**COMSATS University Islamabad,**

**Abbottabad Campus**

**SOFTWARE REQUIREMENTS SPECIFICATION   
(SRS DOCUMENT)**

**for**

**VideoVigil**  
Version 1.0

***By***

**Mubashir Ahmed CIIT/FA20-BSE-063/ATD**

**Nabeel Ahmad CIIT/FA20-BSE-170/ATD**

**Ghulam Abbass CIIT/FA20-BSE-053/ATD**

***Supervisor*Mam Neeli Khan**

***Bachelor of Science in Computer Science (2020-2024)***

**Table of Contents**

**Revision History 3**

**1.** **Introduction 5**

1.1 Purpose 5

1.2 Scope 5

**2.** **Overall description 5**

2.1 Product perspective 5

2.2 Operating environment 5

2.3 Design and implementation constraints 5

**3.** **Requirement identifying technique 6**

3.1 Use Case Diagram 6

3.2 Use Case Description 6

**4.** **Specific Requirements 8**

4.1 System feature X 8

**5.** **Quality attributes** Error! Bookmark not defined.

5.1 Usability 9

5.2 Performance 9

**6.** **External interface requirements 35**

6.1 User interfaces 35

6.2 Software interfaces 35

6.3 Hardware interfaces 35

6.4 Communications interfaces 35

**7.** **Project Gantt chart 35**

**8.** **References 10**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Application Evaluation History**

|  |  |
| --- | --- |
| **Comments (by committee)**  **\*include the ones given at scope time both in doc and presentation** | **Action Taken** |
|  |  |
|  |  |

**Supervised by**

**<Supervisor’s Name>**

Signature\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Introduction**

This document presents the Software Requirements Specification (SRS) for the VideoVigil Project. Its purpose is to offer a comprehensive delineation of the system's requirements, features, and functionalities. Structured for clarity and practical use, this document is tailored for project stakeholders, including developers, testers, and project managers, to ensure a unified understanding of the VideoVigil system.

**Purpose**

The aim of VideoVigil is to offer a robust and user-friendly solution for enhancing online safety by autonomously detecting instances of violence within videos. Using deep learning technology, VideoVigil generates warnings when potentially distressing content is identified, empowering users to make informed choices about their video consumption. The project's purpose is to contribute to a safer online environment by prioritizing the well-being of internet users through effective content moderation.

**Scope**

The scope of VideoVigil includes essential features for enhancing online safety and content moderation:

* Real-Time Video Analysis: VideoVigil employs deep learning models to analyze videos in real-time, focusing on detecting instances of violence within the content.
* Warning Generation: Upon detecting violence, the system automatically generates warnings to alert users about potentially distressing content.
* Web Browser Extension: VideoVigil is implemented as a web browser extension, seamlessly integrating with popular browsers for widespread accessibility.
* User-Friendly Functionality: The extension provides straightforward functionality, allowing users to manage and customize their experience easily.
* Content Moderation: VideoVigil contributes to content moderation by empowering users to make informed choices about their video consumption.
* Accessibility: The system is designed to be accessible across various internet platforms, promoting a safer online environment for users.

While VideoVigil does not include a graphical user interface, it ensures a focused approach to real-time video analysis and automated warning generation, addressing the critical issue of violence in online content.

**Overall description**

**Product perspective**

VideoVigil is an autonomous web browser extension developed to enhance online safety and content moderation. Operating independently, VideoVigil is not part of an existing product line. It specifically targets the identification of violence or sensitive material within online videos, offering users an automated warning system.

Key Features of VideoVigil:

* **Violence Detection:** Utilizes deep learning algorithms to autonomously identify instances of violence within videos.
* **Automated Warning System**: Generates warnings in real-time, alerting users to potentially distressing content and enabling informed viewing choices.
* **Web Browser Extension:** Positioned as a standalone web browser extension, providing easy accessibility for users across various internet platforms.
* **Comprehensive Content Classification:** Takes a comprehensive approach, considering various video features, such as audio, for accurate content classification.

VideoVigil is designed to contribute significantly to online safety by proactively detecting and flagging potentially harmful content, fostering a secure and considerate online environment for users.

**Operating environment**

VideoVigil is designed to provide compatibility across all major web browsers, ensuring users can seamlessly integrate the extension into their preferred browser environment. The supported browsers include but are not limited to:

* Google Chrome
* Mozilla Firefox
* Microsoft Edge
* And more

VideoVigil is platform-agnostic, making it compatible with various operating systems commonly used by individuals worldwide.

**Design and implementation constraints**

**CI-1:** The system shall utilize the Python programming language for its implementation. This constraint is based on the expertise of the development team, aligning with their proficiency in Python and its suitability for learning applications.

**CI-2:** VideoVigil will employ TensorFlow as the primary learning library for model development and real-time video analysis. This constraint is motivated by TensorFlow's established reputation for deep learning tasks and its extensive community support.

**CI-3:** The system shall integrate with major web browsers through standard web technologies, ensuring compatibility with HTML, CSS, and JavaScript. This constraint is chosen to facilitate seamless integration into various browser environments without imposing specific browser dependencies.

**CI-4:** VideoVigil's image enhancement module (Module 4) may be subject to GPU capabilities for optimized performance. This constraint acknowledges the potential dependence on GPU resources for certain image enhancement algorithms, providing enhanced image quality.

**CI-5:** The development of the web browser extension (Module 5) shall adhere to the guidelines and specifications of popular browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple Safari. This constraint ensures a consistent and reliable user experience across diverse browser platforms.

These constraints are established to streamline the development process, leverage existing expertise, and ensure interoperability with widely adopted technologies, contributing to the overall efficiency and effectiveness of VideoVigil.

**Requirement identifying technique.**

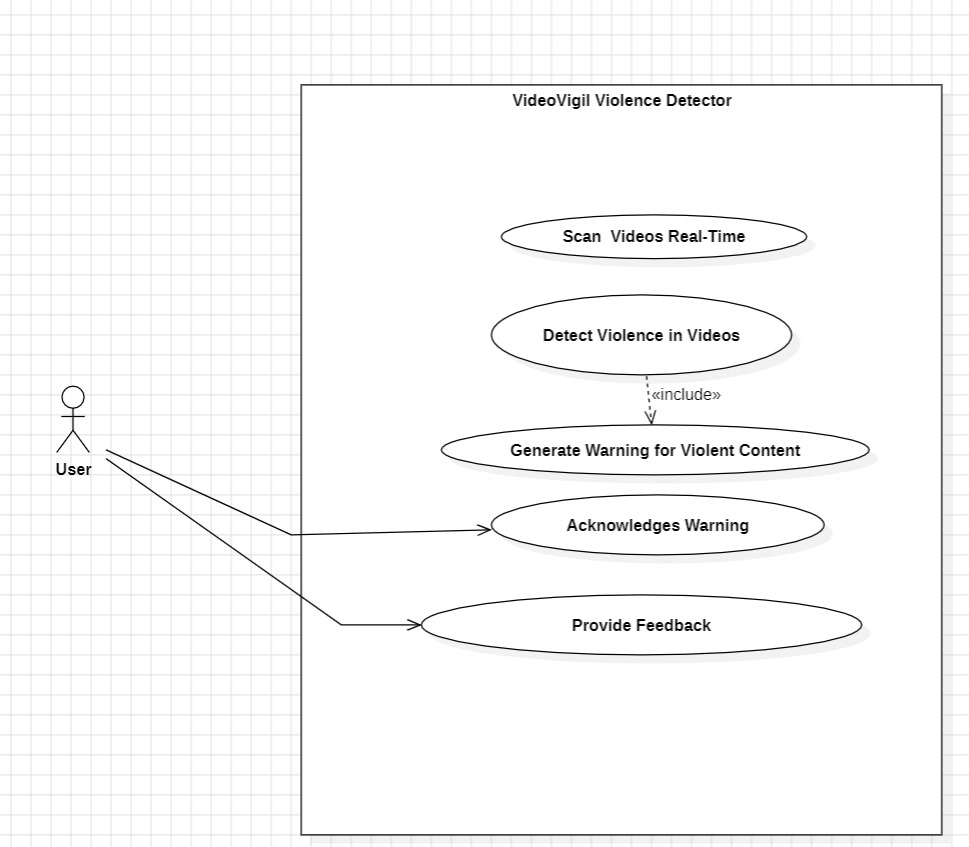
The following storyboard outlines the fundamental user flow for VideoVigil:

1. The user launches the VideoVigil application and is presented with the main menu.
2. From the main menu, the user has the option to choose from the following functionalities:

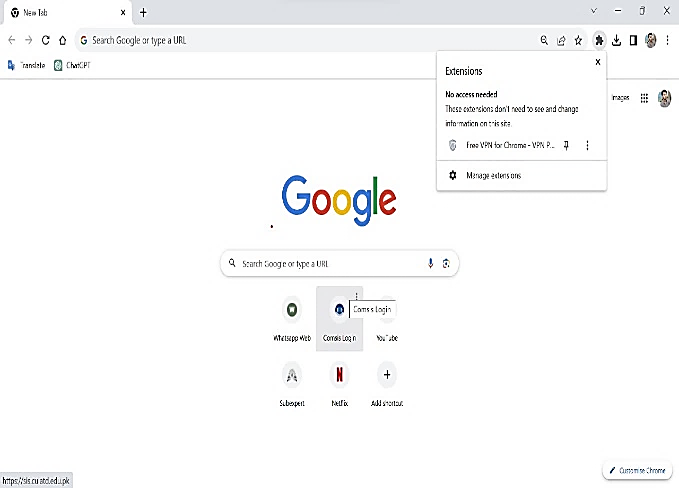
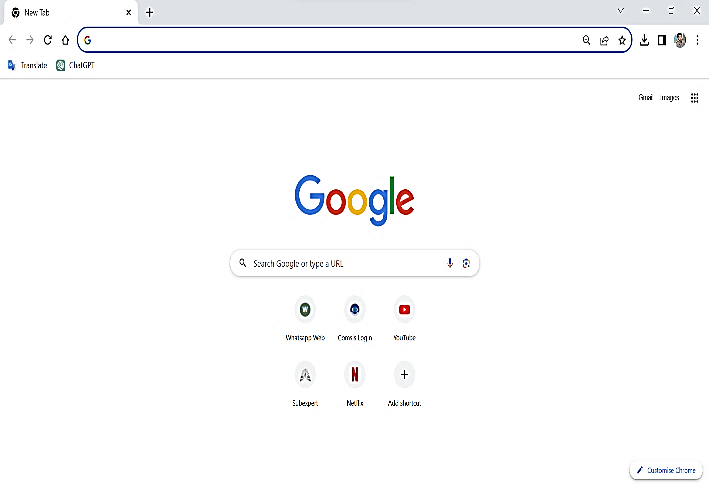
* Video scanning: The user initiates the video scan, allowing the app to analyze the content for potential violence or sensitive material.
* Violence detection: The user activates the violence detection feature, enabling the app to identify instances of violence within the scanned video.
* Warning generation: If violence is detected, the app generates a warning message, alerting the user to the potentially distressing content.
* User acknowledgment: The user acknowledges the warning and takes appropriate actions, such as stopping the video or reporting it.

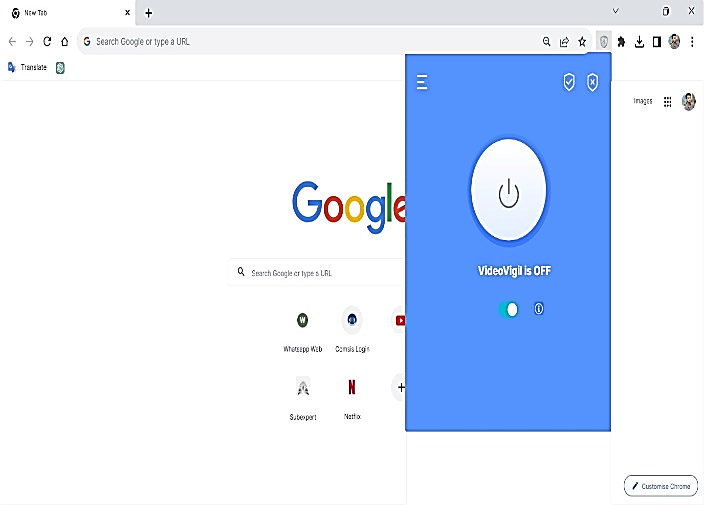
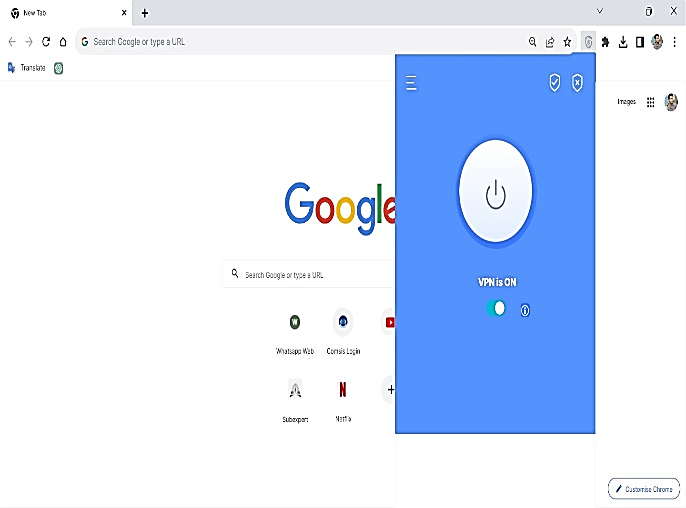
1. The user selects and executes the desired operation, and the app provides real-time feedback.
2. The user can iterate through steps 2 and 3 until they have completed their usage of the VideoVigil application.

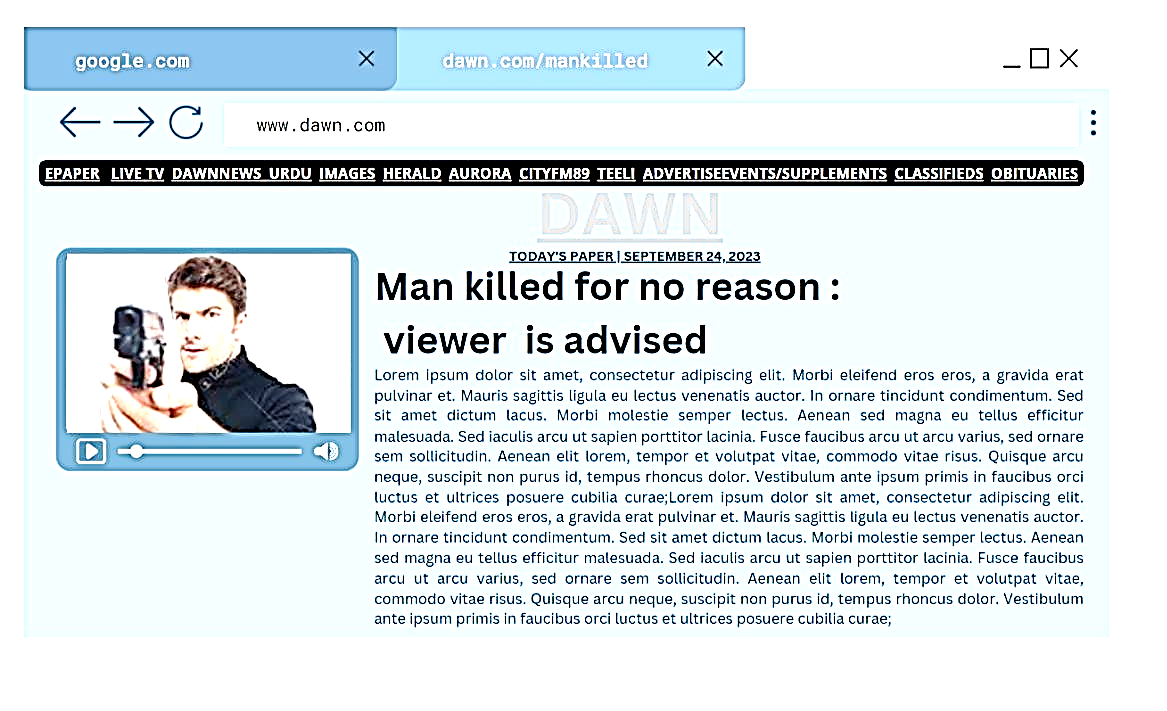
This storyboard illustrates the basic sequence of user interactions, emphasizing the key functionalities offered by VideoVigil for video scanning, violence detection, warning generation, and user acknowledgment.

**Use case diagra****m**

**Story Boards:**

****

****

****

****

****

**Functional Requirements**

* **Feature: Video Analysis and Detection**

**FR\_1:** The extension shall have the ability to analyze online videos in real-time.

**FR\_2:** It shall implement a violence detection mechanism using a deep learning model.

**FR\_3:** The initial human detection module, utilizing Faster R-CNN Inception V2 COCO, must seamlessly integrate with the violence detection process.

* **Feature: Warning Message Generation**

**FR\_4:** Upon detecting violent content in a video, the extension shall generate warning messages.

**FR\_5:** Warning messages should be clear, concise, and informative, advising users about potentially distressing content.

* **Feature: Real-Time Processing**

**FR\_6:** The extension shall perform video analysis and violence detection in real-time to provide timely warnings to users.

**FR\_7:** Real-time processing should be optimized for performance and responsiveness.

Feature: Model Training and Updating

**FR\_12:** The extension should support model training to improve the accuracy of violence detection.

**FR\_13:** It shall provide a mechanism for updating the trained model with new datasets.

Feature: Privacy and Security

**FR\_14:** The extension must prioritize user privacy and adhere to all relevant data protection regulations.

**FR\_15:** Video analysis should be conducted locally on the user's device whenever possible to minimize privacy concerns.

* **Feature: Responsiveness**

**FR\_18:** The extension should be responsive to different video formats and resolutions.

**FR\_19:** It should adapt to various online platforms to ensure broad compatibility.

**Non Functional Requirements**

**Usability**

USE-1: The VideoVigil web browser extension must feature a user-friendly interface, ensuring ease of navigation and comprehension, particularly for users with limited technical expertise.

USE-2: Clear and concise instructions for each feature must be provided within the extension, ensuring an intuitive user experience, and enabling users to perform tasks effortlessly while accessing information.

USE-3: VideoVigil must prioritize accessibility, ensuring users with visual impairments, hearing impairments, and mobility impairments can interact effectively with the web browser extension.

**Performance**

PER-1: VideoVigil must be optimized for swift loading and execution, ensuring a seamless user experience.

PER-2: The extension must handle a high volume of requests efficiently, preventing significant delays even during peak usage periods.

PER-3: VideoVigil should efficiently handle large datasets, such as video libraries, for robust performance in video scanning and analysis.

**Scalibility**

SCA-1: VideoVigil must be scalable to support a large user base, handling a significant number of requests without compromising performance.

SCA-2: The extension must efficiently handle large datasets, ensuring optimal loading and processing capabilities for videos sourced from memory or a network.

**References**

https://medium.com/@madhawavidanapathirana/real-time-human-detection-in-computer-vision-part-2-c7eda27115c6