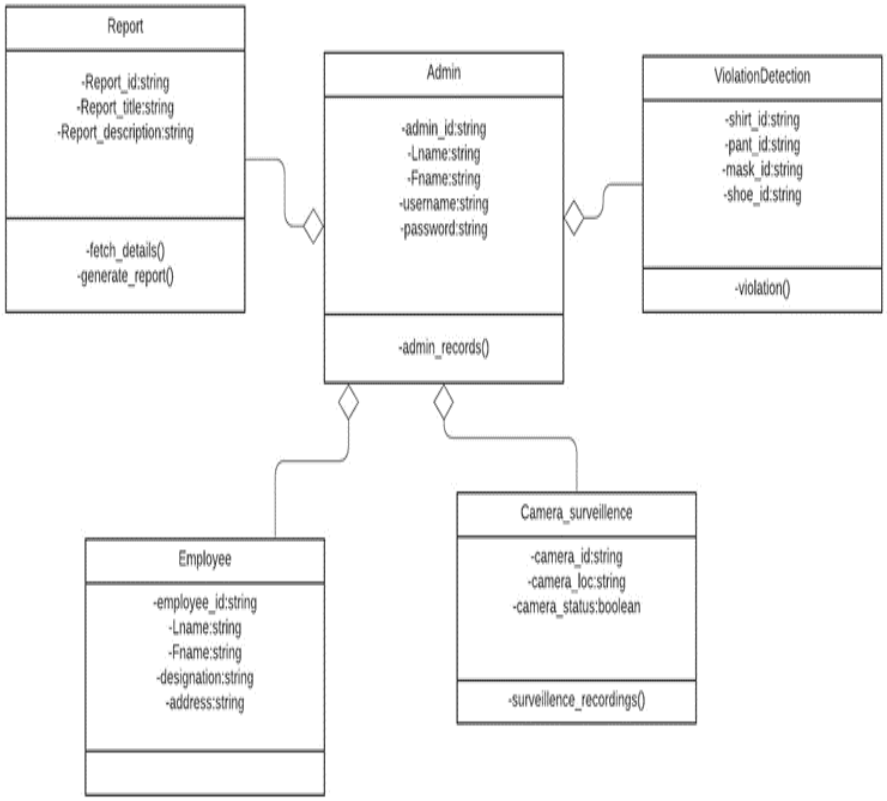
System Design



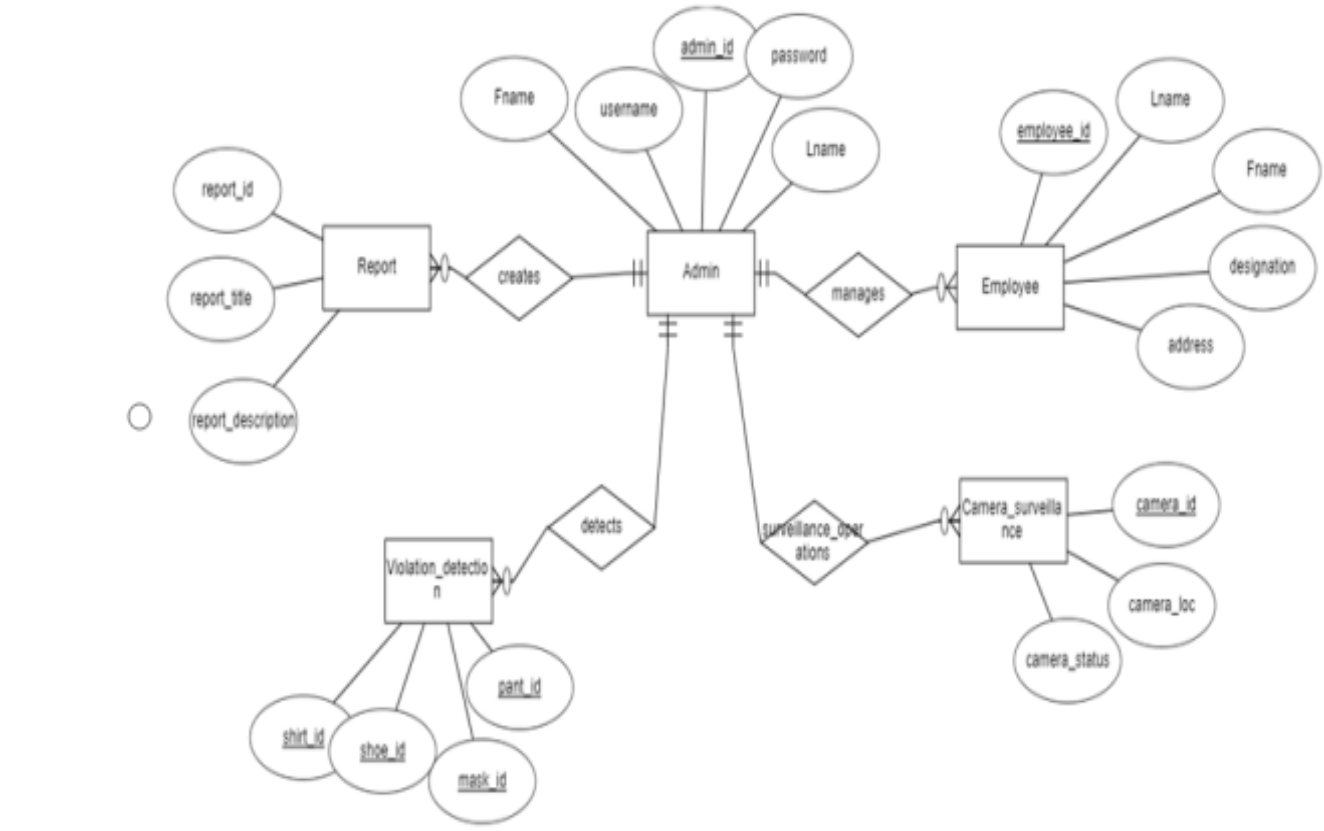
4.1. Architecture Diagram



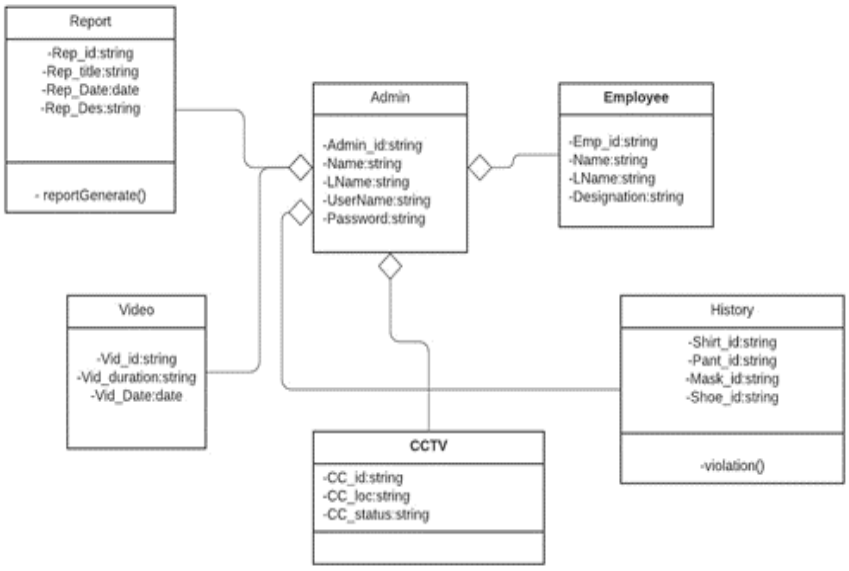
4.2. Domain Model



4.3. Entity Relationship Diagram with data dictionary

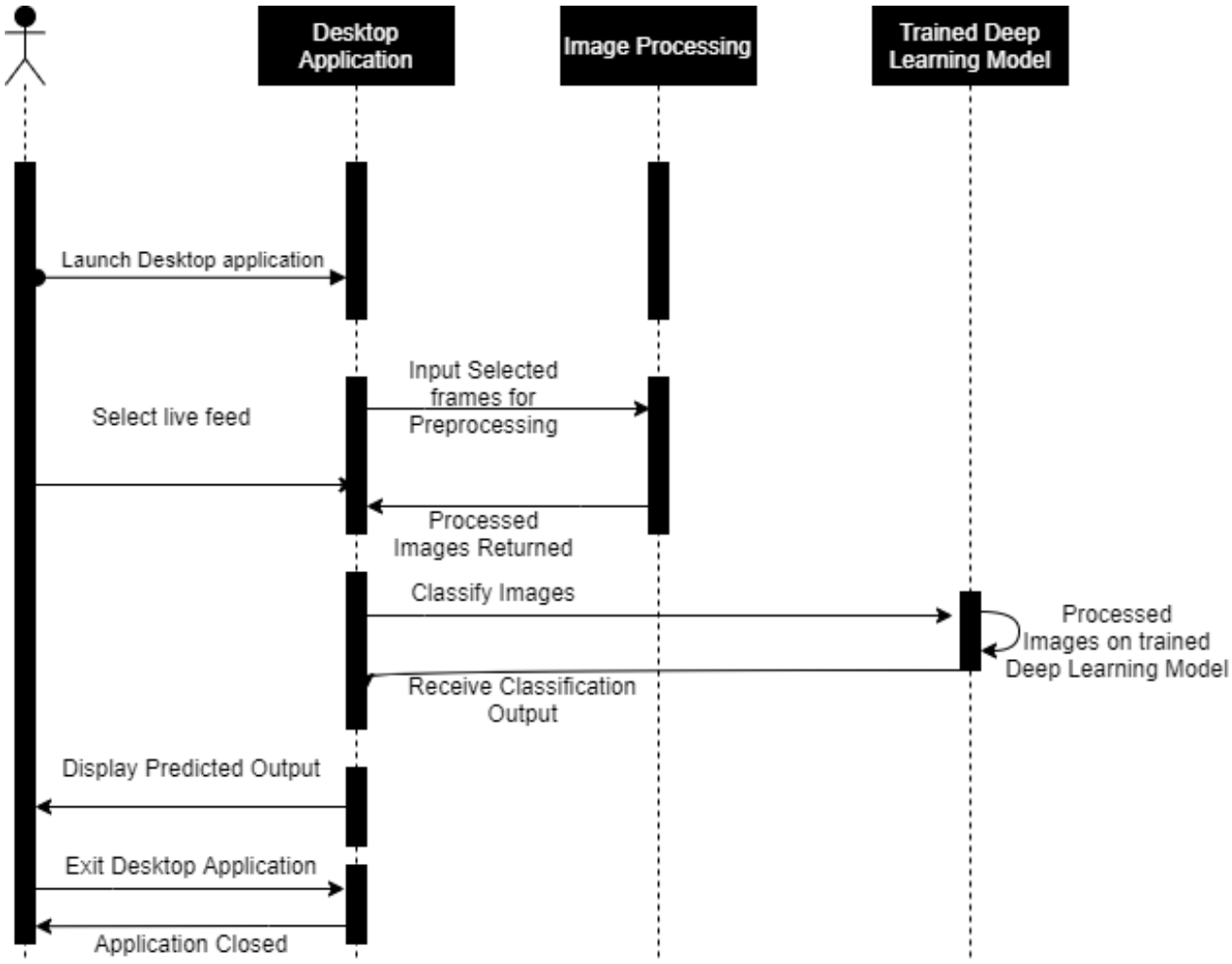


4.4. Class Diagram

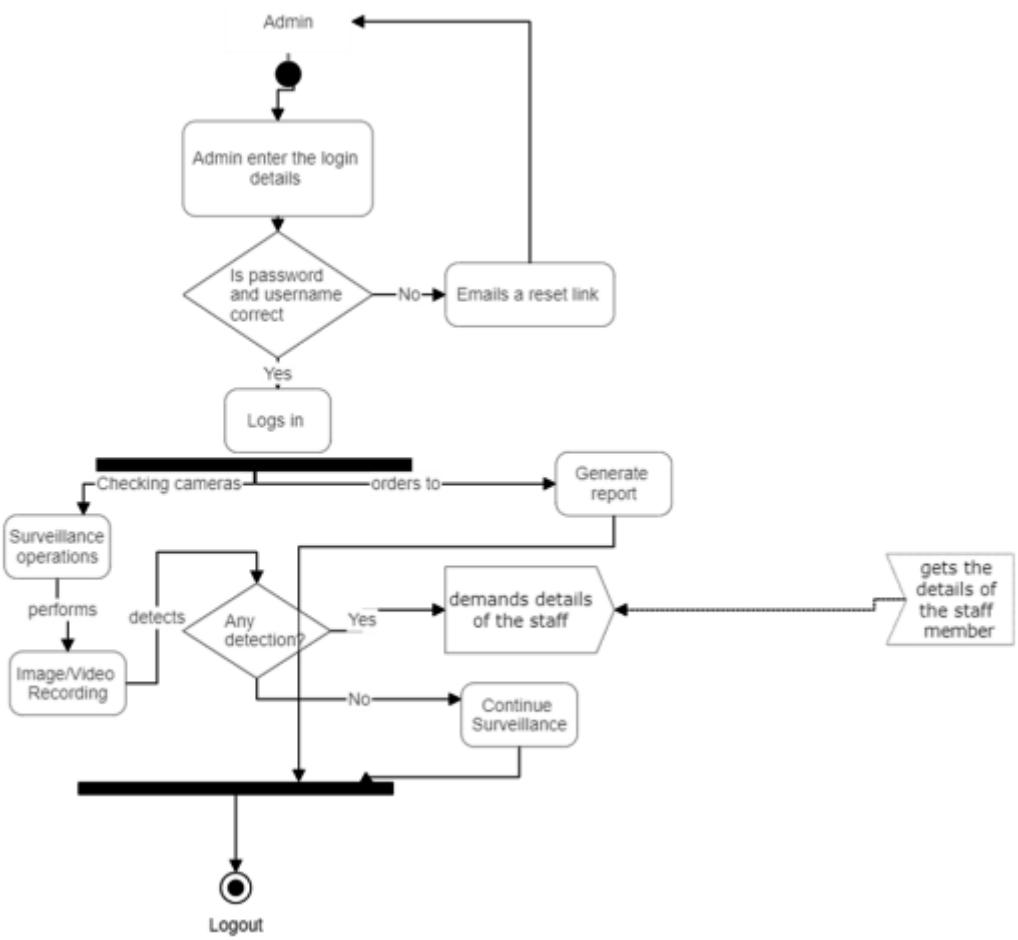


`

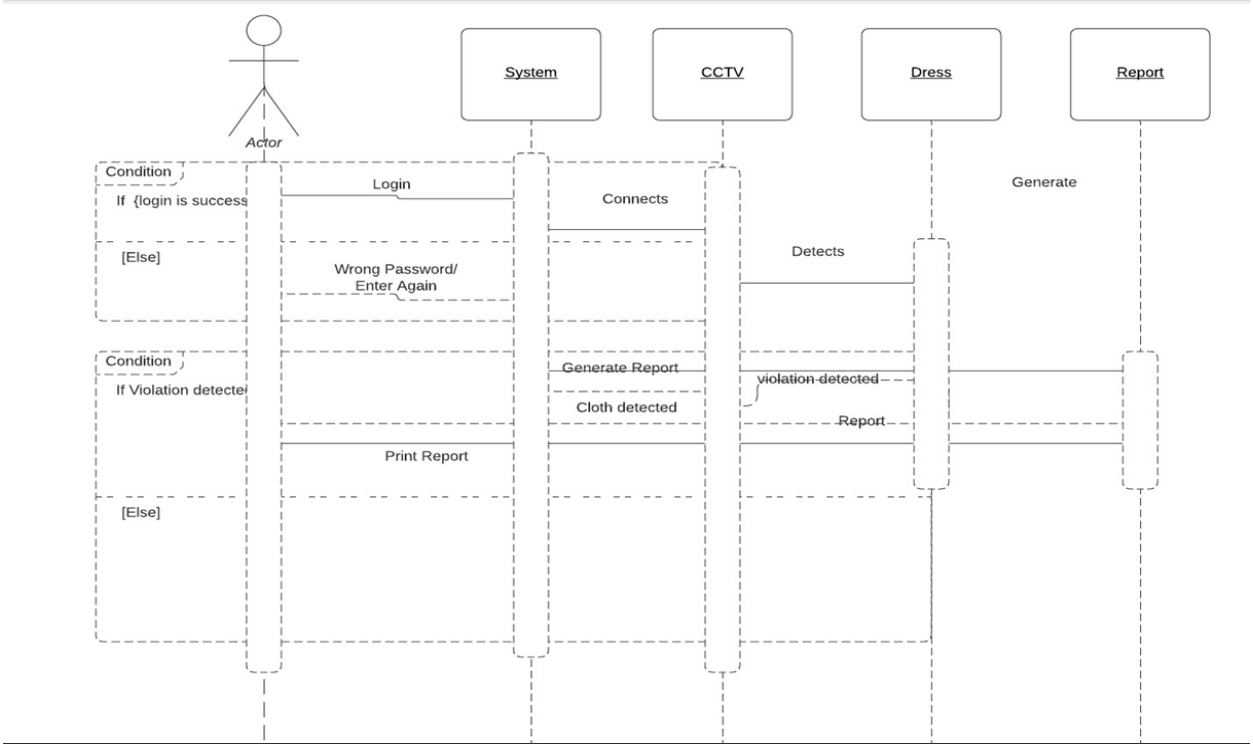
4.5. Sequence / Collaboration Diagram



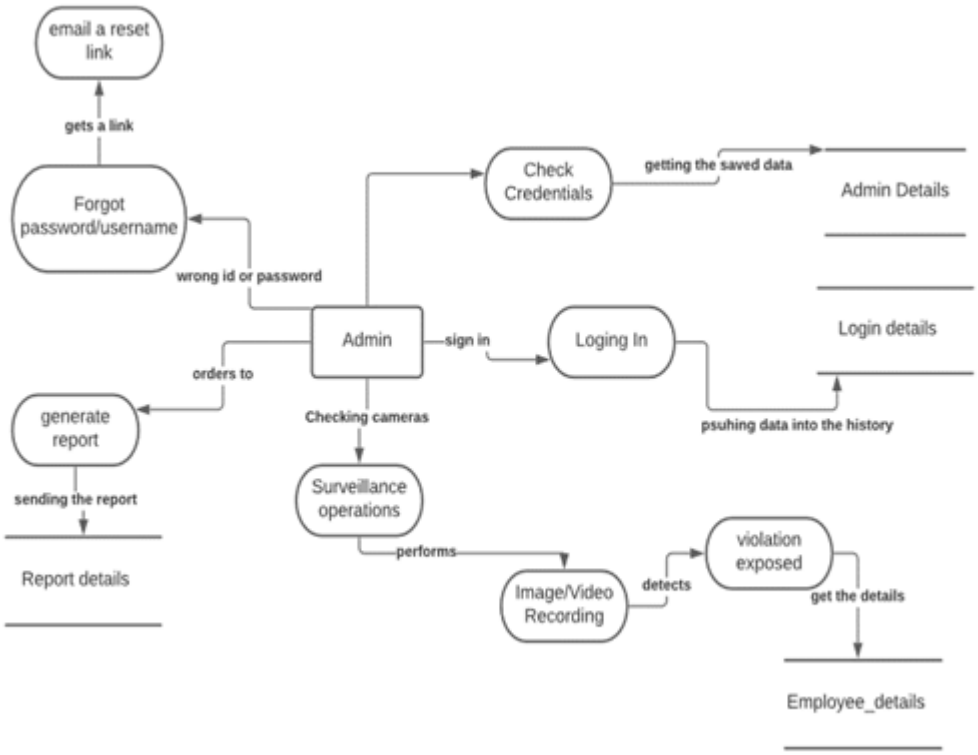
4.6. Activity Diagram



4.7. Component Diagram



4.8. Data Flow diagram



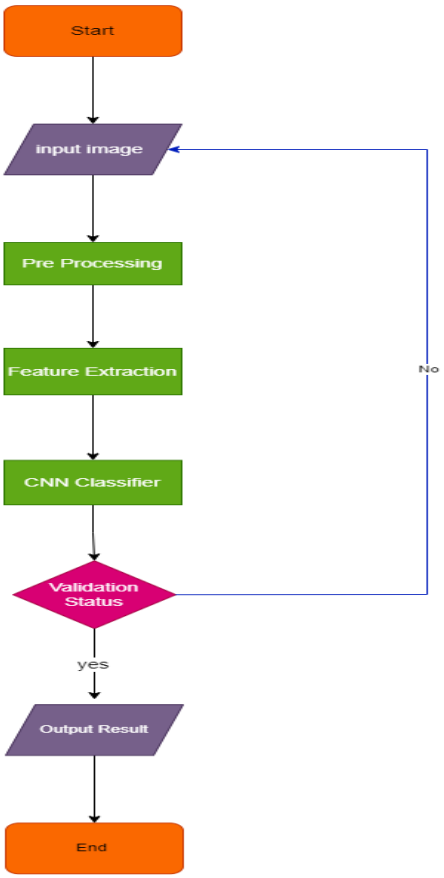
Chapter 5
Implementation

Chapter 5

Implementation

Our Project is a technique for identifying uniform regions in images, or parts of the image with a constant color or texture. The usage of a sliding window approach, in which a small window of a given size is moved over the image and the pixels inside it are examined to see if they are uniform, is a popular method for implementing uniform detection. The window is thought to have discovered a uniform zone if it can be determined that the pixels are uniform.

5.1. Important Flow Control/Pseudo codes



5.2. Deployment Environment

Deployment in our Project refers to the process of using a trained model to detect uniforms in an image or video. The model will typically be trained using a dataset of labeled images that contain uniforms, and then deployed to a computer or device where it can be used to analyze new images. To detect uniforms, the model will typically look for patterns and features in the image that are characteristic of uniforms, such as the shape of the jacket or the color of the pants.

5.3. Tools and Techniques

1. *Camera system will install.*
2. *Latest python version*
3. Operating System**:** Windows
4. *CPU: core i7 6th gen*
5. *Ram: 8gb*
6. *Storage: 20Gb*
7. *Gpu: 2Gb Graphic Card*

5.4. Best Practices / Coding Standards

Data collection: Collect a large dataset of images or videos of people wearing uniforms. This dataset should include a variety of different uniforms and backgrounds to ensure the model can generalize well to new images.

Data preprocessing: Preprocess the images or videos to resize them to a consistent size, and possibly convert them to grayscale. It might be helpful to label the images with the specific type of uniform present in the image.

Model training: Train a machine learning model, such as a convolutional neural network (CNN), on the preprocessed dataset. Use techniques such as data augmentation to increase the diversity of the training dataset and prevent overfitting.

Model evaluation: Evaluate the performance of the model using a separate dataset, and use techniques such as cross-validation to estimate the model's performance on unseen data.

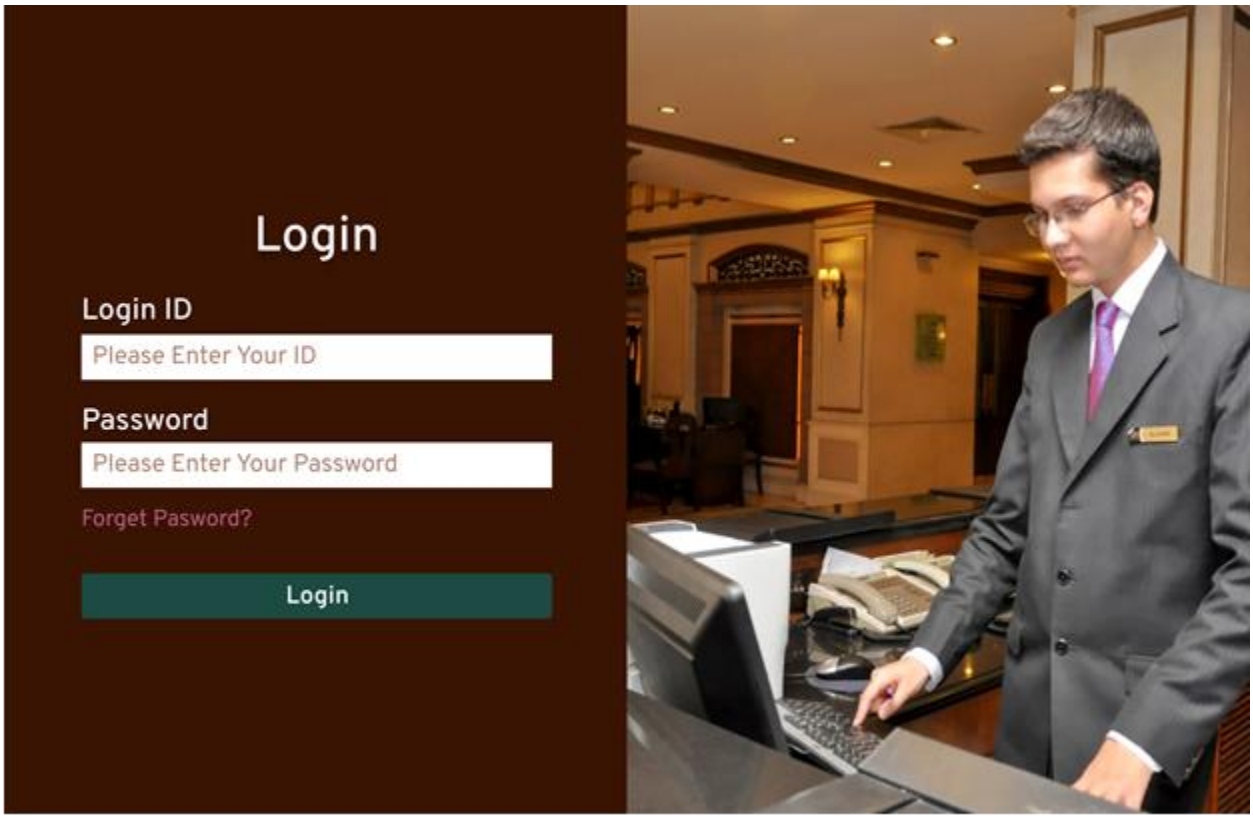
Model deployment: Once the model has been trained and evaluated, deploy it to a production environment where it can be used to classify new images or videos. Version Control

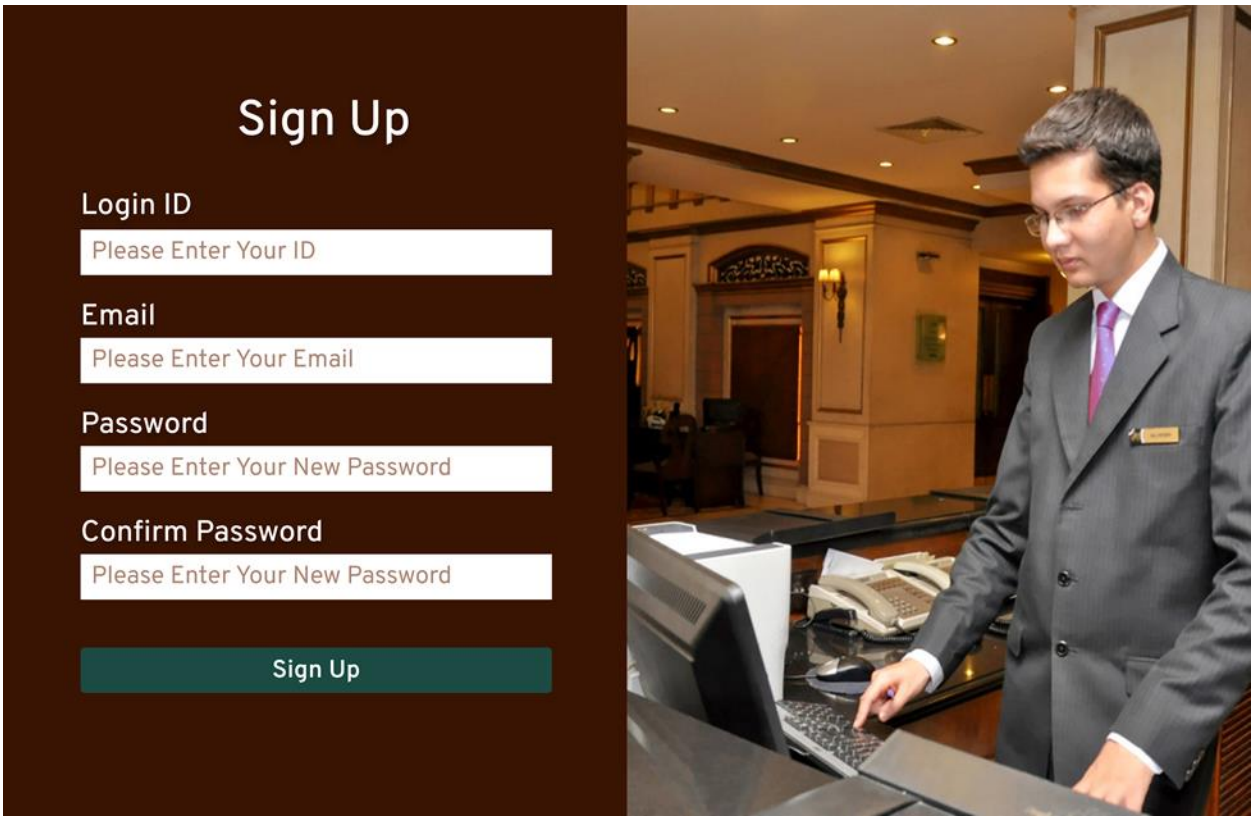
5.5. Version Control

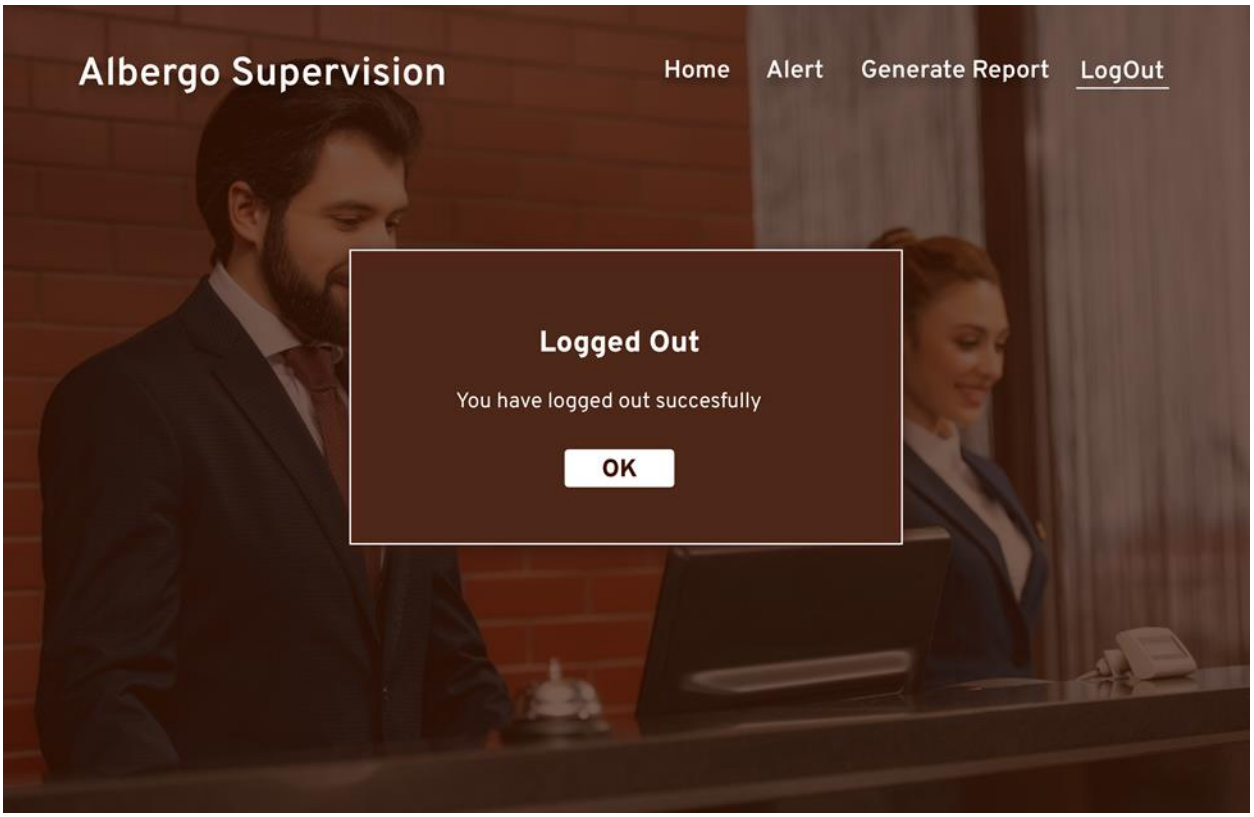
Version control is a method for keeping track of changes to files and documents so that you always know which version is the most recent. We can effectively track and control changes to these documents within our software by using project management or version control systems.

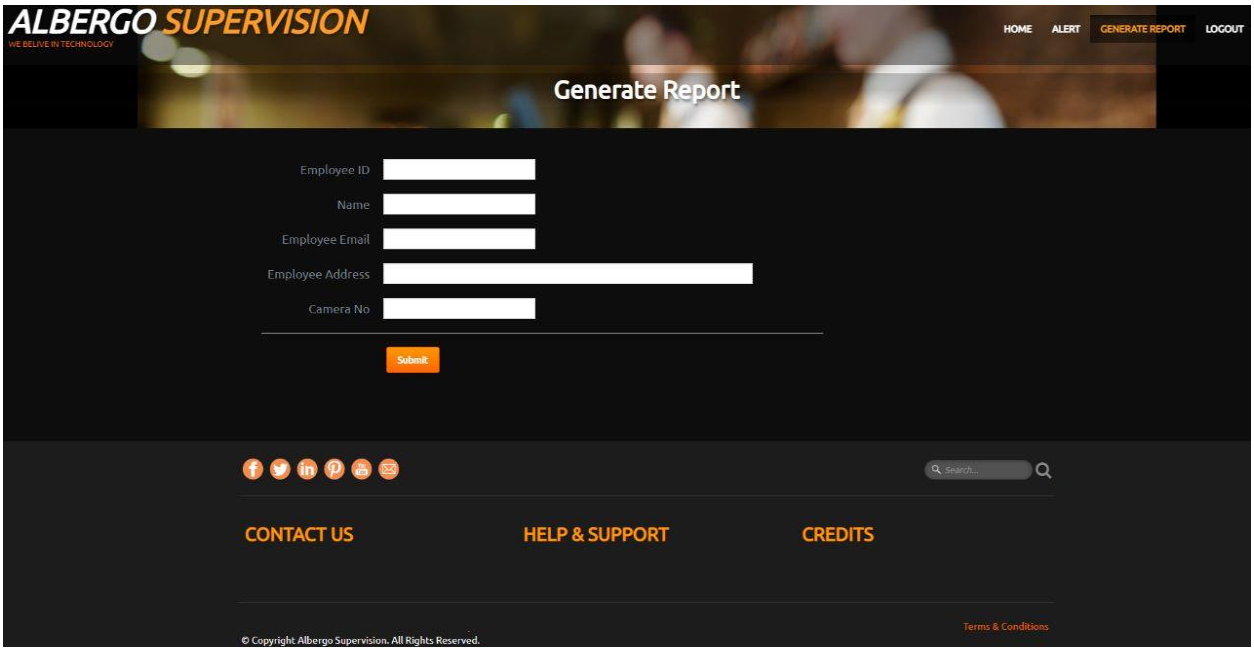
A system called version control keeps track of changes made to a file or set of files over time so that you can later recall versions.

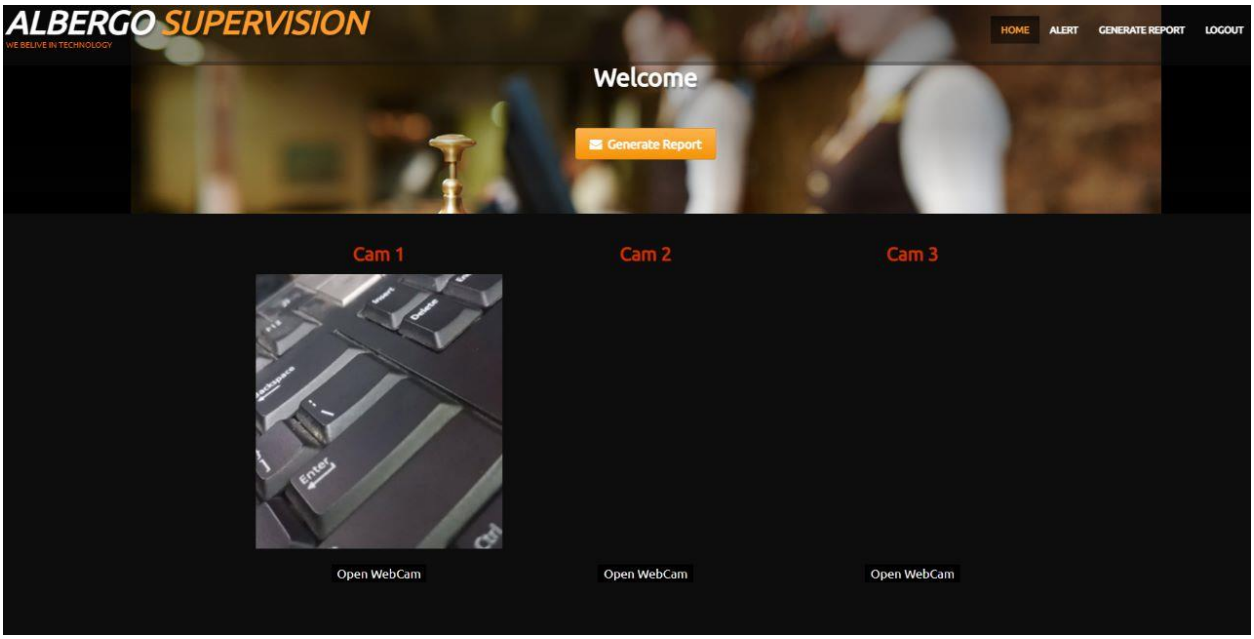
A version management system keeps track of all file modifications so that if needed, a specific version can be called later.




Sign Up
Login ID
Please Enter Your ID
Email
Please Enter Your Email
Password
Please Enter Your New Password
Confirm Password
Please Enter Your New Password
Sign Up


Albergo Supervision
Home
Alert
Generate Report
LogOut
Logged Out
You have logged out succesfully
OK


ALBERGO SUPERVISION
WE BELIVE IN TECHNOLOGY
HOME
ALERT
GENERATE REPORT
LOGOUT
Generate Report
Employee ID
Name
Employee Email
Employee Address
Camera No
Submit
CONTACT US
HELP & SUPPORT
CREDITS
Terms & Conditions
© Copyright Albergo Supervision. All Rights Reserved.


ALBERGO SUPERVISION
WE BELIVE IN TECHNOLOGY
HOME
ALERT
GENERATE REPORT
LOGOUT
Welcome
Generate Report
Cam 1
Cam 2
Cam 3
Open WebCam
Open WebCam
Open WebCam

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Albergo supervision involves identifying and evaluating instances where a certain code or set of rules is not being consistently followed across a system or organization. Here are some common methods for testing and evaluating uniform code detection.

6.1. Use Case Testing

6.1.1Name of Use-Case 1: Login

Identifier	*Admin Login*	
Purpose	*The access of admin to login to the system*	
Priority	*High*	
Pre-conditions	*Admin already have an account*	
Post-conditions	*Admin can access portal or application and can update login password information.*	
Typical Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin enters the username and password to login the system*	*Portal will be log in with the registered account.*
2	*Admin can change username and password for their dire need.*	*Database of the user account will be updated with the new username and password.*
Alternate Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin clicks on forget Password*	*A new password will be created by the admin and admin login again*

6.1.2 Name of Use-Case 2: Logout

Identifier	Admin Logout
Purpose	logout from Account
Priority	High
Pre-conditions	Admin Must be login

Post-conditions	Session is destroyed	
Typical Course of Action		
S#	***Actor Action***	***System Response***
1	User clicks on the menu	Different options appear
2	User chooses logout option from the menu	Session destroyed and after appear login page
Alternate Course of Action		
S#	***Actor Action***	***System Response***
1	User does not choose logout option	Session remains

6.1.3Name of Use-Case 3: Dress Code Checking

Identifier	*Administrator Checking*	
Purpose	*To check and balance over the system results and response to the security regulators.*	
Priority	*High*	
Pre-conditions	*Software Running with integrated cameras.*	
Post-conditions	*Detected people data will be loaded to a file.*	
Typical Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin wants to see the people recognized by the system Staff Dresses*	*System will show the captured and processed results of Stuff*
2		
Alternate Course of Action		
S#	***Actor Action***	***System Response***
1	*Admin wants to update the log list generated by the system of perpetrators*	*System will update the log list for more results concluded in real time video feed*
2	*Admin wants to delete any faulty or mistaken result generated by the system*	*System will delete the specific wrong result miscalculated by the recognition feature.*

6.2. Data flow testing

Identify Code Units: Locate the code snippets or portions that are involved in the data flow. These could be any units that manage or manipulate data, including classes, methods, and functions.

Determine Data Sources: Determine the data sources from which the code receives data and the data sinks from which the data is emitted or consumed. This aids in mapping the data flow over the entire codebase.

Develop Test Cases: Create test cases that confirm the data flow compared to the specified rules. Every test case should concentrate on certain data flow scenarios and confirm that the data is processed consistently in accordance with the established criteria.

6.3. Unit testing

Define Uniformity Standards: Establish a set of rules, criteria, or coding standards that specify the expected uniform behavior of the code. This is known as "defining uniformity standards." This may include naming conventions, error handling, input validation, code organization, formatting norms, and other pertinent coding practices.

Locate Specific Code Units: Locate the specific code units that require uniformity testing. These pieces could be discrete classes, methods, functions, or other types of code that allow for separate evaluation.

Develop Test Cases: Create test cases that specifically address the uniformity specifications mentioned in the standards. Every test case ought to concentrate on one component of the uniformity requirements, such as regular indentation, naming conventions, or the use of patterns.

6.4. Performance testing

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

The project involves the establishment of clear and comprehensive coding standards that encompass various aspects such as code formatting, naming conventions, code structure, error handling, and other relevant guidelines. The primary objective of the project is to detect and address deviations from the defined coding standards to achieve a uniform codebase. By enforcing uniformity, the project aims to enhance code consistency, reduce bugs, improve collaboration among developers, and facilitate code maintenance and scalability.

7.2. Achievements and Improvements

The project succeeded in achieving a greater level of code consistency throughout the codebase by imposing uniform coding standards. Now that developers are using standard naming conventions, formatting, and structure, the code is easier to read and manage.

Reduced Code Duplication: By concentrating on consistent detection, the project was able to spot instances of code duplication. The project greatly decreased code redundancy and increased code efficiency by resolving these duplicates and encouraging the use of reusable code components.

7.3. Critical Review

Uniform code detection is a valuable practice for promoting code consistency and quality, it's important to critically review its implementation and potential challenges. Coding standards best practices change over time as new frameworks, tools, and techniques are developed. The coding standards should be reviewed and updated frequently to account for changes in the industry. Failure to comply with the requirements could result in antiquated procedures and impede the project's overall development.

Reference and Bibliography

1. https://ieeexplore.ieee.org/abstract/document/9182994
2. https://ieeexplore.ieee.org/abstract/document/9361017
3. https://link.springer.com/chapter/10.1007%2F978-981-13-3329-3_16
4. https://www.semanticscholar.org/paper/Clothing-identification-via-deep-learning%3A-forensic-Bedeli-Geradts/b53a55d624269228a761301abab1d80ade9f89c5