WatchDog AI

Project Team

Muhammad Hisan Usman	21I-0336
Shafqat Mehmood	21I-0324

Session 2024-2025

Supervised by

Dr. Usman Haider

Co-Supervised by

Ms. Hifza Umer

Department of Computer Science

National University of Computer and Emerging Sciences
Islamabad, Pakistan

November, 2024

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The purpose of this project proposal document is to clearly state the aims and goals of our project, allowing the stakeholders and proposer to better understand the projects scope and focus. The document will also provide a road-map for how the project will be executed. Moreover, the document will articulate the problem or aim that the project will be addressing, justifying the importance and need of our project.

1.1 Existing Solutions

For the Existing Solutions section, students should research and describe various solutions, methods, or approaches that have been previously developed to solve the problem they are addressing. This section should include an overview of the different techniques, technologies, or models that are currently in use or have been applied in the past. Students should critically assess the strengths and weaknesses of each solution, highlighting any gaps or limitations that their project aims to address. Additionally, this section should show an understanding of the state-of-the-art in the field and provide context for why the proposed solution is necessary or an improvement over existing ones.

Table 1.1: Comparison of Existing Solutions

System Name	System Overview	System Limitations
Deep Sentinel	Smart surveillance camera system that uses AI for real-time human detection.	For outdoor activities/threats only.
Avigilon Appearance Search	Uses deep learning AI to search and track individuals across multiple cameras.	Requires advanced knowledge of operating the system.
Actuate AI	Integrates with existing security cameras to detect weapons, intruders, and other threats using AI-based video analytics.	Focus on specific detections does not suit the broader surveillance needs.

1.2 Problem Statement

Our traditional CCTV systems are able to capture and store the footage, however, they are unable to analyze it in real-time. Moreover, our current systems are manually monitored, resulting in human error and fatigue, leading to missed threats and delayed responses. Even our current AI systems have limitations, whether that is in accuracy, adaptability or real-time processing.

To solve and automate all of these existing issues, we are developing an advanced model, capable of detecting violence and criminal activities, detecting criminals using facial recognition, and implementing scene understanding to automatically generate reports using language models. This software will help in the timely detection of violent activities and criminals, improving security and providing a safer environment for everyone.

1.3 Scope

Due to the high importance and sensitivity of an advanced surveillance system, our software will have multiple functionalities in the scope.
1. The software will be detecting a broad range of criminal and violent activities, like weapons, abuse, arson, burglary, shooting, stealing, vandalism, violence.
2. It will also be trained to detect criminals (using facial recognition), who are known to be associated with criminal activities or who are on a watch-list.
3. The software will be capable of pin-pointing the exact location of the crime in a public/crowded place.
4. It will reduce the need to check the complete CCTV footage, by bookmarking the times of the violent activities.
5. Scene understanding will also be integrated in this system.
6. Automatic report generation will be implemented in the software.
7. The system will also be providing real time and instant alerts to the user and authorities (based on the severity of the crime).
8. The software will provide the user with a web based app for monitoring the footage and surveillance.
9. The system will be integrated with existing security systems, allowing a seamless transition and operation.

1.4 Modules

This section describes the key functional units or modules of our project, each designed to perform a specific task. The breakdown of the project into modules ensures that each part of the system is manageable and addresses specific objectives.

1.4.1 Module 1 - Violent activity + Criminal detection

In module 1, we will be training models to detect violent and criminal activities, and implement facial recognition to detect and recognize criminals in a CCTV footage.

1. Training of models to detect criminal and violent activities - major base of our systems which will have future models integrated with it.
2. Implementation of facial recognition to detect criminals - those individuals listed as a criminal will be detected, regardless of any activity happening or not.

1.4.2 Module 2 - Scene understanding + Report generation

In module 2, we will be implementing scene understanding and adding the feature of automatic report generation. We will also be creating a web app which will allow the user to monitor CCTV footage.

1. Implement scene understanding and integrate automatic report generation - for the ease of our user, the model will automatically generate reports based on the scene understanding.
2. Create a user friendly web app, to allow the user to monitor CCTV footage - all modules will be integrated/implemented in this system.

1.5 Work Division

Work division:

Table 1.2: Work Division

Name	Registration	Responsibility/ Module / Feature
Mr. Muhammad Hisan Usman	21i-0336	(Module 1- Feat 1) Violent activity recognition
Mr. Muhammad Hisan Usman	21i-0336	(Module 1- Feat 3) Pin-pointing exact location of crime
Mr. Shafqat Mehmood	21i-0324	(Module 1- Feat 2) Criminal detection and recognition
Mr. Shafqat Mehmood	21i-0324	(Module 1- Feat 4) Bookmark time-period of activity
Mr. Muhammad Hisan Usman	21i-0336	(Module 2- Feat 5) Scene understanding
Mr. Muhammad Hisan Usman	21i-0336	(Module 2- Feat 6) Automatic report generation
Mr. Shafqat Mehmood	21i-0324	(Module 2- Feat 7) Real-time alerts to authorities
Mr. Shafqat Mehmood	21i-0324	(Module 2- Feat 8) Integration of web-app

Chapter 2

Project Requirements

This section outlines the necessary requirements for the successful completion of the project. Project requirements can be divided into two main categories: functional requirements, which describe the system's core operations, and non-functional requirements, which specify performance, security, and usability standards.

2.1 Use-case/Event Response Table/Storyboarding

Table 2.1: Event Response Table

Event	System Response
User login	The system will allow the user to enter their credentials and login to the web app of the application.
User clicks the "Next/Prev camera" button	The system shifts to the next or previous camera, respectively, and displays the CCTV footage from that camera. The user can click these 2 buttons infinite number of times and the display will start showing footage from the next/previous CCTV camera.
System detects a violent activity	The system will immediately alert the user about the activity detected and its type.
System detects a criminal	The system will immediately alert the user and display the details of the criminal on the web-page, allowing the user to alert the authorities (if needed).
User clicks on "Report generation"	The system will automatically generate a report for the user, containing details of the event occurred along with the scene understanding done by the model.
User clicks on "Bookmarks"	The system will show the time periods that were book marked for detecting any sort of suspicious activity.

2.2 Functional Requirements

This section describes the functional requirements of the system expressed in the natural language style. This section is typically organized by feature as a system feature name and specific functional requirements associated with this feature. It is just one possible way to arrange them. Other organizational options include arranging functional requirements by use case, process flow, mode of operation, user class, stimulus, and response depend on what kind of technique has been used to understand functional requirements. Hierarchical combinations of these elements are also possible, such as use cases within user classes.

2.2.1 Module 1

Following are the requirements for module 1:

1. The system will be capable of detecting violent activities in a CCTV footage.

2. The system will immediately be able to detect any criminals in a CCTV footage.

2.2.2 Module 2

Following are the requirements for module 2:

1. The system will be able to do scene understanding and generate reports based on its understandings.
2. The backend will be integrated with the frontend and a web app will be delivered as a final product.

2.3 Non-Functional Requirements

This section specifies nonfunctional requirements.

2.3.1 Reliability

In case of a loss of signal/connectivity, the system may fail for a few seconds as that specific CCTV camera will be unavailable for recording any footage. This, however, is very rare to happen and may only occur in case of damage to the hardware of the CCTV camera or any cut-off in power lines. If the system is not backed up in case of power outages, it will not be available until the power is restored. The availability of a backup power source can resolve this issue. Moreover, the system will fail to detect any violent activities if the view is blocked by an object.

2.3.2 Usability

The system will be easy to use for not only a CCTV inspector, but anyone who wishes to check the footage's. The feed from camera(s) will be displayed on the screen and the user can go onto the next or previous cameras by clicking on their respective buttons. A button will also be available for automatic report generation from the trained model. The web app will be easy to navigate and control, and will not require any technical expertise.

2.3.3 Performance

No specific performance requirements needed for the system operations except for a web monitor. The system can be integrated with any CCTV camera, thus, there is no specific requirement. However, for faster detections and to avoid any delays, it is recommended to have a system with at least 16GB of RAM and a GPU.

2.3.4 Security

In order to prevent the system being hacked or any data leakage, it is essential that the user keeps their credentials confidential. Moreover, the criminal dataset (for criminal detection) should not be exposed and its database access should not be with every user as the privacy of these individuals is a major concern and threat.

Chapter 3

System Overview

A general description of the functionality, context, and design of our project.

3.1 Architectural Design

In the first module, we'll start by creating multi-models for violent activities detection. These models will automatically detect any activity that happens in the CCTV footage. This subsystem will be integrated with another subsystem, that will be detecting criminals that appear in the CCTV footage. In the second module, we'll be creating a subsystem for scene detection. This system will be integrated with the previous two, as it will be using the results from the trained models for scene understanding. The scene understanding subsystem will be merged with the subsystem for report generation, which would be using the results of scene understanding for automatically generating a report. Lastly, a subsystem for pin-pointing the exact person who was involved in the violent activity, will be blended with the subsystem for scene understanding.

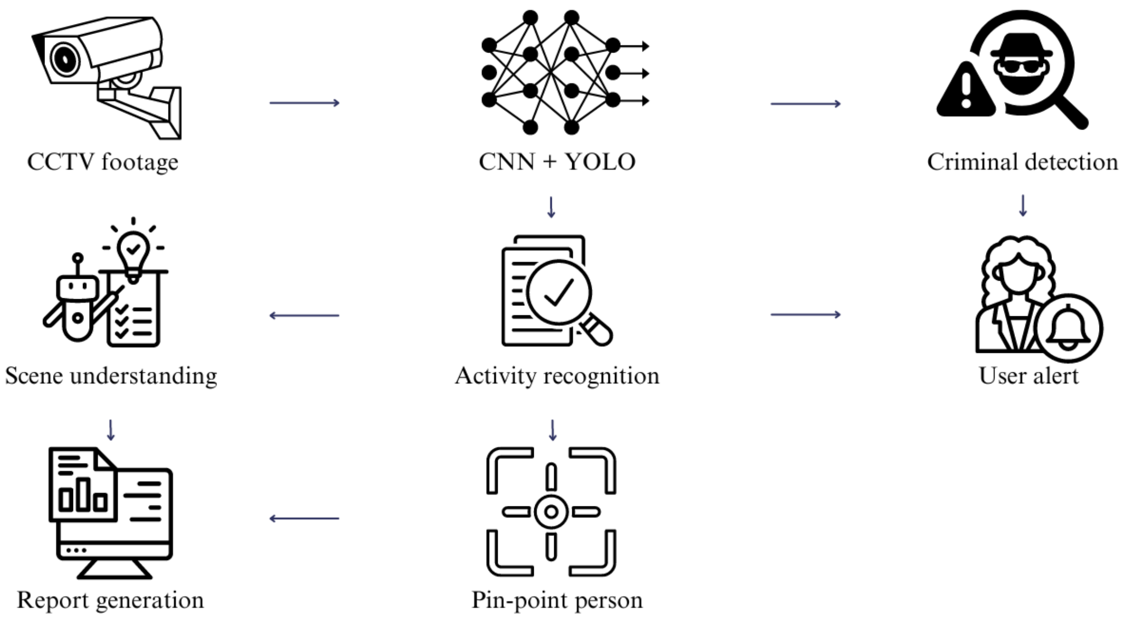

Figure 3.1: Architecture Diagram

3.2 Data Design

Criminal dataset, which consists of criminals and their details, will be stored in a database, and will be used for detecting criminals in a CCTV footage, in real time. This database will be integrated with the facial detection subsystem. Only a few members of the development team (not the end user) will have access to this database, ensuring that their is no compromise on security. Any other images that are used for model training will simply be downloaded on the local system and will be used to train and test the model on.

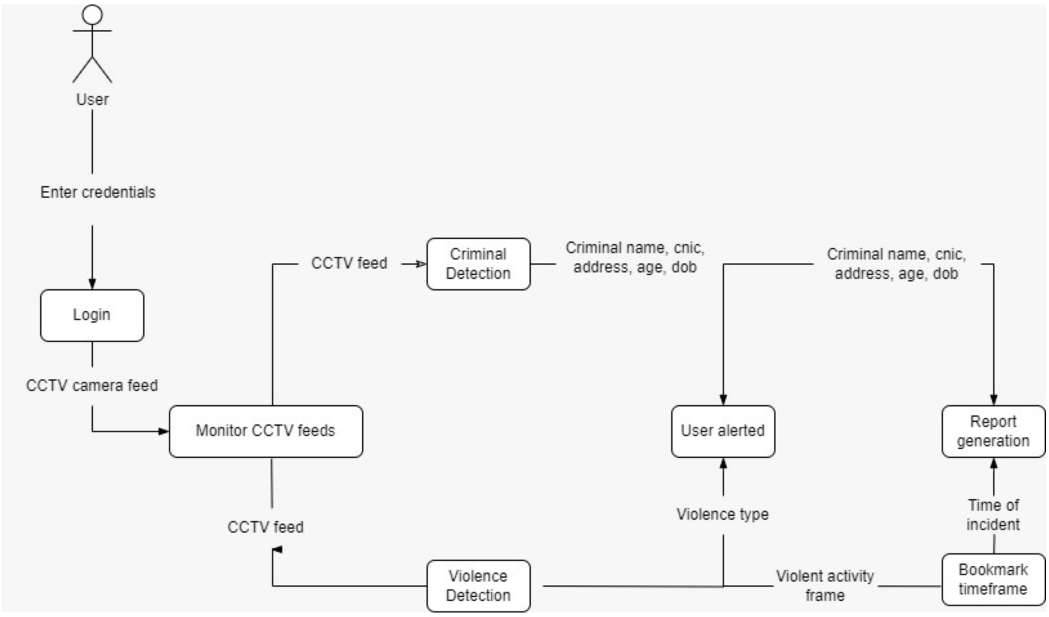

Figure 3.2: Dataflow Diagram

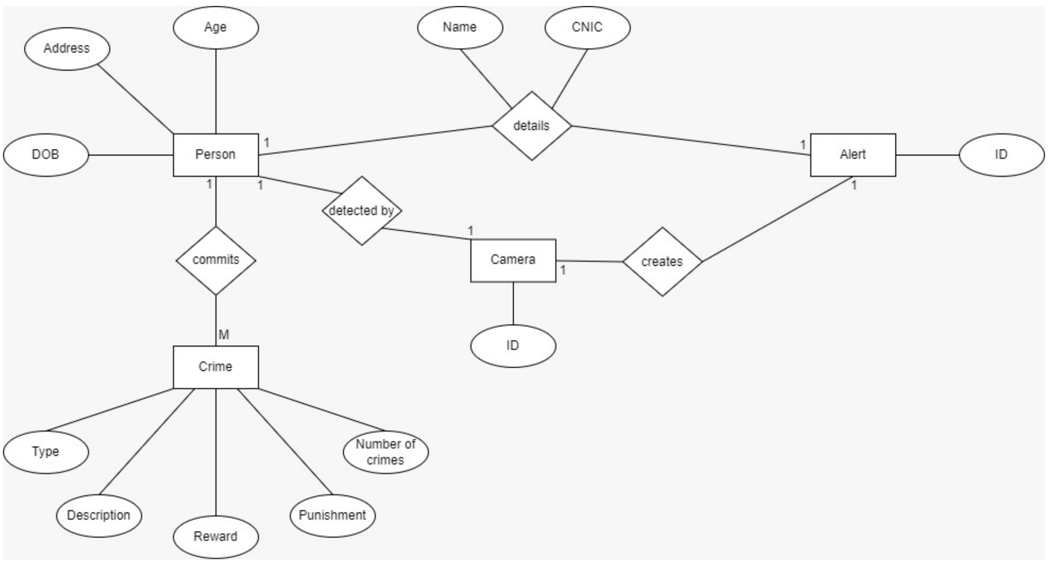

Figure 3.3: ERD Diagram

3.3 Domain Model

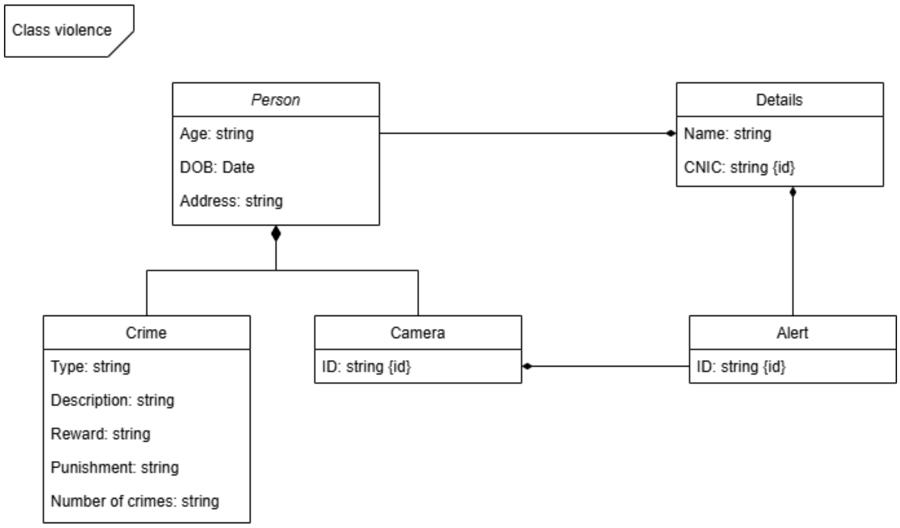

Figure 3.4: Domain Model Diagram

3.4 Design Models

The activity diagram helps us better understand the flow from one system to another. Initially a user is asked to either login (if returning user) or sign-up, after which they're taking to the home page/dashboard where they can monitor the feed. If any violent activity or a criminal is detected, the user will be immediately alerted. The exact person will be pin-pointed and the user may generate a report for the event(s) that happened.

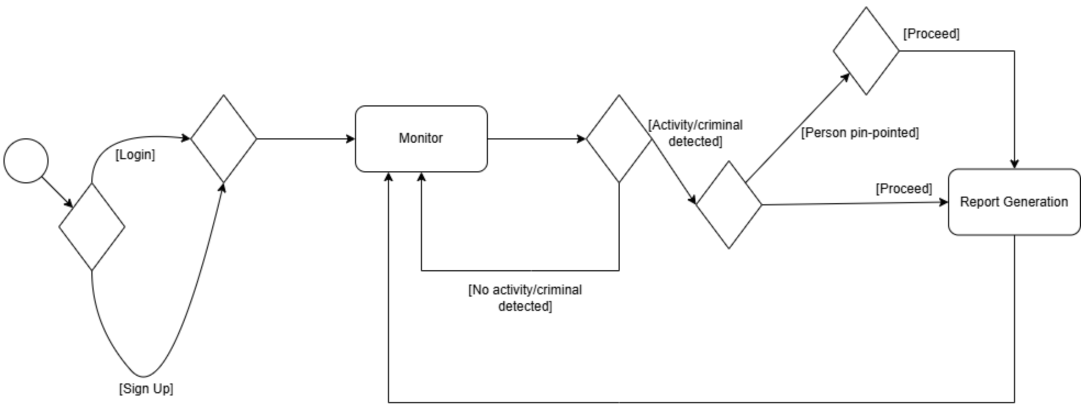

Figure 3.5: Activity Diagram

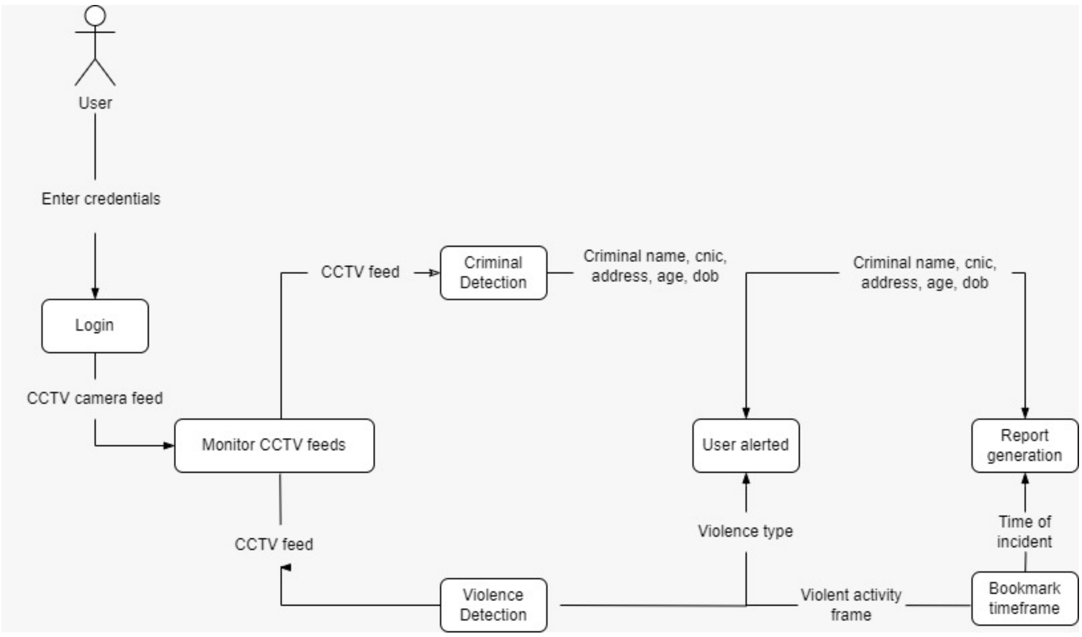

Figure 3.6: Dataflow Diagram

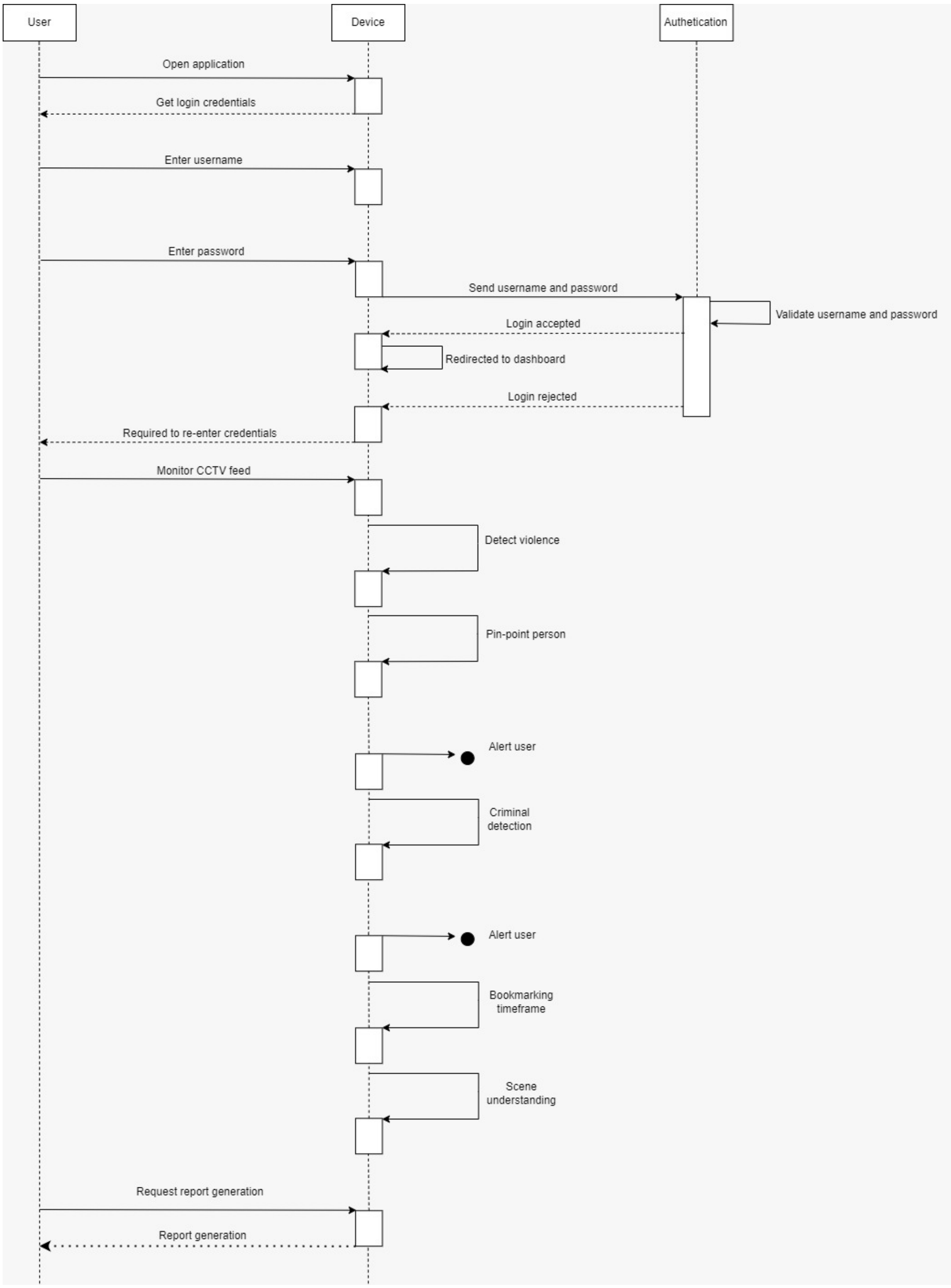

Figure 3.7: Sequence Diagram

Chapter 4

Implementation and Testing

To counter the problem of human error in CCTV monitoring, WatchDog AI is automating this process by providing its users with a security system capable of detecting violent activities and criminals in the CCTV footage, immediately alerting the user and bookmarking the particular time-frame for later review. The system also allows the user to generate automatic reports which can further help in the event of a violent activity.

4.1 Algorithm Design

```
Pseudocode for violence detection and pin-pointing:

Initialize pose detection model (YOLO-Pose)
Initialize weapon detection model (YOLO-Weapon)


WHILE video has frames:
    Read the next frame
    IF frame is valid:
        IF current frame index is a multiple of frame_interval:
            Run scene understanding model to get object detections
            Run pose detection model to detect human poses
            Run weapon detection model to detect weapons

            Format detections from each model:
                Extract bounding boxes, labels, and confidence scores

            Combine detections from all models

            Map combined detections to activity labels

            Annotate frame:
                Add activity label as text overlay
                Draw bounding boxes with labels and confidence scores

            Find the person closest to a weapon:
                FOR each detected person:
                    Calculate distance to each weapon
                    Track the person with the minimum distance

            IF closest person is found:
                Crop and resize the person's face
                Display cropped face in a designated corner of the frame

            Write the annotated frame to the output video file

    Increment frame index
    Check for quit signal (e.g., pressing 'q')

Release video capture and video writer resources
```

Figure 4.1: Algorithm Design

<u>Pseudocode for Facial Recognition Pipeline:</u>

Load a pre-trained MobileNetV2 model.
Replace the final classification layer for binary output:
Output: 1 (criminal) or 0 (not criminal).
Move the model to GPU if available.
Define Training Pipeline

Loss Function: Binary Cross-Entropy Loss (BCEWithLogitsLoss).
Optimizer: Adam optimizer with a learning rate of 0.001.
Training Loop:
For each epoch:
Pass data through the model.
Compute loss.
Backpropagate and update weights.
Print the epoch-wise loss.
Validation

Set the model to evaluation mode.
Loop through validation data:
Make predictions using the trained model.
Compute accuracy as the percentage of correct predictions.
Save Trained Model
Save the model weights to a .pth file for future use.

Prediction Pipeline

Load the saved model and set it to evaluation mode.
Preprocess an input image (resize, normalize).
Pass the image through the model to get predictions.
Convert predictions into human-readable labels (e.g., "criminal" or "not criminal").

Figure 4.2: Algorithm Design

4.2 External APIs

Table 4.1: External API's Table

API and version	**Description**	**Purpose of usage**	**API endpoint/function/class used**
Roboflow	Dataset import	Importing weapons and violence datasets	https://api.roboflow.com/
Kaggle	Dataset import	Importing violence dataset(s)	https://github.com/Kaggle/kagglehub ?tab=readme-ov-file authenticate

4.3 Testing Details

4.3.1 Unit Testing

Test case ID	Test Objective	Precondition	Steps	Test data	Expected result	Post condition	Actual Result	Pass/Fail
TC0001	Display bounding boxes and basic detections	System should be passed some input data	Pass the input video and display the output (containing frames around detections like person, etc.)	Input video (.mp4)	Basic objects detected and bounding boxes drawn	-	As expected	Pass

TC0002	Detect violent activities	System should be passed some input data	Pass the input and display the output (containing detections and labeled activities)	Input video (.mp4)	Displayed video contains detections and labels for violent activity	Model should be running in the background	As expected	Pass
TC0003	Detect criminals	System should be passed some input data	Pass the input and display the output (containing detections and labeled activities)	Input video (.mp4)	Displayed video contains detections of criminals	Model should be running in the background	As expected	Pass
TC0004	Pin-point the person involved in violence	System should be passed some input data	Pass the input and display the output (containing detection of the person involved)	Input video (.mp4)	Displayed video contains detection of person involved	-	As expected	Pass

Bibliography

Appendix A

Appendices

A.1 Appendix A

A.1.1 Use Case Diagram example (Online Shopping System)

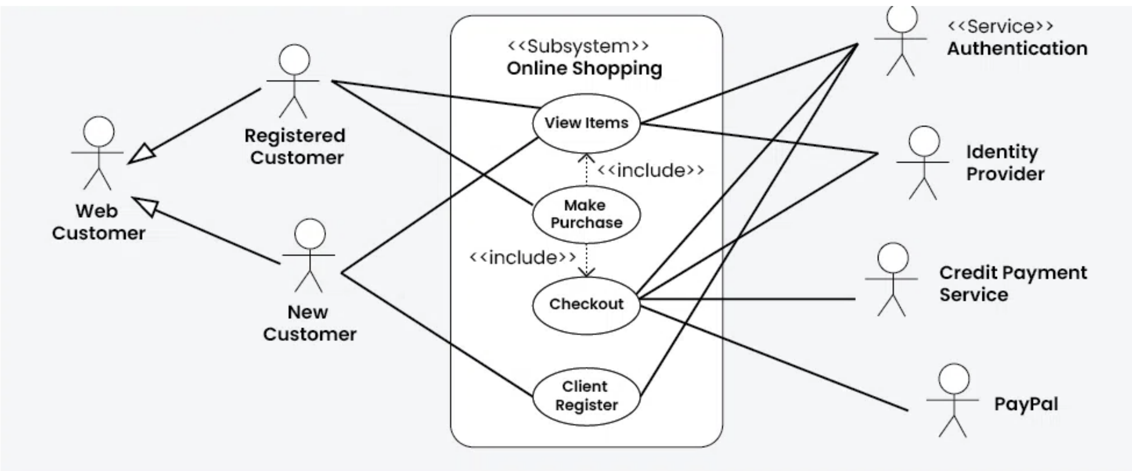

Figure A.1: Use Case Diagram for the Online Shopping System

A.1.2 Detail Use Case Example

ID	#1
Name	Overview elements
Short Description	All elements are shown in a list, where the user can set different filters.
Goal	Displaying elements in a changeable view.
Preconditions	The user got access to the system and is logged in. The user got the right to see schedules.
Success End Condition	The correct elements (filter and sorting) are displayed.
Fall End Condition	Elements not matching the criteria are displayed.
Stakeholder	Customer manager
Trigger	Login in as customer manager (because overview is on the starting screen for this role).
Normal Flow	1. The list of all elements matching default criteria are shown. 2. The user changes the filter criteria. 3. The user presses the Filter button. 4. The list shows all matching elements. Optional: 5. The user presses the Clear button. 6. The list of all elements matching default criteria are shown.
Alternative Flows	With click on the column headers, the list sorting can be changed. Different sorting for the different columns is described in the table below.
Includes	Login
Frequency of Use	About 50 times per day
Constraints and Special Requirements	None
Assumptions	None
Notes and Issues	None

Figure A.2: Detail Use Case Example

A.1.3 Event-Response Table for a Highway Intersection

Event	System State	Response
Road sensor detects vehicle entering left-turn lane.	Left-turn signal is red. Cross-traffic signal is green.	Start green-to-amber countdown timer for cross-traffic signal.
Green-to-amber countdown timer reaches zero.	Cross-traffic signal is green.	1. Turn cross-traffic signal amber. 2. Start amber-to-red countdown timer.
Amber-to-red countdown timer reaches zero.	Cross-traffic signal is amber.	1. Turn cross-traffic signal red. 2. Wait 1 second. 3. Turn left-turn signal green. 4. Start left-turn-signal countdown timer.
Pedestrian presses a specific walk-request button.	Pedestrian sign is solid Don't Walk. Walk-request countdown timer is not activated.	Start walk-request countdown timer.
Pedestrian presses walk-request button.	Pedestrian sign is solid Don't Walk. Walk-request countdown timer is activated.	Do nothing.
Walk-request countdown timer reaches zero plus the amber display time.	Pedestrian sign is solid Don't Walk.	Change all green traffic signals to amber.
Walk-request countdown timer reaches zero.	Pedestrian sign is solid Don't Walk.	1. Change all amber traffic signals to red. 2. Wait 1 second. 3. Set pedestrian sign to Walk. 4. Start don't-walk countdown timer.

Figure A.3: Example of Event Response Table

A.1.4 Story Board Example For Android App

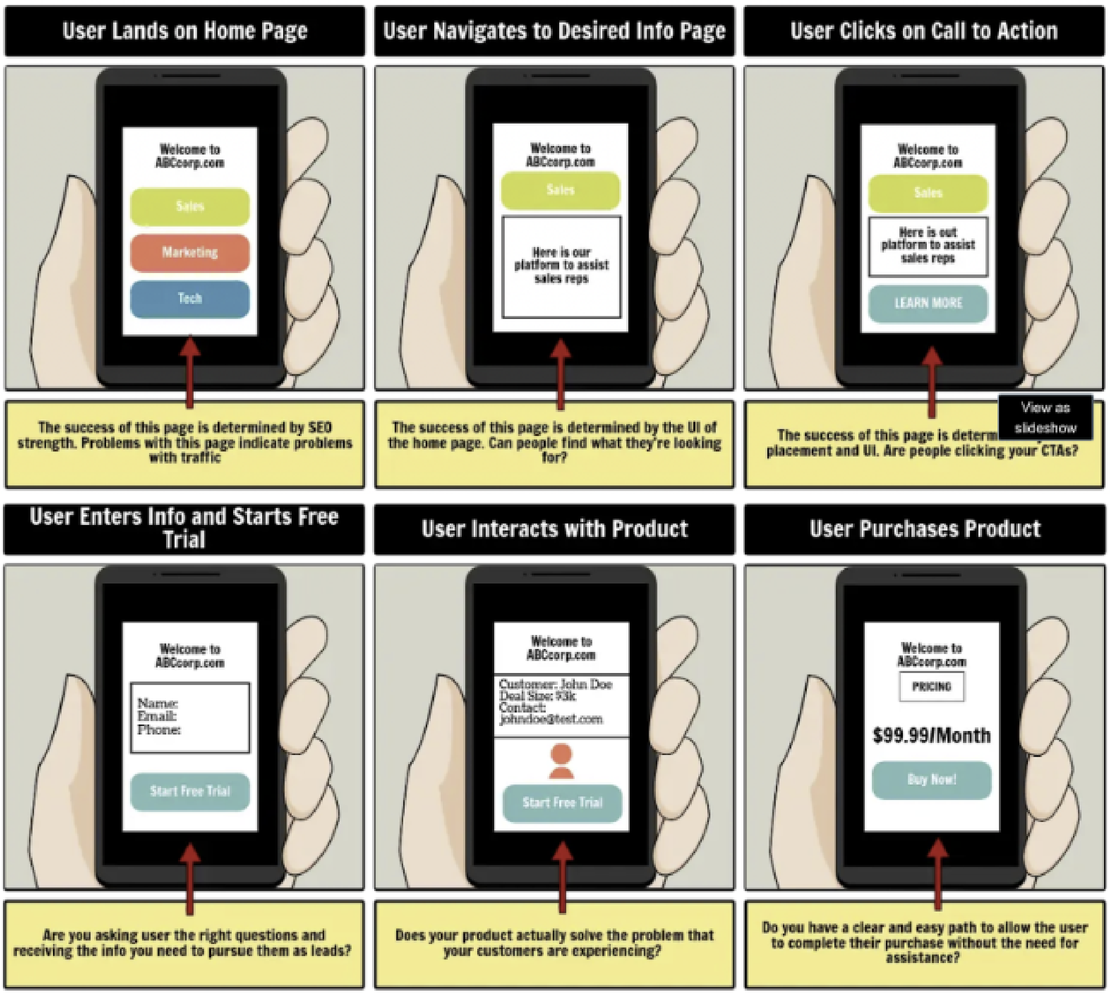

Figure A.4: Example of Story Board

A.2 Appendix B

A.2.1 Domain Model Example For Online Shopping

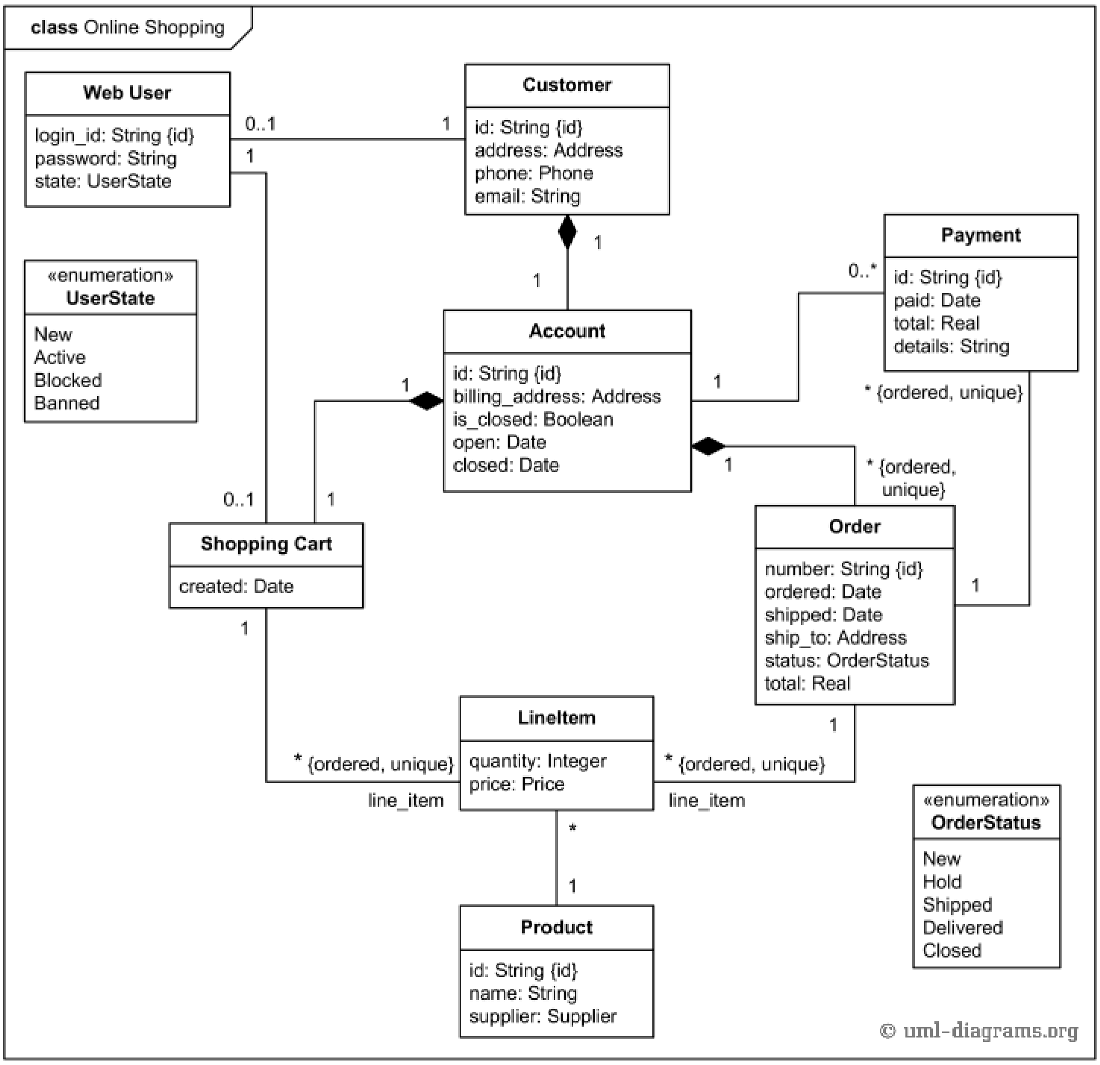

Figure A.5: Domain Model Example For Online Shopping Application

A.3 Appendix C

A.3.1 Box And Line Example For Online Shopping

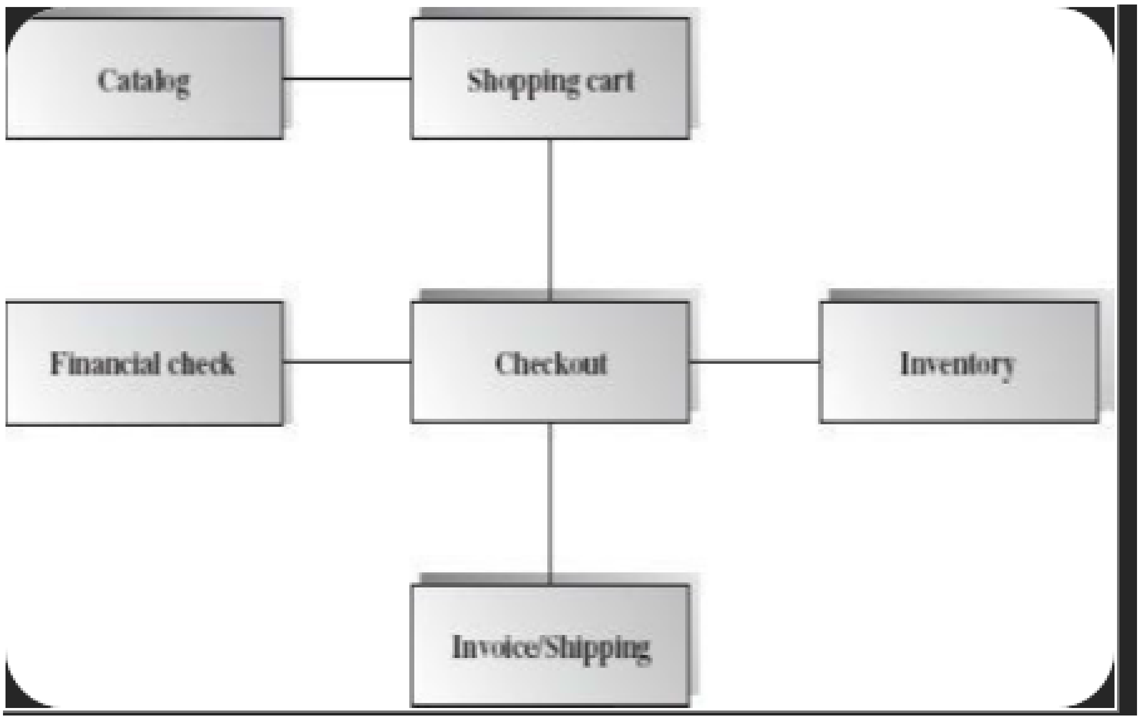

Figure A.6: Box and Line Diagram For Online Shopping Application

A.3.2 Architecture Pattern Example For Online Shopping

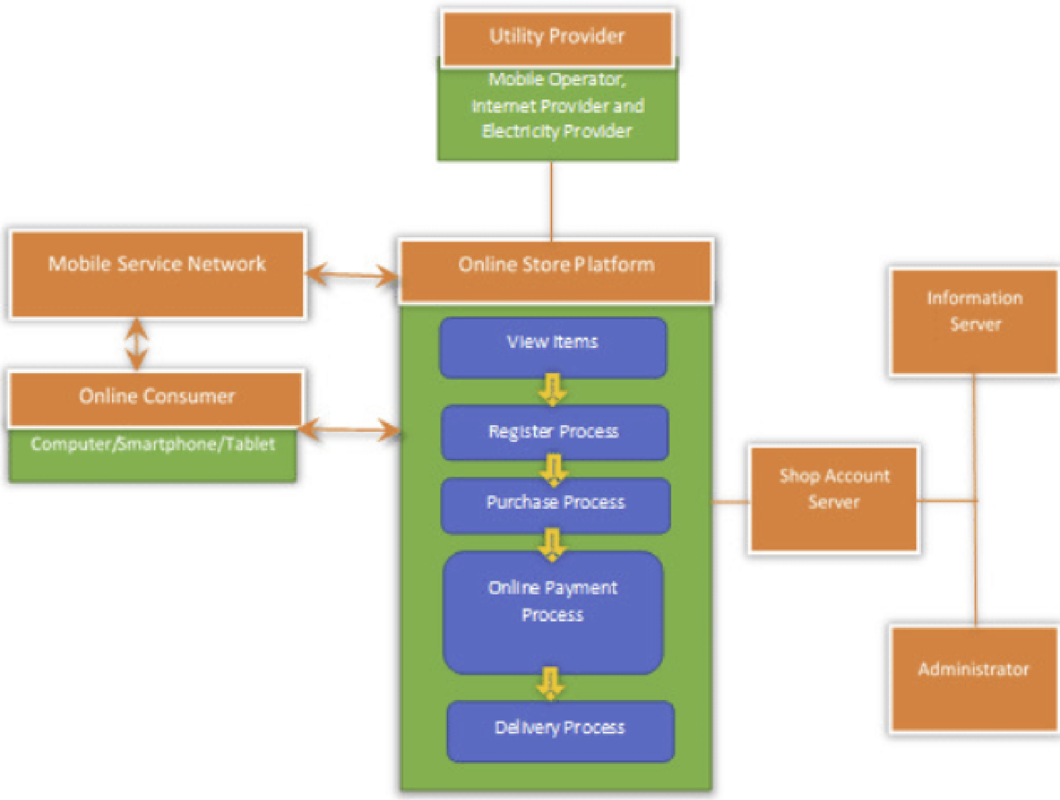

Figure A.7: Architecture Pattern For Online Shopping Application

A.4 Appendix D

A.4.1 Activity Diagram

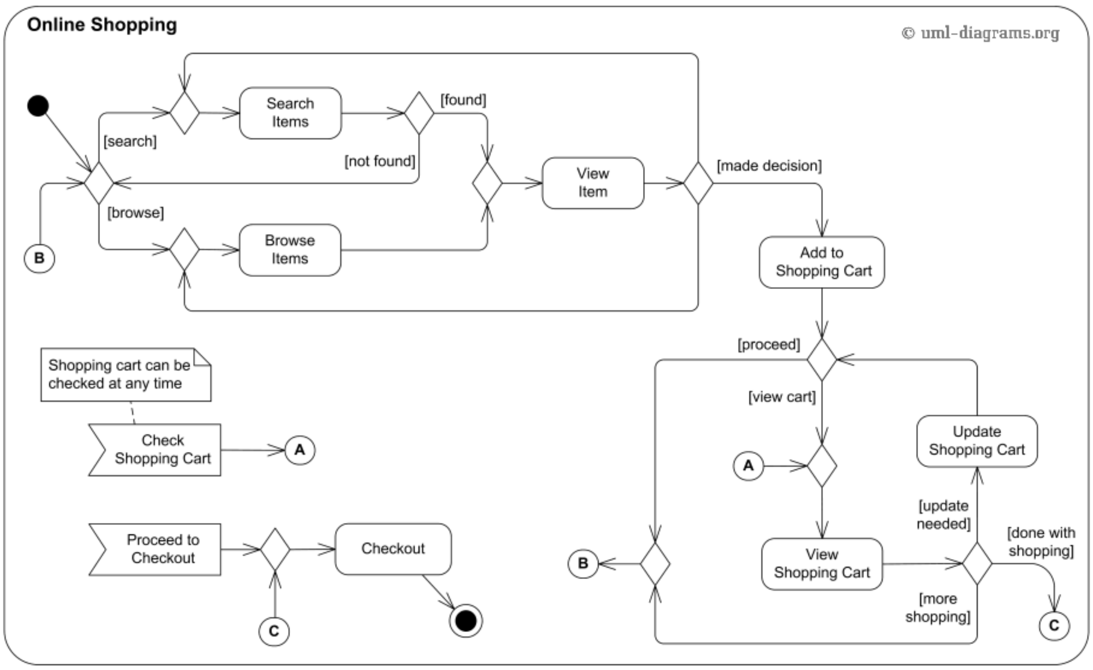

Figure A.8: Activity Diagram For Online Shopping Application

A.4.2 Class Diagram

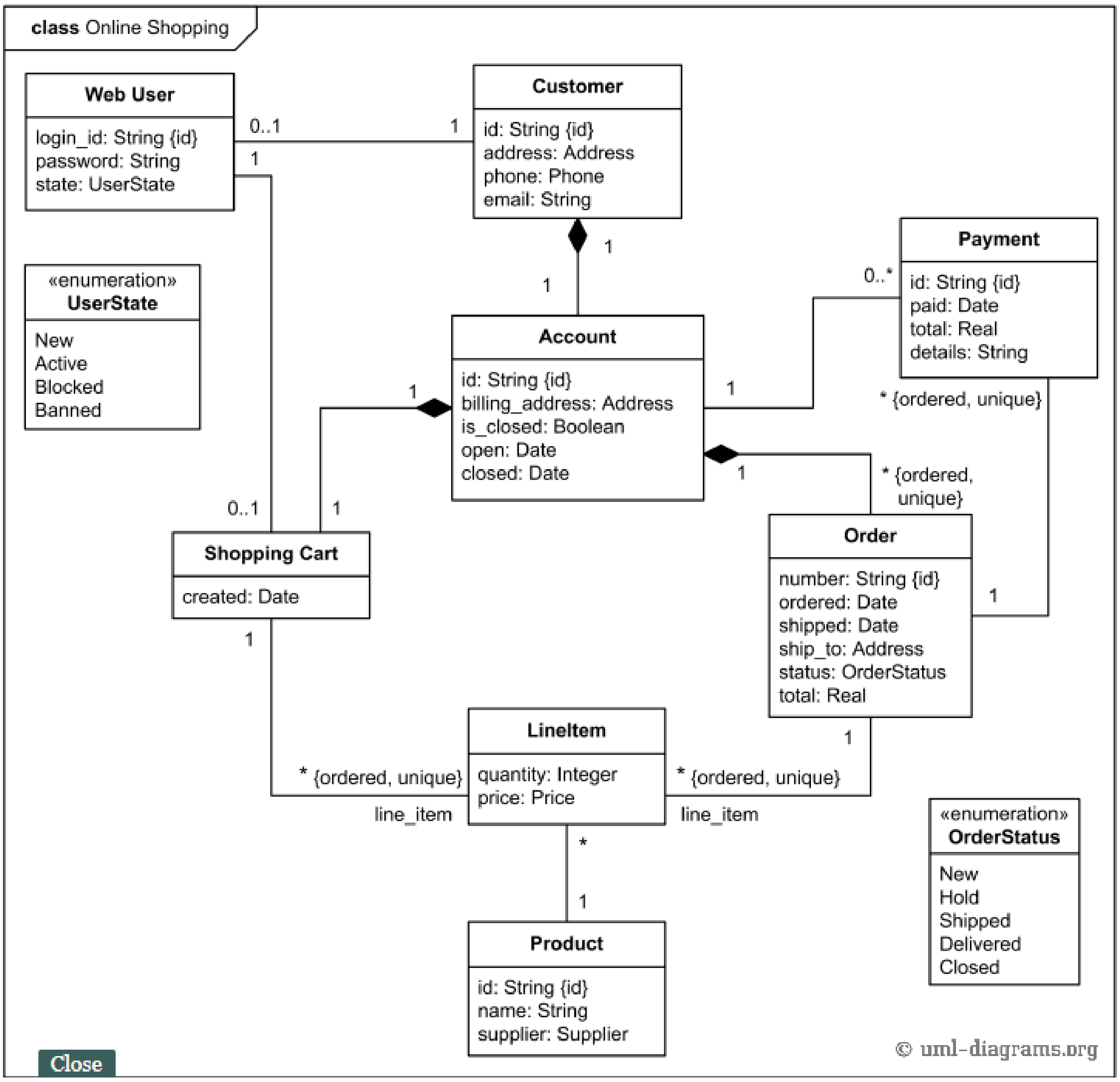

Figure A.9: Class Diagram For Online Shopping Application

A.4.3 Sequence Diagram

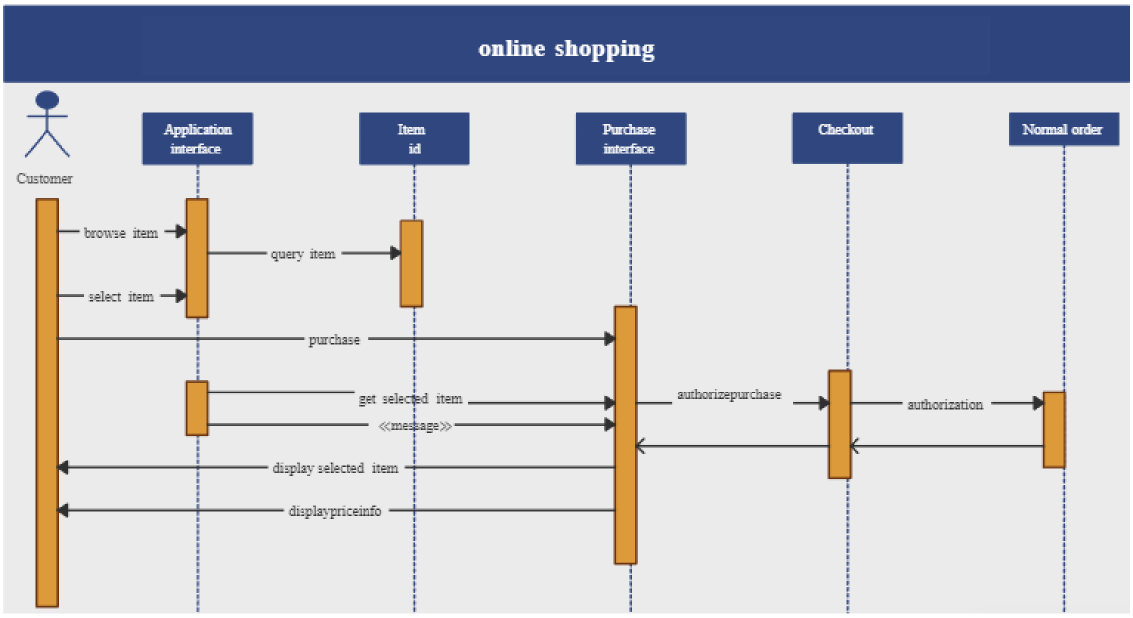

Figure A.10: Sequence Diagram For Online Shopping Application

A.4.4 State Transition Diagram

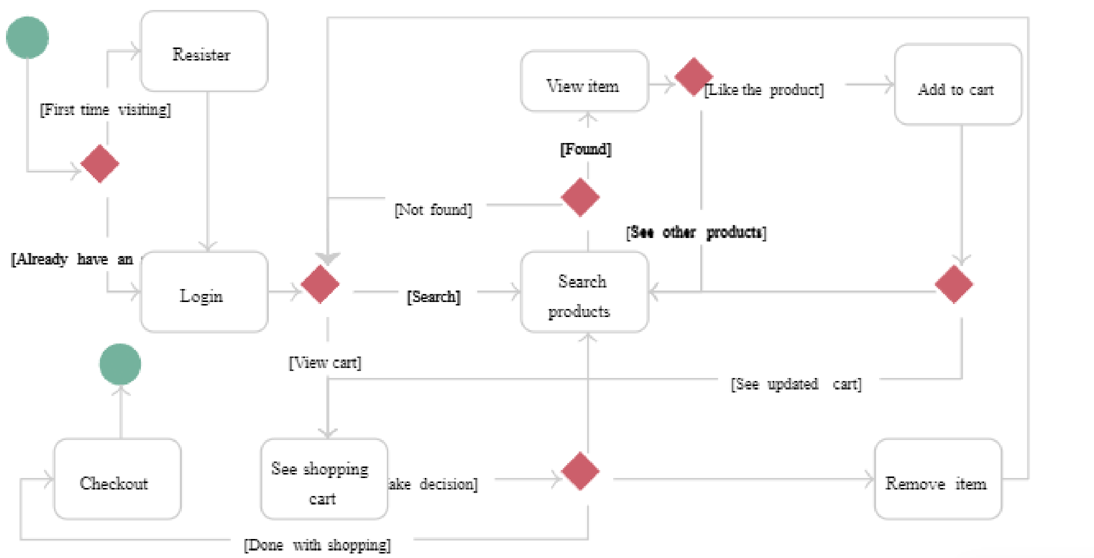

Figure A.11: State Transition Diagram For Online Shopping Application

A.4.5 Data Flow Diagram

Data Flow Diagram

Data flow diagram symbols, symbol names, and examples

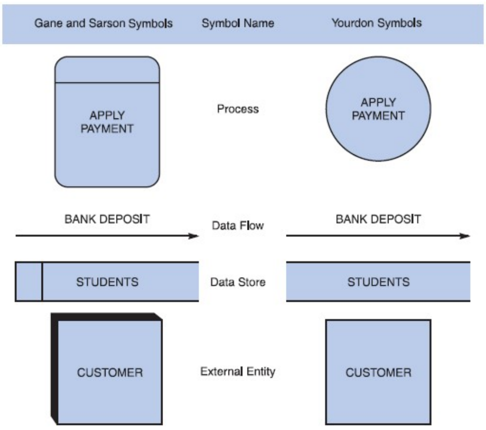

Figure B-5 Data flow diagram symbols, symbol names, and examples of the Gane and Sarson and Yourdon symbol sets.

Guidelines for Drawing DFDs

Step 1: Draw a Context Diagram: The first step in constructing a set of DFDs is to draw a context diagram. A **context diagram** is a top-level view of an information system that shows the system's boundaries and scope. Data stores are not shown in the context diagram because they are contained within the system and remain hidden until more detailed diagrams are created.

Example

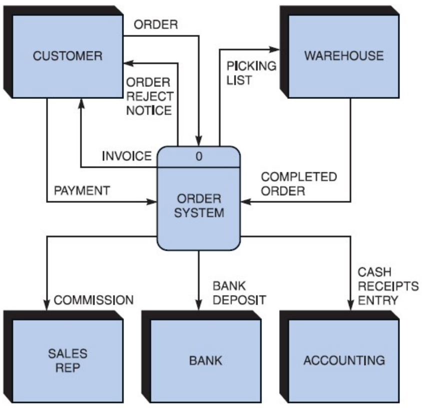

Figure B-6 Context diagram DFD for an order system.

Step 2: Draw a Diagram 0 DFD: To show the detail inside the black box, you create DFD diagram 0. **Diagram 0** zooms in on the system and shows major internal processes, data flows, and data stores. Diagram 0 also repeats the entities and data flows that appear in the context diagram. When you expand the context diagram into DFD diagram 0, you must retain all the connections that flow into and out of process 0.

Example

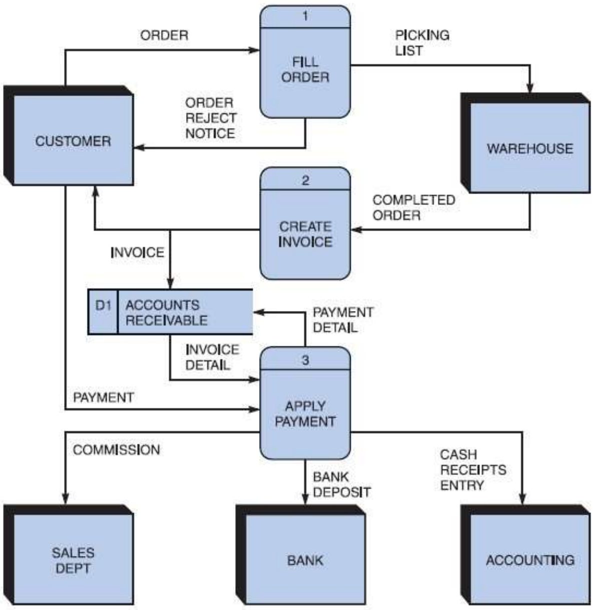

Figure B-7 Diagram 0 DFD for the order system.

Step 3: Draw the Lower-Level Diagrams:

To create lower-level diagrams, you must use leveling and balancing techniques. **Leveling** is the process of drawing a series of increasingly detailed diagrams, until all functional primitives are identified.

Leveling Example

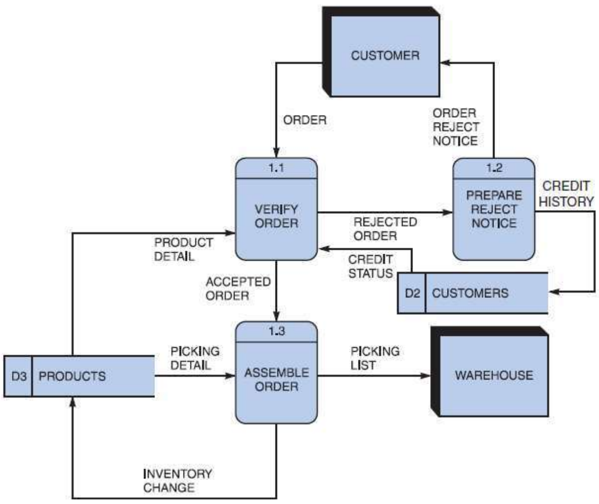

Figure B-8 Diagram 1 DFD shows details of the FILL ORDER process in the order system.