Video Manipulation Detector

Author

PC Hamza Pervez

PC Kamran Afzal

PC Muhammad Nauman

PC Muhammad Ammar Akram

Supervisor

Dr. Naima Altaf

Submitted to the faculty of Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, in partial fulfillment for the requirements of B.E Degree in Computer Software Engineering.

(July), 2020

CERTIFICATE FOR APPROVAL AND CORRECTNESS

This is to officially state that the thesis work contained in this report "Video Manipulation Detector" Is carried out by PC Hamza Parvez, PC Muhammad Nauman, PC Kamran Afzal and PC Muhammad Ammar Akram under my supervision and that in my judgment, it is fully ample, in scope and excellence, for the degree of Bachelor of Software Engineering from Military College of Signals, National University of Sciences and Technology (NUST). And is original with 14 % of plagiarism.

Approved By

Dr. Naima Altaf

Department of CSE, MCS

Dated:

_____

Department of CSE, MCS

Dated:

_____

DECLARATION OF ORIGINALITY

We hereby declare that the work contained in this report and the intellectual content of this report are the product of our work. This thesis report has not been formerly published in any structure nor does it include any verbatim of the published resources which could be treated as violation of the international copyright decree. We also affirm that we do recognize the terms 'plagiarism' and 'copyright' and that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.

ACKNOWLEDGEMENTS

In every work, it is impossible to achieve success without the will of Almighty Allah. He granted us the motivation and drive to ensure our project reaches fruition. Whatever we achieve, it is thanks to his blessings. We are also grateful to our parents for their unconditional support. Above all we are eternally thankful to Dr. Naima Altaf. It was because of her valuable suggestion, her to the point inputs, her patience and endurance and her decision to give us autonomy over the project implementation, due to which we were never under pressure for deliverables and had a fun learning experience while at the same time we will be remorseful over the fact the our project does not reflect the height it should have achieved under her supervision.

Copyright Statement

DEDICATION

To our parents, teachers and friends who gave their love and support. Without their help, this project would not have come to fruition. A special thanks to our supervisor Dr. Naima Altaf whose valuable suggestions helped us throughout the project implementation.

Abstract

Technology has become an essential part of our life. It has been the case since time immemorial that man tried to create something to ease his life. Forgery detection is one such avenue of technology where advancements are being made on a daily basis and realities are being unveiled to seek truth.

Video manipulation detection tools are one such production of the field of detection systems and they have displayed their application from sensitive institutions to common institutions. They have aided in detecting forgeries in videos of every field to unveil facts. Their major utility have been in crime investigating institutions like courts, NAB, FIA, intelligence agencies and many other national and international institutions. This technology is not becoming obsolete anytime soon, so further innovations in this technology and its principles will only be for the better.

Video manipulation detection have now gotten entire Software Development kits for them where on dependable hardware, one can deploy a software control application for them with open source software, API's.

Our main purpose is to design a mechanism to allow Videos to be detected by a Web based application. A web based application will be designed to allow the conceptual idea of the project to be realized.

Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Overview

Video manipulation detectors can be a great utility of their era. They can aid relevant institutions a great deal. Also they can be toned down to be usable at a domestic level. There are major companies and institutions that must have their own VMDs. But unfortunately, there is no such tool found to detect manipulations made in videos at commercial level. We could scarcely found some documental work and research papers published by individuals but not any tool. So, it's a great triumph to have our own VMD with maximum efficiency and security. In order to use it, user will have to first create an account and sign in to this web application. After that he can upload video that will be processed by server applying model. After complete processing of video at server side, the results will be displayed to the user.

1.2. Project Specific VMD

The VMD for our project is a web application. It is independent of any specific hardware modules. Firstly the user will log in to this web application. If he doesn't have an account already, he will need to create it to proceed further.

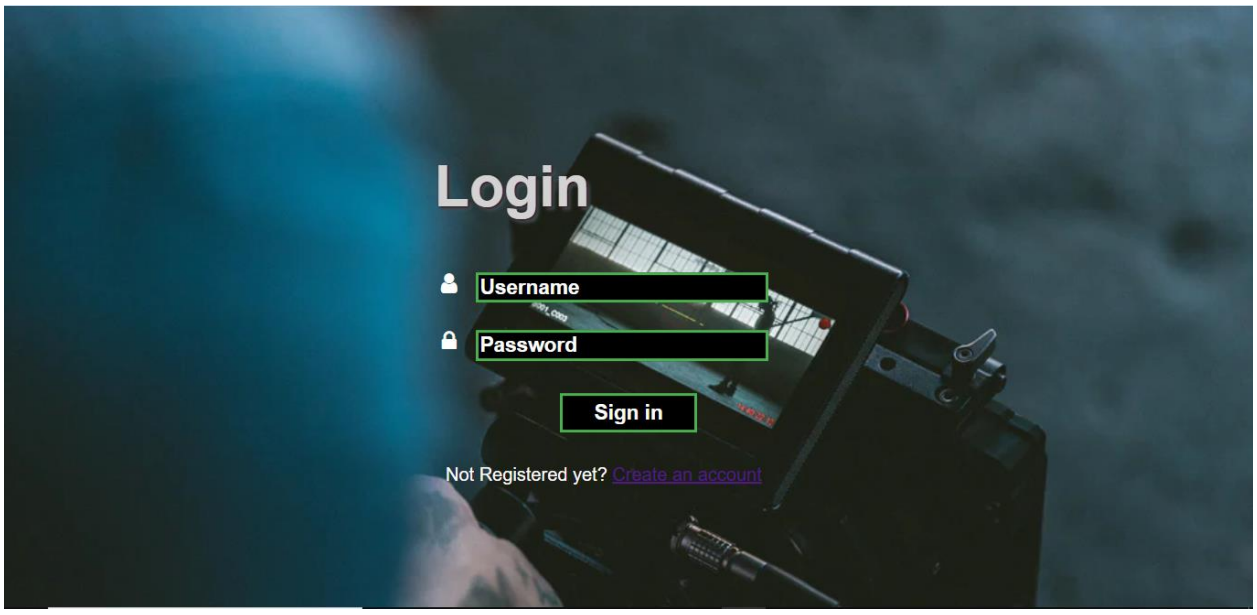


Figure 1-1: Login page interface.

In case he needs to create account, he will go to sign up page:

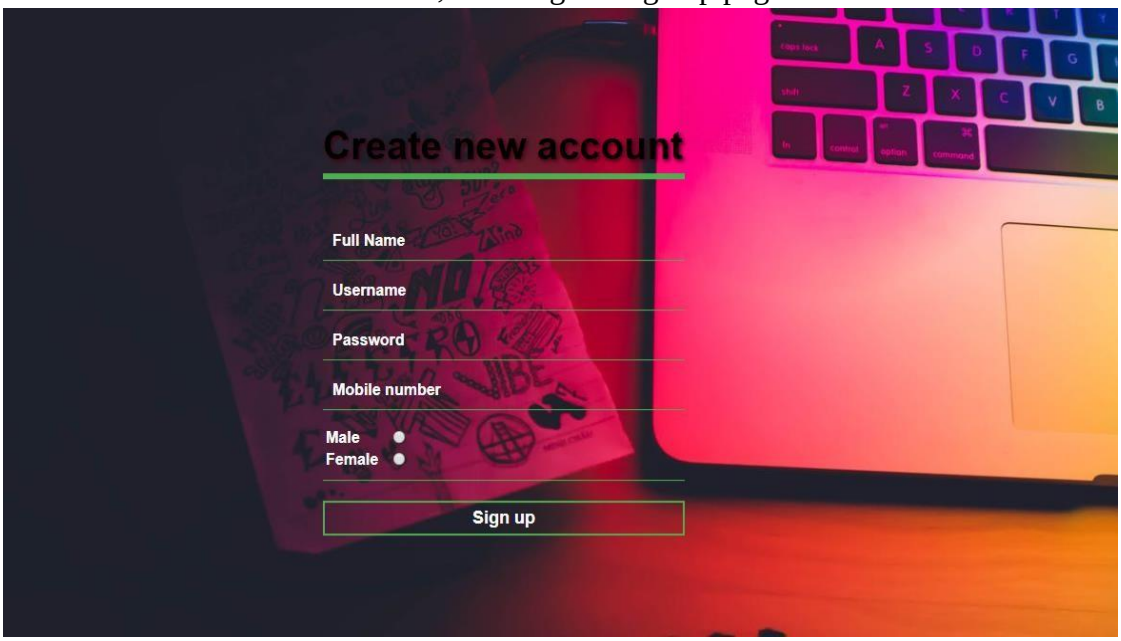


Figure 1-2: Sign up page.

After the user logs in his credentials, he is prompted to upload video of specific size (25mb) and quality to see if it's forged or not. Video uploading page looks like this:

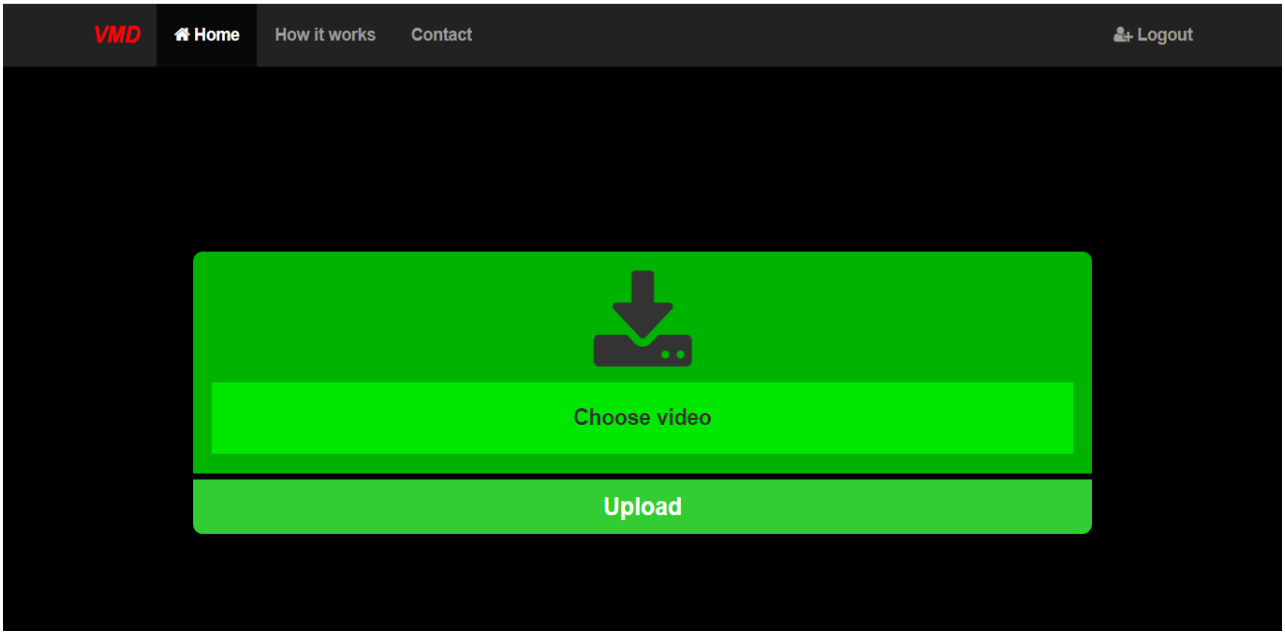


Figure 1-3: Video uploading page.

After that, server processes the video by applying its model. This way the forged video frames are detected and the result is displayed on screen:

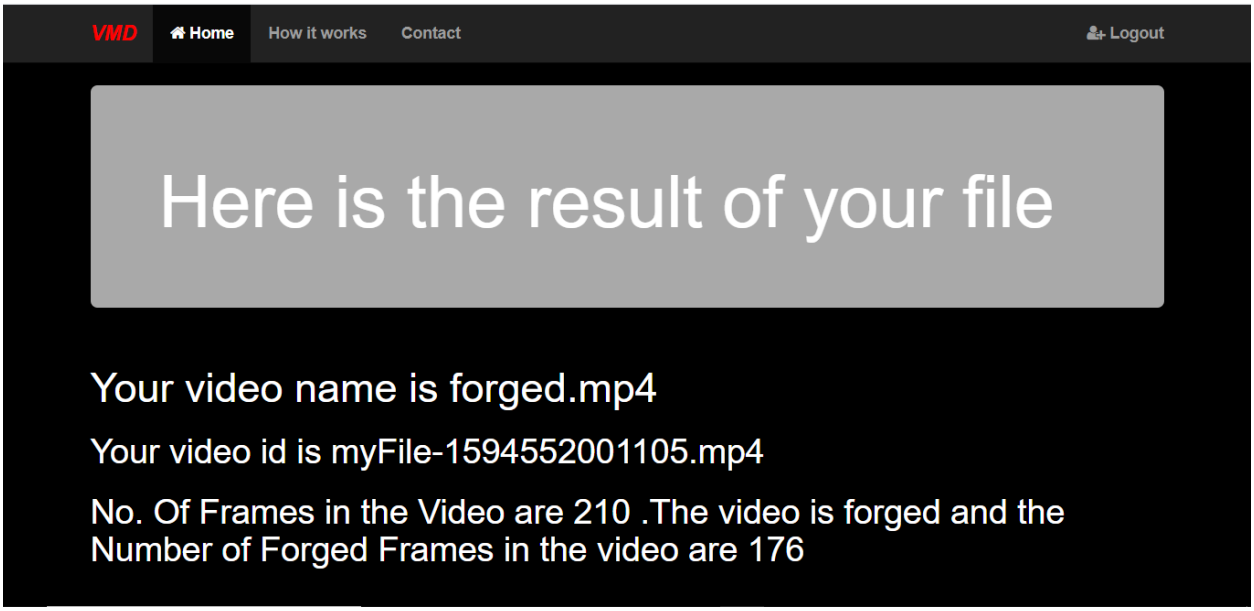


Figure 1-3: Results.

1.3. Problem Statement

To make a web-based application that takes video as an input and passes it through forgery detecting algorithm/model to see if the video is forged or not. The output is both qualitative and quantitative form. Output shows how many and which of the video frames have been manipulated. This way user can get assured about the authenticity of the video.

1.4. Approach

For the purpose of our project we will develop a web-based application named 'Video manipulation Detector'. This will be web-based where relevant Open API's will be used to realize our conceptual idea.

1.5. Scope

With the Help of this web application, a person will be able to detect the forged videos. It will help people as well as organizations (like courts, police, FIA, ISI) to detect manipulated videos.

1.6. Objectives

The main objective of this project is to build a web application that takes video from user as an input and then using server model it processes video to check authenticity of video by detecting if it's forged or not.

During the different phases of this project, any and every technique and theory we have learned during our course, we have considered, used and covered. In strict terms of software engineering we have covered: feasibility analysis, requirement gathering, architectural and detailed design, implementation and testing along with documentation (SRS, SDS, Test Document, final report). Students are also expected to develop extensive knowledge and technical skills in the following fields:

1. Web Programming using different frontend and backend languages
2. Working with open source software.

1.7. Deliverables

Sr.	**Tasks**	**Deliverables**
1	Literature Review	Literature Survey
2	Requirements Specification	Software Requirements Specification document (SRS)
3	Detailed Design	Software Design Specification document (SDS)
4	Implementation	Project demonstration
5	Testing	Evaluation plan and test document
6	Training	Deployment plan
7	Deployment	Complete application with necessary documentation

Table 1-1 Deliverables

1.8. Overview of the Document

This document shows the complete working implementation of ours project. It starts off with the literature review which shows previous work done of similar nature and how we are working to change or improve them, requirement analysis of the system, system architecture which highlights the modules of the software and how they interact with one another and represents the system in the form of component diagram, Use Case Diagram, Sequence Diagram and general design of the system. Furthermore, it will discuss the detailed purpose of every components involved. Further the dependencies of the system and its relationship with other products and the capacity of it to be reused will be discussed. At the end test cases and any future work recommendation or improvements have been presented.

1.9. Document Conventions

Heading are written numerically with the main heading having a single digit and any sub-headings below have more than one digits separated by a decimal point. The Font used to write is Times New Roman. All the main headings are of size 18 and bold. All the second level sub-headings are of size 16 and bold. All the further sub-headings are of size 14 and bold. Any reference relevant to the project has been disclosed at the end and any technical or ambiguous terms used have been explained in a glossary.

1.10. Intended Audience

This document is intended for:

1. **Developers:** (Project Group)

   They are the people working on the project. They are the people who possess technical knowledge of the concepts and skills required in this project and its domain.

2. **Testers:** (Project Group, Supervisor)

   They are the people who view the project and test out the functional aspects of the project to ensure that the project conforms to the user requirements.

3. **Users:**

   The operators of the project. This document will help familiarize themselves with the inner working and mechanics behind this project and help them troubleshoot a problem if any.

4. **Documentation writers:** (Project Group)

   To know what features are present and explain them. How they behave in certain conditions. What can cause them to fail and how to troubleshoot it.

5. **Project Supervisor:** (Dr. Naima Altaf)

   This document shall be used by the supervisor so that they can ascertain that all requirements have been met in the most apt way possible.

6. **Project Evaluators:** (CSE Dept. MCS)

   So that they may know the scope and working of the project and help them in deciding how to grade our work.

Chapter 2. Literature Review

The detail of the study for this project and projects related to our own shall be discussed. With daily improvements going on and on in the field of VMDs, many useful projects have been made that have been extremely beneficial to their use cases.

Initially we studied what VMDs are, what they are meant for. Where are the uses of VMDs? Our project specific being a web application, we studied basics and working principles of this specific type of a detectors.

Furthermore, we studied different kinds of such documents near our requirements that is web based detector. This works on the principle of taking a video as an input and the model processes the video through its modules and detects if there is any manipulated made between frames.

Generally, video forensics tools are made available for detecting videos with very limited specifications. Besides this, the results are not satisfactory, really time taking and unfortunately they do not offer a great deal of timely efficiency.

In Pakistan, we can scarcely find a video forensics tool for general use. These are only limited to the relevant institutions like investigating and surveillance institution, courts, FIA and Army etc. Anyhow, these tools became the motivation behind our work and we decided to implement this behavior but using a web application. Given the fact that these tools were customizable.

Additionally, this customizability comes at a price. Forgery tools, just like other commercial forensic tools contained security flaws and were susceptible to denial of Service, buffer overflow or ARP cache poisoning attacks.

For this purpose, we decided to change things. Look towards more viable solutions. We used an open API software framework for VMDs. This allowed us the freedom to design what we like and how we like it. Having a Desktop based Ground Station, it also has a web SDK allowing us to design an application suited for our use case. With abundant help, to the point documentation and available of good open source examples. This was the path to take.

Chapter 3. Software Requirements Specification (SRS)

3.1 Introduction

3.1.1. Purpose

The document provided purpose is to describe briefly yet detailed description of a web based platform provided to detect video manipulation or forgery in videos. The document is going to explain it to you about the features and purpose of the solution provided to the problem statement. The document also explaining the constraints under which the solution is going to operate. This document provided is intended for users and developers.

3.1.2. Project Vision

With the Help of this web app, a person will be able to detect whether his video is manipulated or not.

For	Everyone with an account on VMD
What	A web application for detecting manipulation of videos.
Is	An web application
That	Provides a platform for the detection of forgeries made in videos.

Table 3-1 Project Vision

3.2. Overall Description

3.2.1. Product Perspective

This application will help save time and cost and mainly help reveal facts. A person sitting at his home can simply go to this web app, create an account on it and upload his videos to find if his videos is forged or not. He doesn't need to run after forensics teams or other relevant institutions for this.

3.2.2. Product Features:

This web application is taking the field to reveal facts about the authenticity of videos

1. A Login/signup interface.
2. A main page interface.
3. Results page interface

.

3.2.3. User Classes and Characteristics

Following are user classes and their brief description.

The User

The person with technical and operational knowledge of the web apps, not necessarily the detailed version, who will operate the project.

Tester (occasional user)

The testers are the individual with technical, and project related knowledge who while keeping the requirements of the project under consideration test out the functional aspects of the project to ensure that they conform to the requirement specification

Developers

The developers are the individuals with technical, conceptual and project specific knowledge who shall use this knowledge to realize the project concept within the requirement specification.

Documentation Writers

The people who have the conceptual or technical knowledge about the project who will write about the detail of the project like requirements, detailed design, test cases etc. This will serve as a basis for any future developer or document writer.

3.2.4. Operating environment

The final product shall be operating in a web environment. It shall be compatible with all the higher versions of web.

Hardware

1. GPU
2. CPU (intel core i7 processor)

Software

1. Learning Framework:- Keras + Tensorflow
2. IDE Anaconda (Python 3.6)
3. Deep:- Jupyter Notebook
4. Python Libraries :- numpy, sklearn, opencv-python, keras, tensorflow-gpu, pillow
5. For Application Dashboard:- Tkinter(Python) + NodeJS

3.2. 5. Model

The model which we are using is based on ResNet 50 architecture provided by Keras. ResNet-50 convolutional neural network is a smaller version of ResNet-152 that is 50 layers deep. A Convolutional Neural Network is a special kind of multi-layer neural networks, designed to recognize visual patterns directly from pixel images with minimal preprocessing.

3.2.6. How model works?

Before the training of the model we converted the videos into NumPy data, Xtrain and Ytrain respectively. In the training of model we give the path of Xtrain.npy and Ytrain.npy files. The model process these files and produce checkpoints. The path of these check points will be given in the predict_forgery.py file in order to detect forgery in the video.For now our model will detect those video having resolution of 320*240 pixels and are in mp4 format.

3.2.7. Design and Implementation Constraints

1- Due to complex model training, the model might not be precise to every frame.
2- Internet may cause latency in command transmission to web app. So, at all time a good working connection is required.

3- Also Because of file size the web app might not behave like how it has to perform.

3.2.8. User Documentation

To allow the user to detect manipulated videos using the web app. As of this version of the SRS, a detailed User documentation is yet to be made.

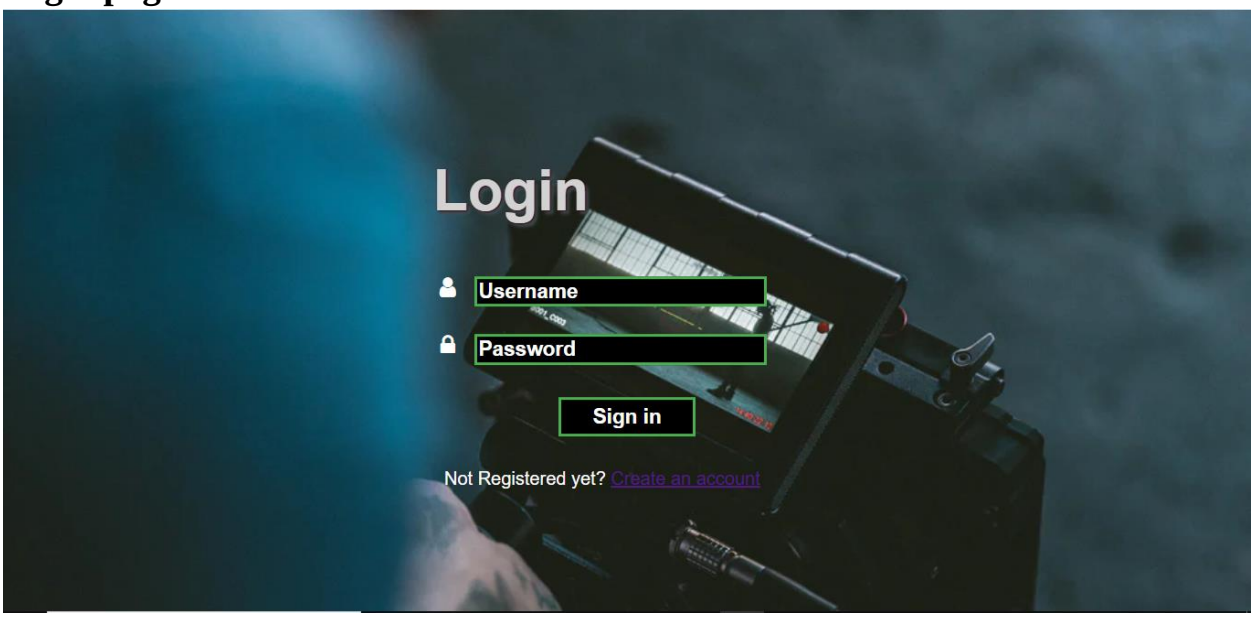
3.2.9. Assumptions and Dependencies

It is assumed that the person has enough knowledge and technical skills to use the web browser and quite familiar with the browser interface. Additionally, it is also assumed that the frames of uploaded video are forged.

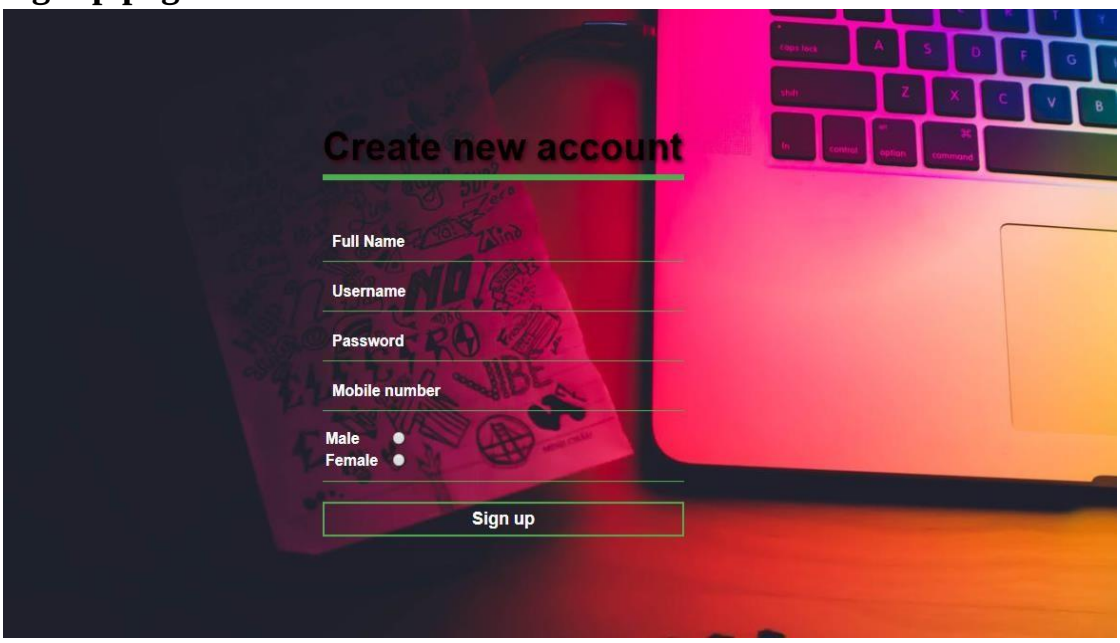
Interfaces Requirements

3.3.1. User Interfaces

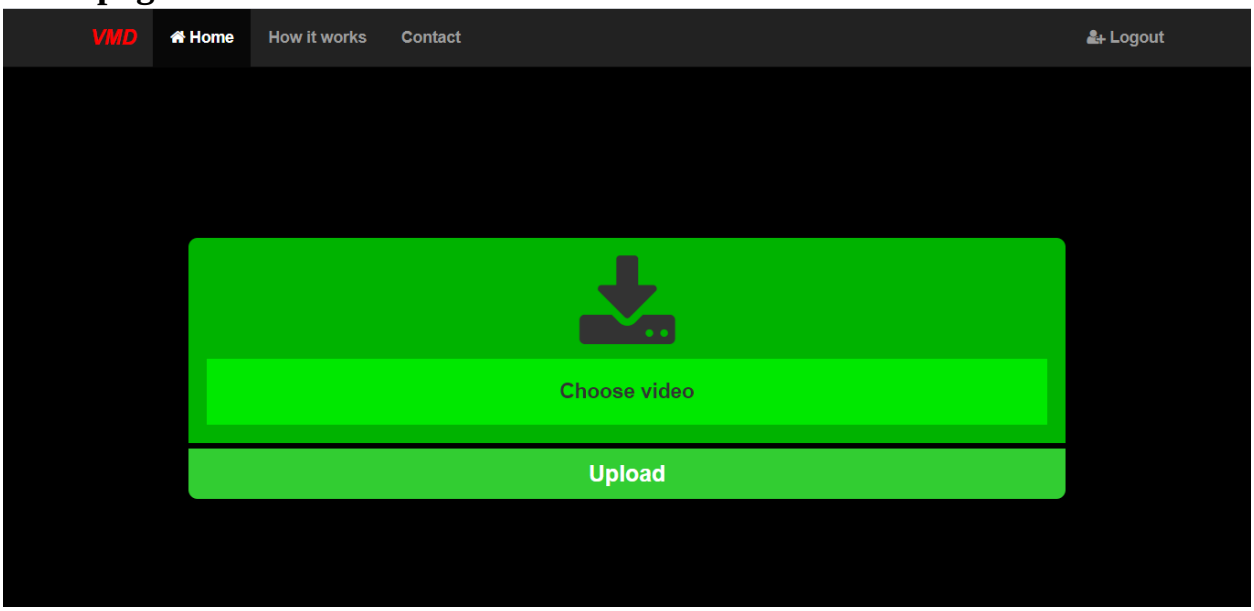
Login page:



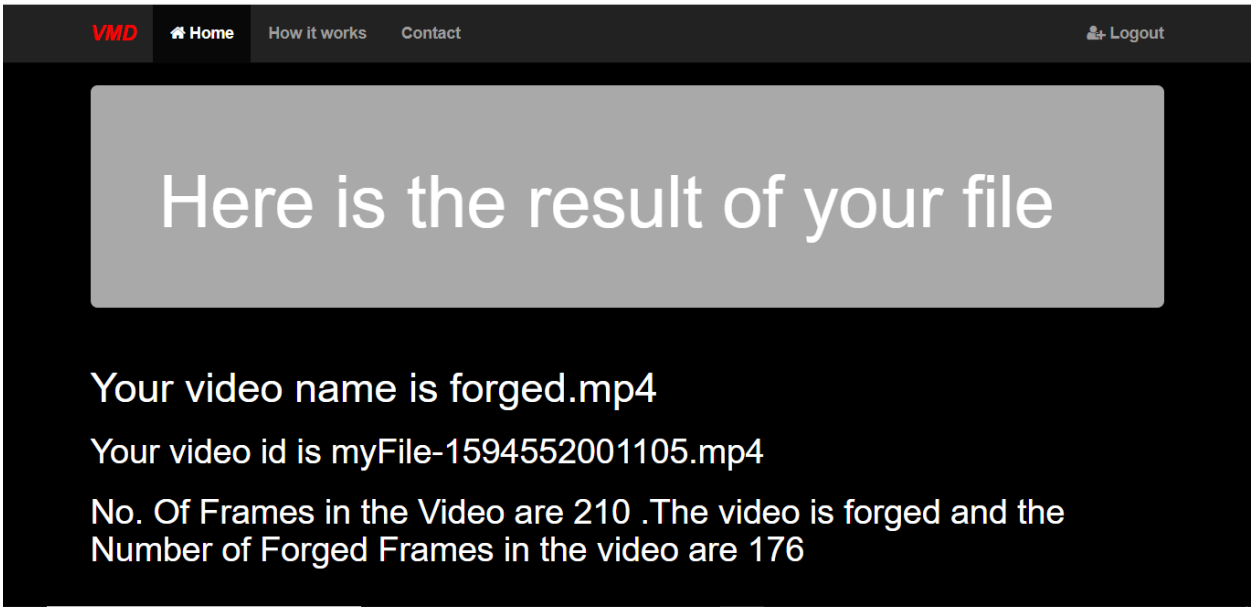
Signup page:



Main page:



Results:



3.3.2. Software Interfaces

The UI of the software is going to be compatible to any browser so that user feel comfortable to access to the software. The browser we see in our daily life such as Internet Explorer, Mozilla or Google all are compatible with the software.

3.3. System Features

This section illustrates organizing the functional requirements for the project VMD by system features: -

1. User Authentication
2. Upload video
3. Detect forged videos
4. Display result

3.3.1. User Authentication

Description

This feature will check that the current user who is trying to access the main page is registered or not.

Stimulus/Response sequence

Normal Path: User get access to main page
Preconditions • User entered a valid username and password
Interactions • A green signal is raised by the database
Post conditions • Main page is displayed
Categorization • **Criticality:** High • **Probability of Defects**: Low • **Risk:** High

Table 3-2 Authentication

Exceptional Path: Error occurred
Preconditions • User entered an invalid username or password
Interactions • An error signal is raised by the database
Post conditions • A notification appears
Categorization • **Criticality:** High • **Probability of Defects**: Low • **Risk:** High

Table 3-3 Authentication Error

Functional requirements
The system shall give access to only registered users.

3.3.2. Upload video
User uploads his video to be detected for forgery.

Description

Through this feature user can upload his video

Stimulus/Request Sequences

Normal Path: Video is uploaded successfully
Preconditions • Video is not corrupted, and its size is not greater than 25MB
Post conditions • Video is not corrupted, and its size is not greater than 25MB
Categorization • **Criticality:** Medium • **Probability of Defects**: Medium • **Risk:** Medium

Table 3-4 Video Upload

Functional Requirement

1. System shall be able to detect defects in videos.
2. System should not upload videos having size greater than 25MB.

3.3.3 Detect forged videos

Description

This feature enables the system to detect inter-frame manipulated videos uploaded by the user.

Stimulus/Response Sequence

Normal Path: Processing uploaded video
Preconditions • Uploaded video is not corrupted.
Interactions • Uploaded video is sent to the system for processing.
Post conditions • System will check that the uploaded video is manipulated or not.
Categorization • **Criticality:** High • **Probability of Defects**: High • **Risk:** High

Table 3-5 Detect Forged Video

Exceptional Path: Error occurred
Preconditions • Server become down
Post conditions • Video will be uploaded again
Categorization • **Criticality:** High • **Probability of Defects:** Low • **Risk:** High

Table 3-6 Detection Error

Functional requirements

System shall be able to detect that the video uploaded by the user is manipulated or not

3.3.4 Display results

Description

Through this feature result of the video uploaded by the user is displayed.

Sequence/Response Sequences

Normal Path: Result is displayed
Preconditions • Result is generated
Post conditions • The user will be able to see result
Categorization • **Criticality:** High • **Probability of Defects:** Medium • **Risk:** High

Table 3-7 Result Display

Exceptional Path: Result not displayed
Preconditions • An error occurred Result is generated
Post conditions • Error message is displayed
Categorization • **Criticality:** High • **Probability of Defects:** Low • **Risk:** High

Table 3-8 Result Display Error

Functional requirements

Correct result of the video uploaded by the user, shall be displayed by the system.

3.5. Other Nonfunctional Requirements

3.5.1. Performance Requirements

- Use of preferred browser for the web application
- Fast and Secure internet connection
- No large file sizes.
- Response time to the view information shall take less than 5 seconds to appear on screen.

3.5.2. Safety Requirements

The use of the system should not cause any damage to the user.

3.5.3. Security Requirements

- Secure internet connection
- Secure Authorized access to web application via means of password.
- System will use secured database.

3.5.4. Software Quality Attributes

Availability

The availability of the system should be at all time except the maintenance time so that the user can access at any time required. In scenario of the hardware or any kind of failure to the system the page must be displaced in replacement of the original one to indicate the user of the failure.

Maintainability

The maintenance service provided should be easy and can be done at required moment of time. In case of any kind of failure, the re-initialization of the desired program should be done. In addition to the software the software design provided must be done with modularity keeping in mind the maintainability of the software can be done efficiently.

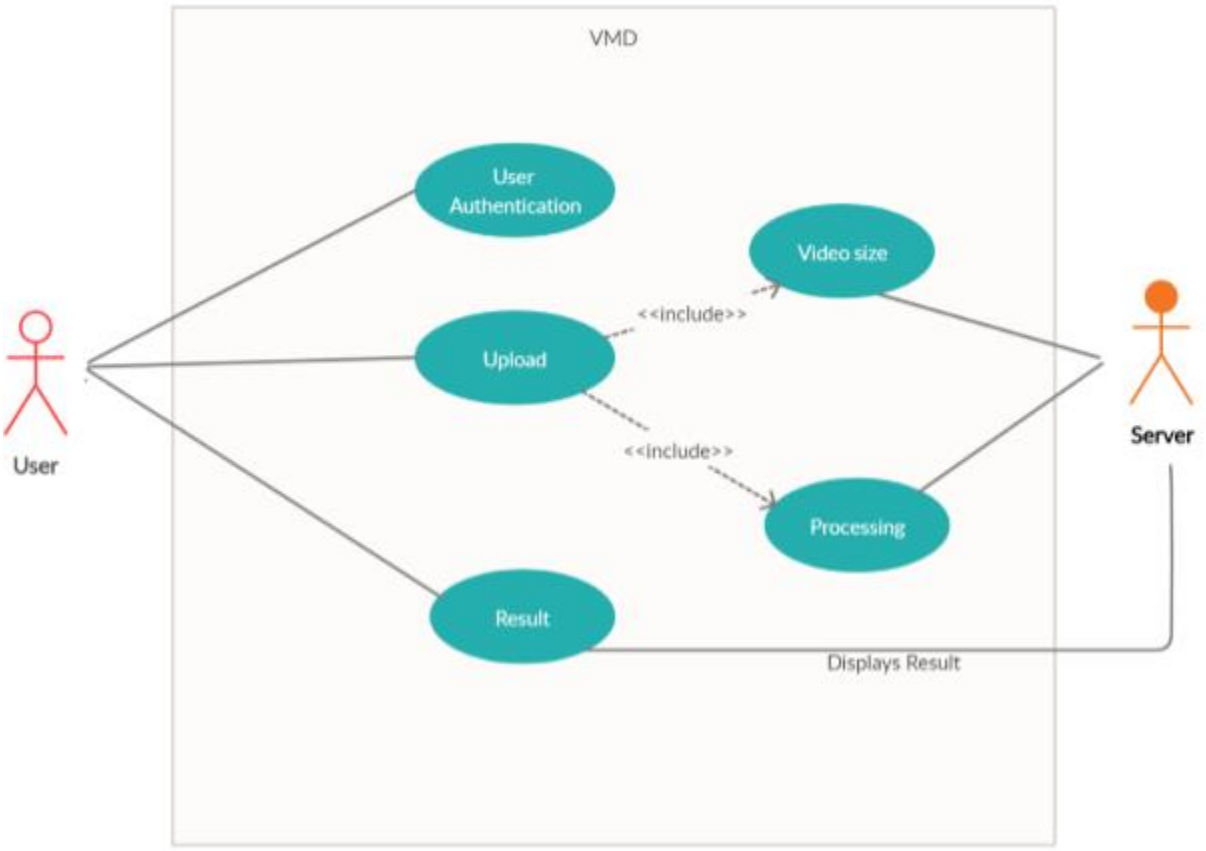
Portability

The web based application is based on HTML and scripting language. The solution provided to the end user is fully accessible through any location and any system using any web browser providing the portability for the end user. The user can access the system through any operating system.

Flexibility

The design of the application is made such that we can add further more videos to detect them simultaneously.

3.6. Other Use case Diagram



Chapter 4. Design and Development

4.1. Introduction

4.1.1. Purpose

The purpose of the document is to present system models & architecture in several different kinds of UML Diagrams, based on the refined software requirements for the project “Video Manipulation Detector”. This exhibition of the system architecture (modules, their interrelations, constraints, and working etc) will help in understanding the system working and to remove ambiguities. Moreover, it will facilitate the development team in recalling any characteristic of a module or the system itself in a speedy manner using abstract models, rather than finding it in plain text.

The main purpose of this descriptive software design document is to give you detailed overview of the architectural, user-interface, component and development level of the system. The provided design model shall be contained in software design document that will be latterly used as means for communicating information of software design, quality assessed and improved earlier code is generated. Several types of graphical representation such as class diagrams and verbal explanations to the system were added to this document to achieve the aim of developing the software of the system in the context of design model.

4.1.2. Project Overview

Main idea of the project is to develop a web application which will help a person who wants to detect the forged videos. It will also help crime investigation authorities detect that the video given by a person is forged or not. If a person wanted to check that the vide he has is manipulated or not then all he needs to do is to sign up and then upload the video. After that Server will process the video and apply model. After that, result will be displayed to the user after the complete processing of video at server side.

With the Help of this web application, a person will be able to detect the forged videos. It will help people as well as organizations (like police, FIA) to detect manipulated videos.

4.1.3. Definitions and Acronyms

The provided definition to the terms acronyms and abbreviations exist to properly interpret SDD. The definition provided are the items used in the SDD might not well known to the intended audience.

UML: Unified Modeling Language
App: Application
VMD: Video Manipulation Detector
Web: website

4.2. System Overview

4.2.1. Product perspective

The web app will be designed to help a person who will be able to detect the forged videos. It will also help crime investigation authorities to detect that the video given by a person is forged or not.

4.2.2. Product Functions

The main features of VMD are highlighted below:

1- Authentication of user
2- Video upload
3- Limitation of size and quality
4- Analysis
5- Detect Manipulation

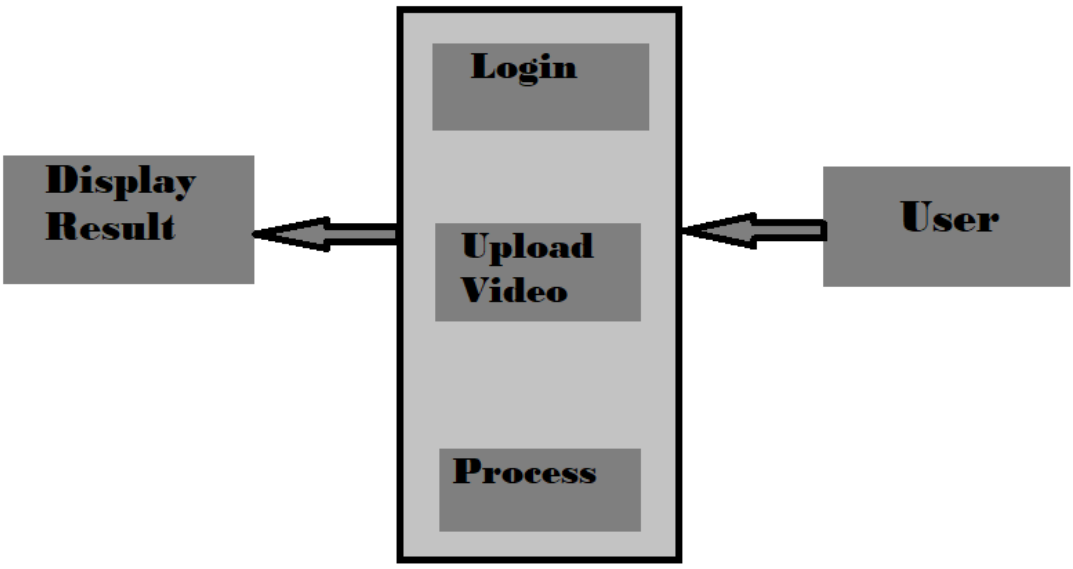
4.3 System Architecture

The overall working of the system is simple and basic. To establish a communication between User and Server, User have to login if he is registered otherwise, he must have to Sign up. Now after login user can upload a video for forgery detection. The Video uploaded by the user will be analyzed by the system and if uploaded video fulfills our criteria then model will be applied to the video. After the complete processing the result will be displayed on the dashboard of web application which is visible to user.

4.3.1 Architecture Design

The architecture design intended to visualize how a system is organized and displays the overall design structure of the provided system. It critically relates the design of the system and requirements engineering as it recognizes the structural components in the system and the links between them.

Following diagram shows how the components are interconnected:



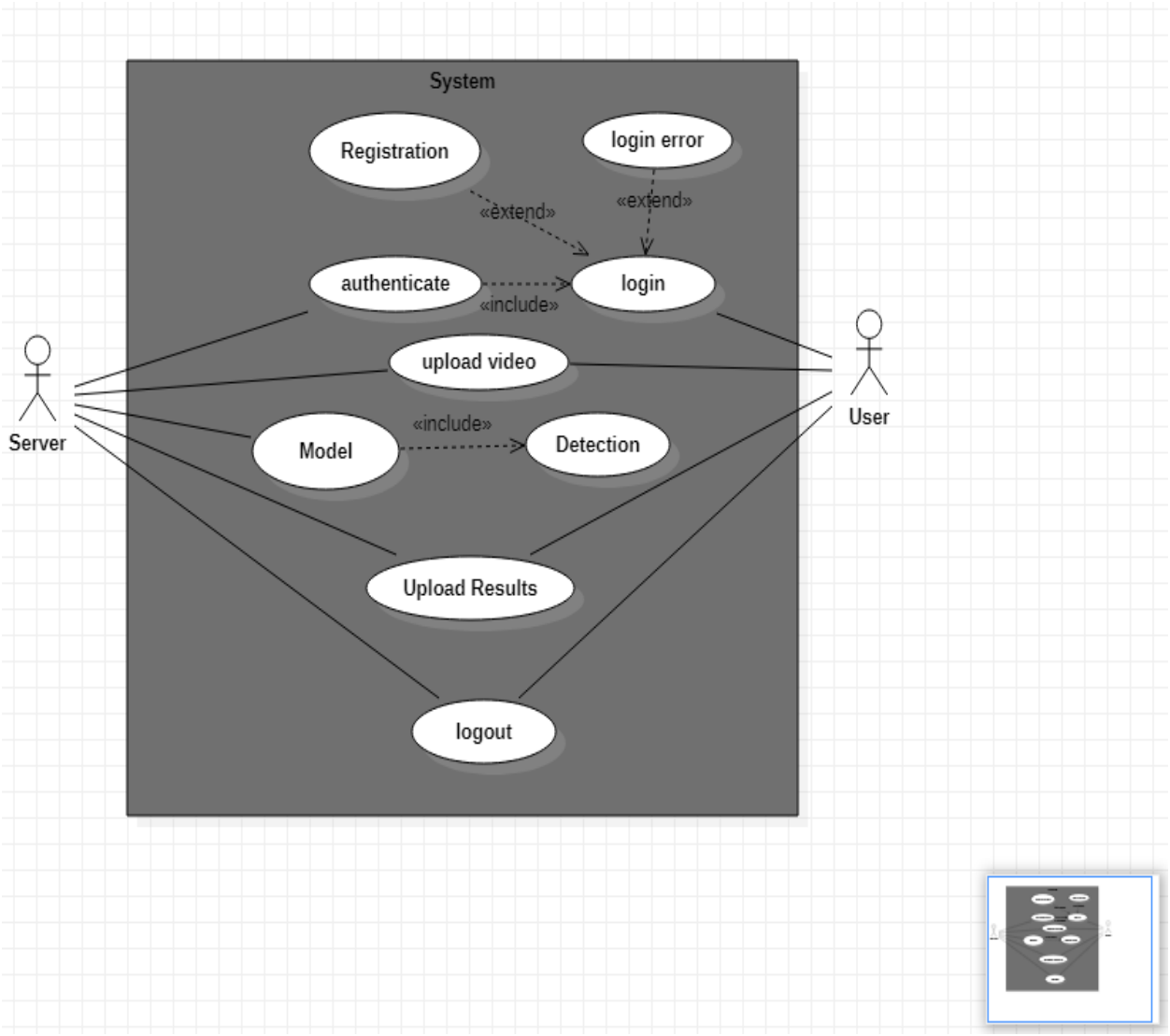
The web application will work as following:

- First, User of the application is required who want to use the application.
- Then User will go to the login form (if he is registered) to login otherwise a signup form will be displayed to the user where he/she gives related details in order to register with the system.
- When a user is logged in the he can upload video for forgery detection.
- The uploaded video will be processed by the system in order to check error. If there is no error, then model will be applied to the video in order to detect forgery.
- After the complete processing of video, the result will be displayed on the dashboard.

4.3.2. Decomposition Description

Now we will discuss the description of the decompositions of the system using different UML diagrams.

Use Case Diagram:



a) **Login:** The user enters the credentials to use the application.
b) **Login error:** The user enters invalid username or password, or it may be possible that user is not registered.
c) **Registration:** If user is not registered the he has to register himself/herself.
d) **Authenticate:** The database check for pre-registered user.
e) **Upload video:** The user can upload video and server will analyses the video (is there any error or not).
f) **Model:** Model Will check that the uploaded video is manipulated or not.
g) **Upload Result:** The server will display result on the dashboard.
h) **Logout:** The user can logout.

Login

Use case	Log in
Pre-condition	The User has to access the application and open the login section and User is previously added in the system.
Post-condition	The user is successfully logged into the system.
Basic Path	1. User must enter username and password. 2. The username and password given by the user are matched from the entries in the database. 3. If match found, user will see dashboard of application.
Alternative path	-
Exceptional path	User is not registered with the system or User entered wrong username or password.

Table 4-1 Use Case 1

Login Error

Use case	Login Error
Pre-condition	1. User is not Registered 2. User Entered wrong username or password.
Post-condition	User cannot access the dashboard of web application.
Basic Path	1.User access login form and entered his data.
Alternative path	-
Exceptional path	User is nor connected to internet.

Table 4-2 Use Case 2

Registration

Use case	**Registration**
Pre-condition	1. User is not Registered
Post-condition	User data is added to the system and now user can access the dashboard of web application.
Basic Path	1.User access Sign up form and entered his data.
Alternative path	-
Exceptional path	User entered something invalid.

Table 4-3 Use Case 3

Upload Video

Use case	**Upload video**
Pre-condition	1. User is registered. 2. User has video to upload.
Post-condition	Video is successively uploaded
Basic Path	1.User login and after that he/she uploaded video.
Alternative path	-
Exceptional path	Size of uploaded video is greater than specific limit.

Table 4-4 Use Case 4

Model

Use case	Model
Pre-condition	Video is successfully uploaded.
Post-condition	Model Will check either video is manipulated or not.
Basic Path	1.User login 2. User upload video 3 Apply model to check that video is manipulated or not.
Alternative path	-
Exceptional path	Server down during processing of video.

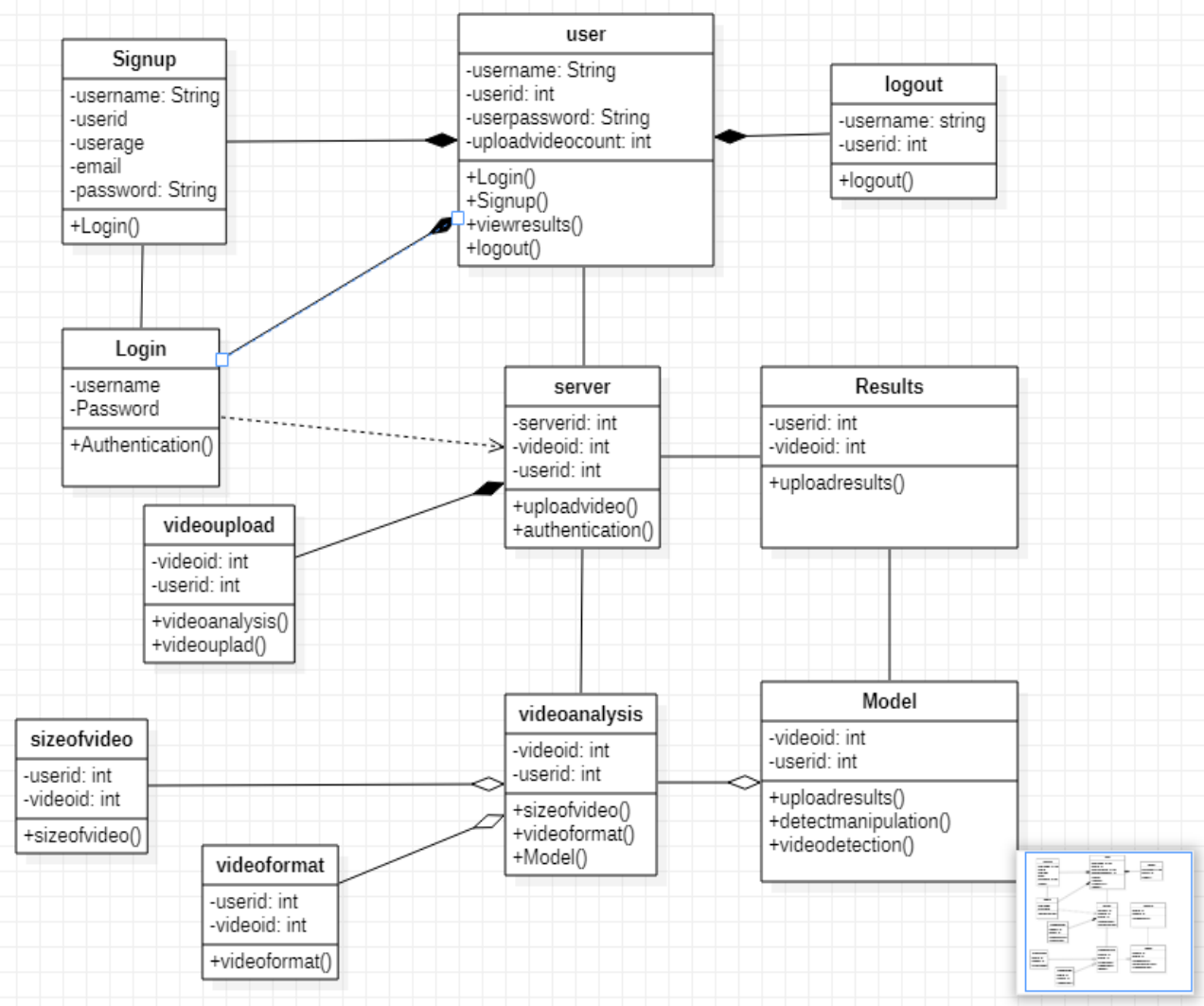
Table 4-5 Use Case 5

Upload Results

Use case	Upload result
Pre-condition	1.User is registered. 2. User has uploaded a video
Post-condition	Result is displayed on dashboard.
Basic Path	1.User login 2. User upload video 3 Apply model to check that video is manipulated or not.
Alternative path	-
Exceptional path	Server down or User lost internet connection.

Table 4-6 Use Case 6

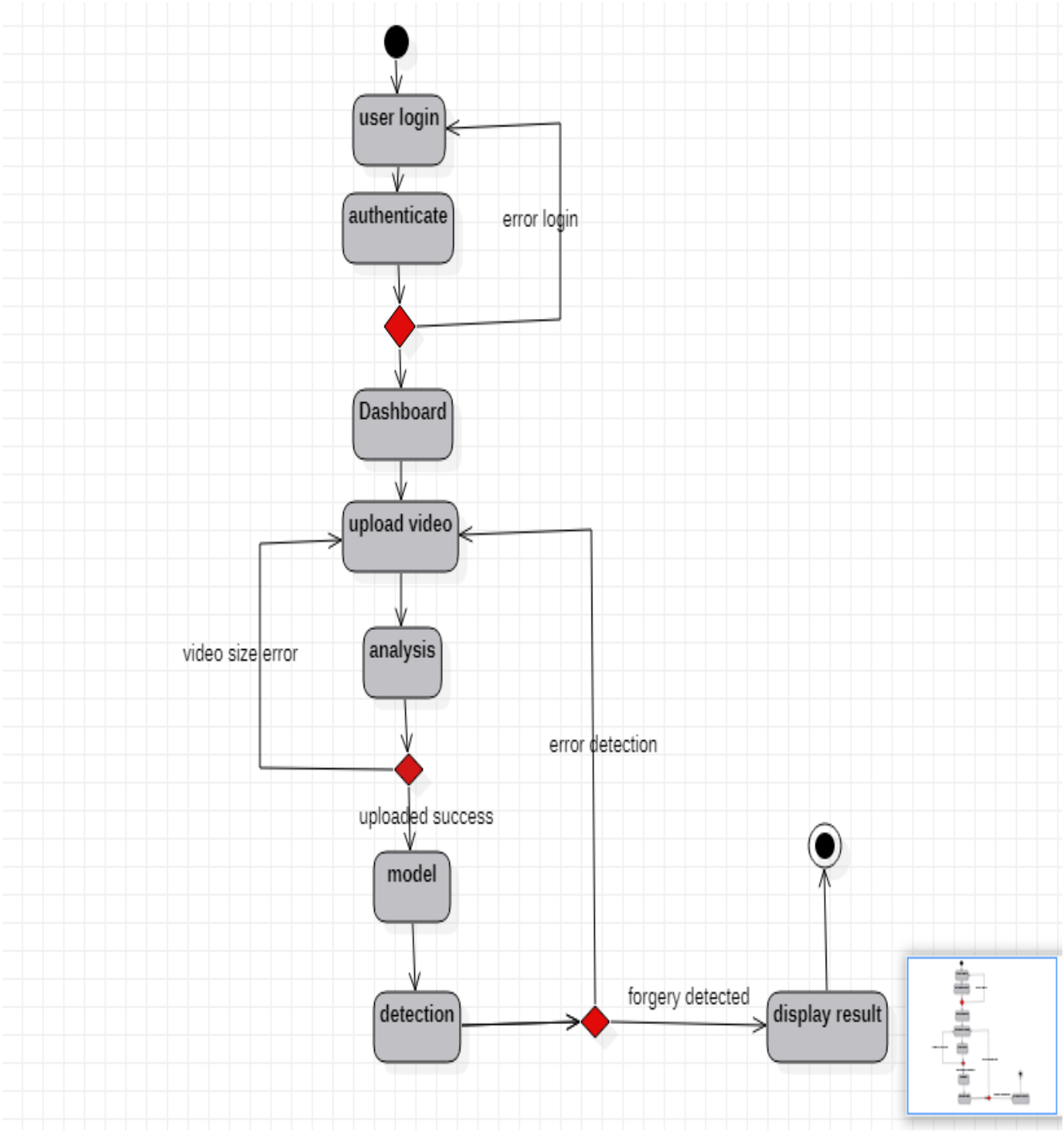
Class Diagram



The class diagrams shown have three sub systems to we generate classes to divide and simplify the execution.

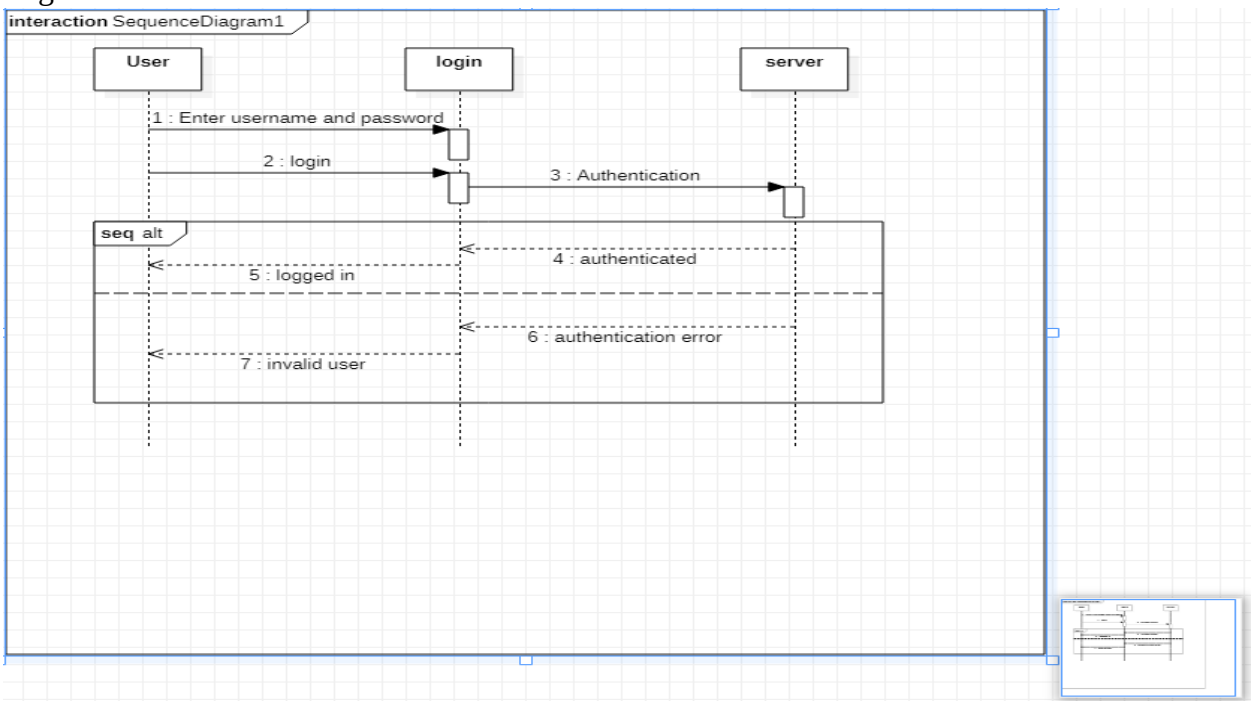
a) **Login:** Verify and grants access to a registered user**.**
b) **Sign up:** Register new user.
c) **Upload video:** Allow user to upload video.
d) **Size of video:** Checks the size of video uploaded by user.
e) **Video Format:** Checks the format of video.
f) **Video analysis:** Analyze the video uploaded by user.
g) **Model:** This will detect manipulation in the video.
h) **Result:** This will display result of video uploaded by user.
i) **Server:** This will authenticate user, analyze video and upload result.

Activity Diagram:

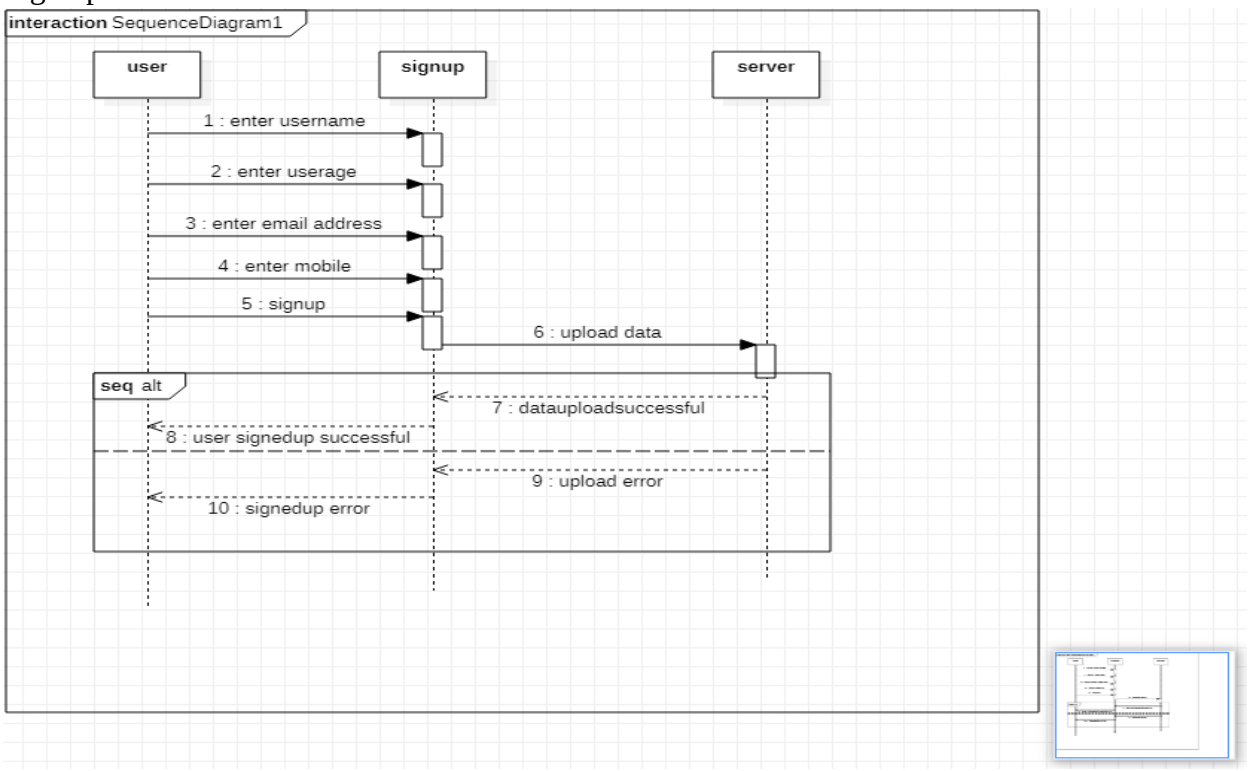


Sequence Diagram:

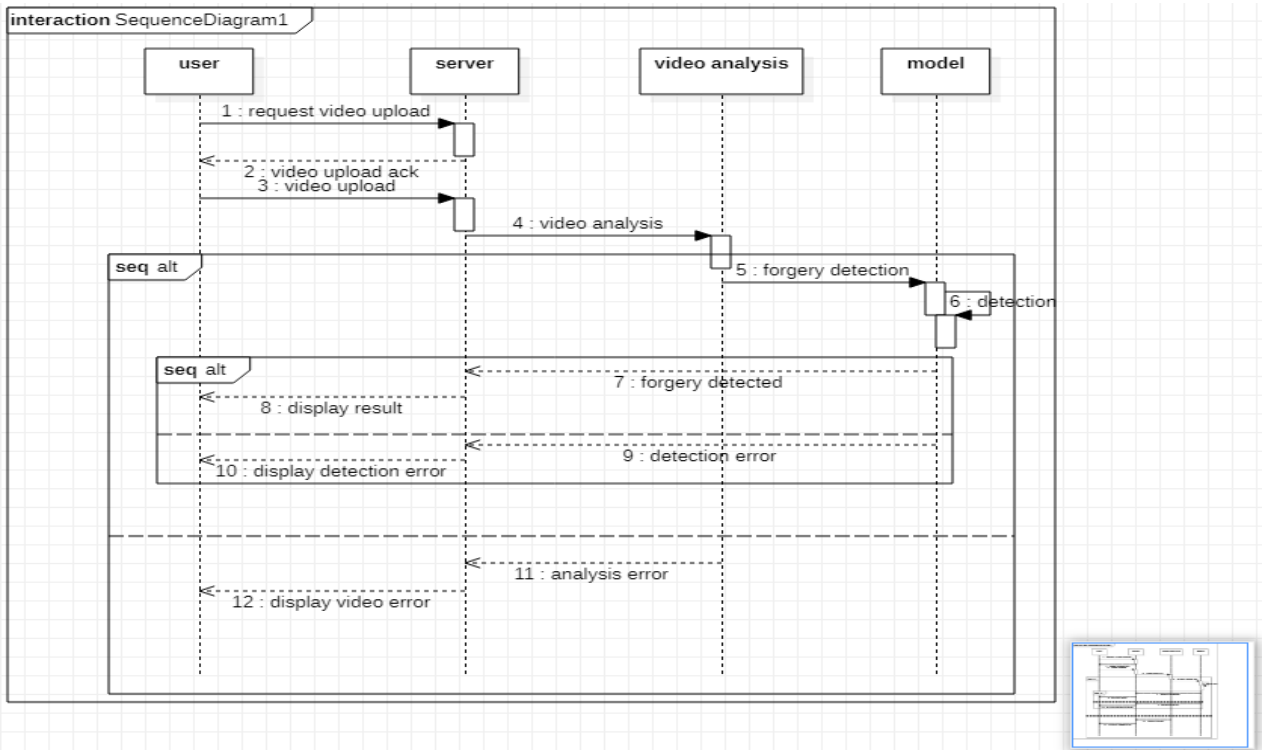
Login



Sign up



Upload Video



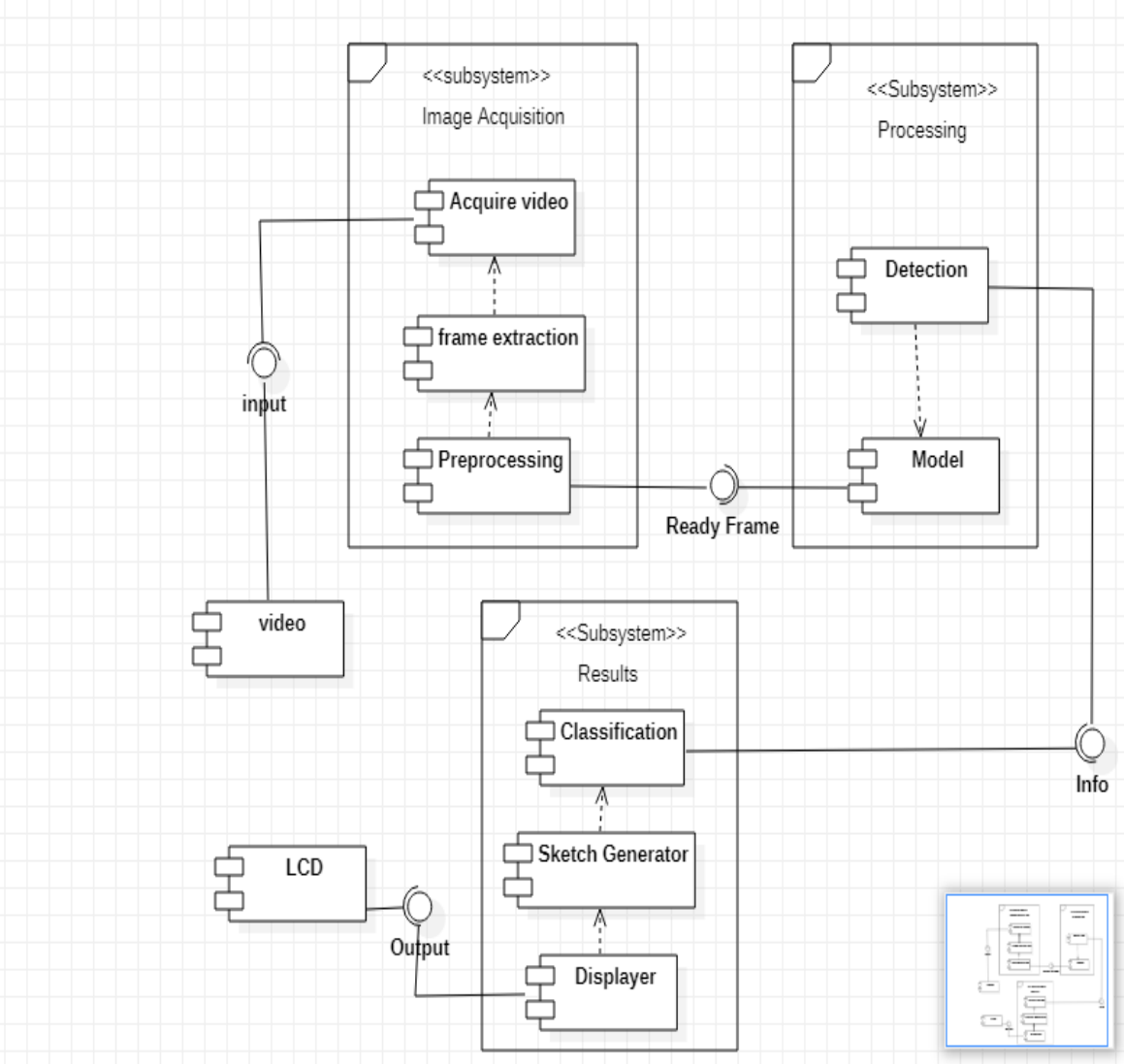
4.3.3. Design Rationale

We have used Client server model for our project. Client-server model partitions it into server and client both residing on the same system linked through a computer network on a separate hardware. A server running one or several process share its resources with clients. Following are the main benefits of the architecture client based results in resolving many different airing issues:

- The data is manipulated on the server side and is retained by usual business process for clients having access.
- The architecture is a distributed model portraying dispersed responsibilities between independent computers linked through a computer network. That is the reason it is easy to replace, repair, update or relocate server rather client remaining undisturbed.
- Servers having optimized control access and services to verify only authentic clients have access or manipulate data and also server updates are administered efficiently.
- Providing a framework for coding libraries where provided software components can easily be adjusted and optimized by the developer. This is quite useful for developing graphical user interfaces.

4.4 Component Design

This section describes all the modules of Video Manipulation Detector. These modules have been assigned responsibilities. Modules are further sub classified into components.



4.4.1. Image acquisition Module

This module performs all the pre-processing for VMD which includes video from user, frame extraction and frame enhancement technologies. This module provides the base for successful working of other modules.

4.4.2. Processing Module

This module performs all the processing related to the detection of forgery in the video. After the processing this module sends result info to the next module.

4.4.3. Results Module

This module will display the result info coming from the processing module. This module will display all the frames which are detected by the system during the processing of video.

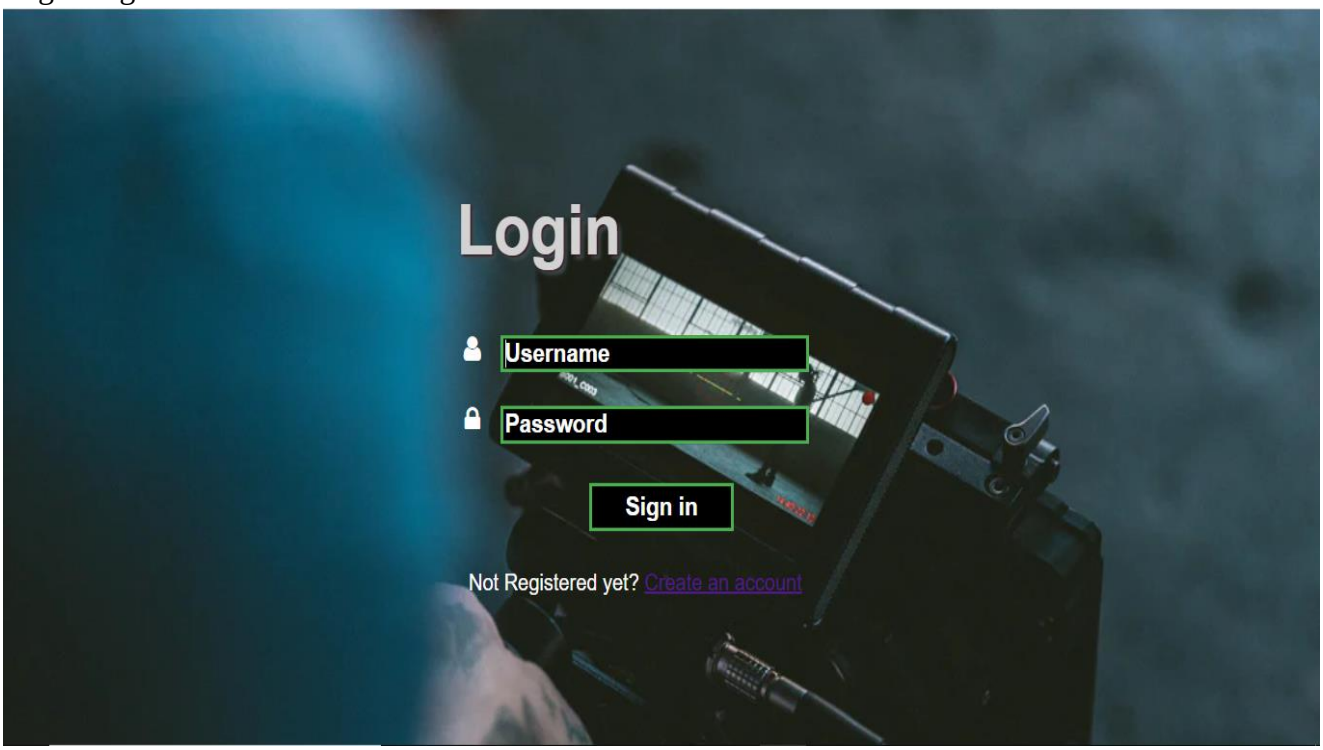
4.5. Human Interface Design

4.5.1. Overview of User Interface

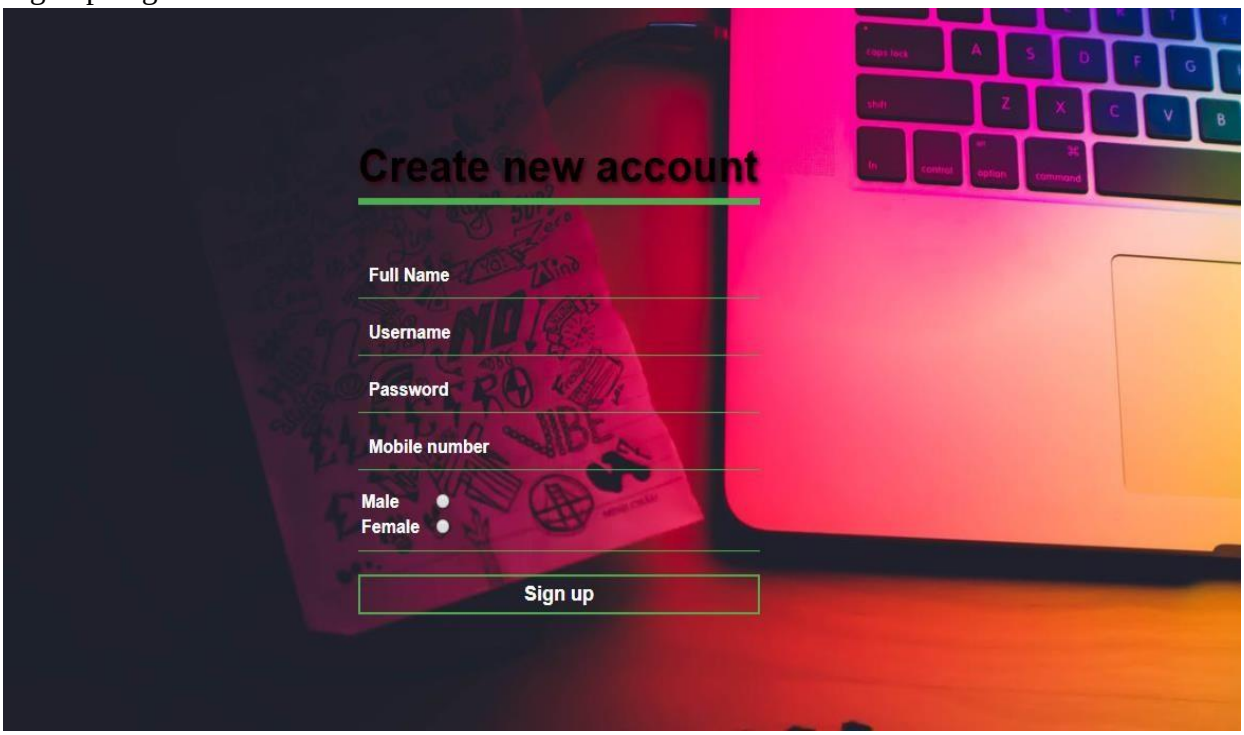
The system that is to be developed should support an intuitive and comfortable to use graphical user interface that is going to have a very ignorant learning curve evenly requiring minimum training to operate the system effectively. This is how, end user of the system won't feel reluctant to operate the system while the staff members and faculty won't feel reluctant to endorse the new provided system as a manner of academic analysis either.

4.5.2. Screen Images

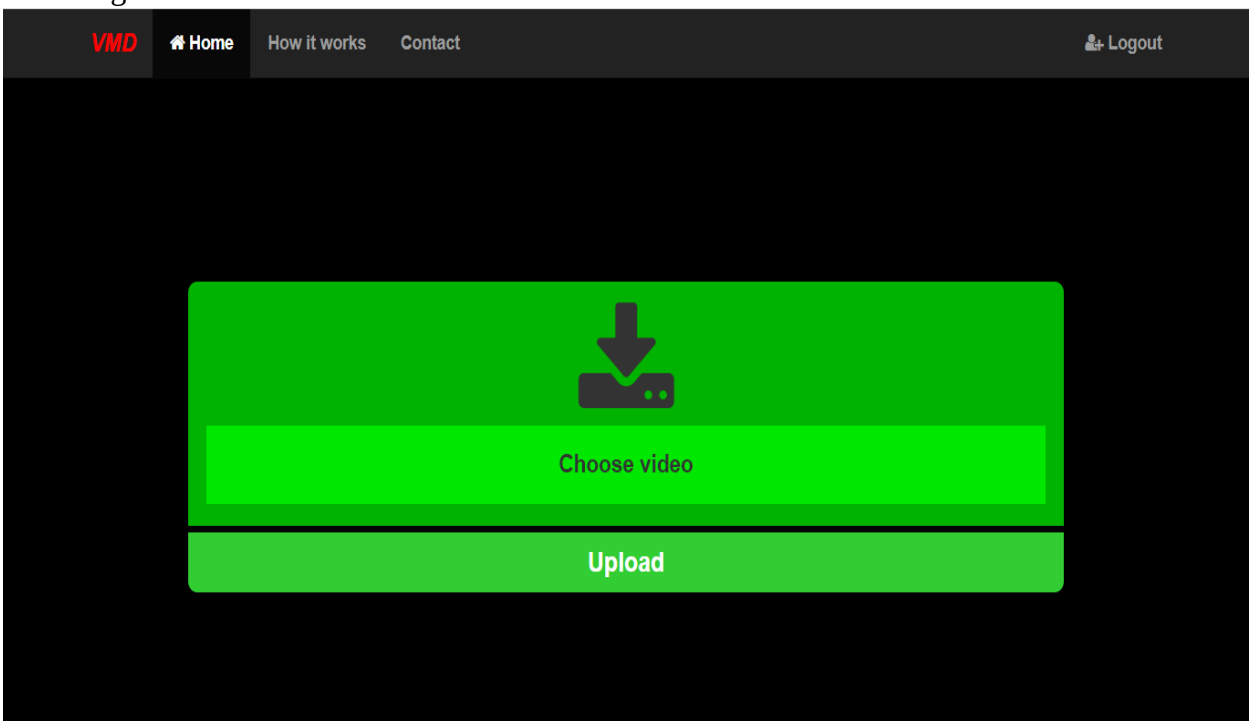
Login Page



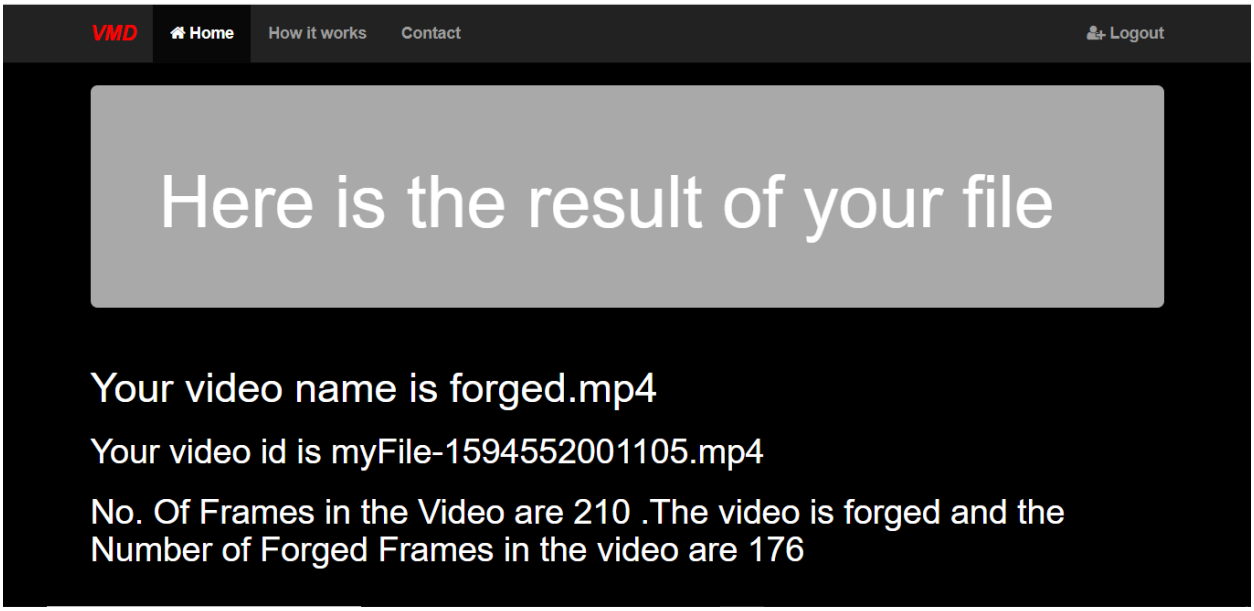
Sign up Page



Main Page



Result Page



4.5.3. Screen Objects and Actions

Login Page

Username

Username must be in 6 to 20 letters, in according to the industry standard. No use of special characters and space.

Password

Password must be in 6 to 20 letters can even include special characters.

Sign in

After pressing sign in button, the server authenticating the authenticity of the end user credentials the server takes them to the main page of the system.

4.5.4. Main Page

After successful login, user will be shown an interface with video component and buttons, the user will be able to:

- Click on choose video, in order to browse video.
- Click on upload button to upload video
- Click on “How it works” link in order to check how to upload a video.
- On logout button click, the user will logout from application.

4.5.5. Result Page

This page will show the number of frames which are forged and are detected by the system.

Chapter 5. Project Test and Evaluation

5.1. Introduction

This test plan section describes the methods and plans, processes and strategies employed to execute and manage the testing of the Application to detect forgery made in video. The test plan will ensure that all functionality is working in accordance with the defined requirements.

Manual Testing will be performed majorly which includes testing the software manually without the help of any automated testing tool/script by providing it inputs from the tester and see how the application behaves in different inputs. Whether it passes when it should and fail when it should and any unexpected action be noted and later on rectified.

A Unit test for every Unit of the project shall be performed independently and then will be joined together to test out the integrated behavior of all modules involved in the project, that us, integrated testing shall be done. Per unit we perform Black Box Testing. Per Integrated Model, we employ Acceptance Testing.

The Test scope includes all functional requirements and every critical use case discussed in the context of this project.

Any minor software test can be done during the development lifecycle albeit most of the testing will contextually make sense once the Unit has been fully constructed and integrated.

This document shows the plan, scope and procedure of testing the Project. This document shows what are the pass and fail criteria for the inputs to the modules.

5.2. Test Items

The test items selected for testing revolve around the following aspects.

1. Performance of VMD
2. Interface of the VMD
3. User control of the detection process via application

5.3. Features to Be Tested

In light of our project requirement, the design plan and every aspect of the project deemed important are to be tested. For maximum working efficiency the project requirements must be met and project working environment must be satisfied.

1. The VMD and its detection capabilities
2. The application
3. The Application over an internet.
4. The control experience of the application
5. Relevant feedbacks e.g. forged frames, number of forged frames.

5.4. Test Approach

Functional Testing will be done on every critical use case of the project where the behavior of different modules will be checked based on the inputs given and their actual vs expected outcomes to make a pass/fail criterion afterwards an integrated test will be performed which to ensure that all modules behave well together. Black box testing will be done on the modules.

5.5. Item Pass/Fail Criteria

Details of every test case have been explained in good detail in the Test Deliverable subsection. To make a pass/fail criterion for the Test cases, following approach is employed.

1. Preconditions are met .
2. The inputs are taken as stated.
3. The result works as what specified in output => Pass
4. The system doesn't work or not the same as output specification => Fail.

5.6. Suspension Criteria and Resumption Requirements

The test procedure will be immediately halted when a defect is found in a module. It shall be worked upon so as to ensure that maximum functionality is reached as per requirement.

Work has been done to make sure any probable failure does not occur but many unpredictable factors like network failure and hardware issues. Steps have been taken to make sure they remain minimized but they are not prone to unforeseen circumstances.

5.7. Test Deliverables

Testing tasks

1. Make the Test Cases.
2. Execute the test style on the test document.
3. Report the results of the test either as pass or fail.
4. Manage the changes made after testing.

Test cases

Following are the Test Cases:

i) **Login:** The user enters the credentials to use the application.
j) **Login error:** The user enters invalid username or password, or it may be possible that user is not registered.
k) **Registration:** If user is not registered the he has to register himself/herself.
l) **Authenticate:** The database check for pre-registered user.
m) **Upload video:** The user can upload video and server will analyses the video (is there any error or not).
n) **Model:** Model Will check that the uploaded video is manipulated or not.
o) **Upload Result:** The server will display result on the dashboard.
p) **Logout:** The user can logout.

Login

Test case ID	**TC 1**
Description	When given this command, user must be able to log in
Pre-condition	The User has to access the application and open the login section and User is previously added in the system.
Post-condition	The user is successfully logged into the system.
Basic Path	1. User must enter username and password. 2. The username and password given by the user are matched from the entries in the database. 3. If match found, user will see dashboard of application.
Actual Output	User logs in
Expected Output	User will log in
Status	User logs in successfully

Table 5-1 Test Case1

Registration

Test ID	**TC 2**
Description	When given this command, registration should be carried out
Pre-condition	1. User is not Registered
Post-condition	User data is added to the system and now user can access the dashboard of web application.
Basic Path	1.User access Sign up form and entered his data.
Actual Outcome	User registers himself
Expected Outcome	User will register himself
Status	User registers himself successfully

Table 5-2 Test Case2

Upload Video

Test ID	**TC 3**
Description	When command is given, video gets uploaded
Pre-condition	3. User is registered. 4. User has video to upload.
Post-condition	Video is successively uploaded
Basic Path	1.User login and after that he/she uploaded video.
Actual Outcome	Video gets uploaded
Expected Outcome	Video will be uploaded
Status	Video successfully got uploaded

Table 5-3 Test Case3

Model

TestCase ID	**TC 4**
Description	When command is given, model will detect manipulation
Pre-condition	Video is successfully uploaded.
Post-condition	Model Will check either video is manipulated or not.
Basic Path	1.User login 2. User upload video 3 Apply model to check that video is manipulated or not.
Actual Outcome	Model detects manipulation
Expected Outcome	Model will detect manipulation
Status	Model successfully detects if manipulation is made or not.

Table 5-4 Test Case4

Upload Result

TestCase ID	**TC 5**
Description	When command is given, results are uploaded
Pre-condition	1.User is registered. 2. User has uploaded a video
Post-condition	Result is displayed on dashboard.
Basic Path	1.User login 2. User upload video 3 Apply model to check that video is manipulated or not.
Actual Outcome	Results are uploaded
Expected Outcome	Results will be uploaded
Status	Results are successfully uploaded

Table 5-5 Test Case5

5.8. Responsibilities, Staffing and Training Needs

5.8.1. Responsibilities:

All developers involved in the team are responsible for testing out the project modules. Black box and Integration testing are to be done by them. For good measure the developer not involved in the module should test it to remove bias. Any defect found should be noted, reported and then rectified

5.8.2. Staffing and Training Needs:

Elementary knowledge of testing techniques should be known to every developer. Along with a good sense of reasoning and ability to think out of the box to make cases where application might suffer from defect is good so as to cover any and every sort of problem possible to the maximum.

As mentioned earlier, all developers shall be testing each other's work that they were not directly involved with to remove the developers bias and effectively improve the project.

5.9. Risk and Contingencies

Extensive work has been done and proper research and planning was done since the inception of the project, to every change involved till finality. However as emphasized earlier, the project is not immune to unpredictability and as security was not a huge part of the project lifecycle, the unpredictability factor exists manifold. However, problem rectification per project requirement has been performed and any problem was solved using following management strategies.

5.9.1. Schedule Risk:

The project requirement was changed and unexpected and unwanted results were brought up. Coupled with the time it took to make wanted output and create one unit ready to be tested by the application put serious dents on our schedule. However, with suitable "Just in Case" plans and "Plan B's" such was taken care of in what may not be described as the best way but a way that worked towards a solution not add on a problem.

5.9.2. Operational Risks:

The operational risk should be minimized by scheduling daily meeting and regular deadlines to achieve the aims of the solution of the project even providing better communication in between the group members.

5.9.3. Technical risks:

To minimize technical risk, any predefined requirement shall be kept constant and any refined one shall be thoroughly studied and consulted with by the supervisor.

5.9.4 Programmatic Risks:

To minimize programmatic risks, the requirements and project functionalities shall be kept under the functional requirement constraints and non-functional requirements shall be kept on lower priority.

Chapter 6. Future Recommendations

In recent years, VMDs have seen an immense increase in utility going across quite a number of fields like Military, Investigating institutions and courts. In the same way much work has been done to enhance the current ability of these VMDs.

VMDs are state of the art, robust, scalable. However, using our concept, they are quite few in the market and expensive to boot not to mention a security threat.

This project has a greater potential than what is currently perceivable. With the help of the Open Source APIs, great technical knowledge and a drive for improvement, this project can reach even greater heights. Currently, in context to the use idea involving limited sized and quality detectors for forgery detection purposes, following additions can be made to make it suitable or better for work

1) It can be upgraded to detect videos with .AVI and 3gp extensions.
2) Currently it has been trained for a limited sized videos. However, in future, it can be made efficient for larger sized videos as well.
3) For processing speed, it can be increased and forged frames can be detected within less time as compared to its current time. Latency can be minimized.
4) Basically, it detects inter-frames manipulation, but in future it can be improved in a way to detect intra-frames manipulation too.
5) For now, it detects manipulations in videos. In future, it can be upgraded to detect manipulation made in audio as well.
6) Also it can be able to find objects addition or removal from videos.
7) It can be shifted to android and mac platform as well.
8) For now, it takes one video at a time for detection. In future, can be upgrade to process multiple videos simultaneously.

Chapter 7. Conclusion

With the availability and use of modern technology, open source software's, the team's tireless efforts and our Supervisors support in us, the project has successfully fulfilled its idea. Although detection effort had to be done with the suitable hardware as well, the major idea of the project was successfully reached.

This project as said in the Future Work section has a long way to go with many improvements needed. Our work is not even the tip of the ice berg for what this project can be. Hopefully someone in the future will take this work up and improve upon it but as the team that worked hard on this project, we aim to study and improve upon this project greater and greater to make sure it reaches the maximum height possible while providing the utility aimed at our specific target.

Not that VMD technology is a new thing, many multinational companies have produced them for commercial purposes but are proprietary. This project aims to tell people to work on their own customized and specific VMD behavior to learn more in depth rather be fascinated by commercial detectors no matter how pretty they are.

In conclusion we would sign off by saying that the project is not difficult. The technology used is not difficult and any person with a technical background can work on this easily. That is what we seek. We are hopeful some ambitious group will take on this project and improve it.

Great Job to the Team

Thank you to the Supervisor

Best wishes for our friends and family.

Chapter 8. Appendices: Glossary, Abbreviations

VMD: Video manipulation detector
Web: Website
App: Application
SDK: Software development kit
UML: Unified Modeling Language

Activity diagrams - are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows).

Class diagram - In the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

SDS – In the context of software, Design Specification is usually a design document that describes all data, architectural, interface and component-level design for the software. A design specification provides explicit information about the requirements for a product and how the product is to be put together.

Sequence diagram - is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

SRS–A software requirements specification is a description of the software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide.

Use case diagram - At its simplest is a representation of a user's interaction with the system and depicts the specifications of a use case. A use case diagram can portray the different types of users of a system and the case and will often be accompanied by other types of diagrams as well.

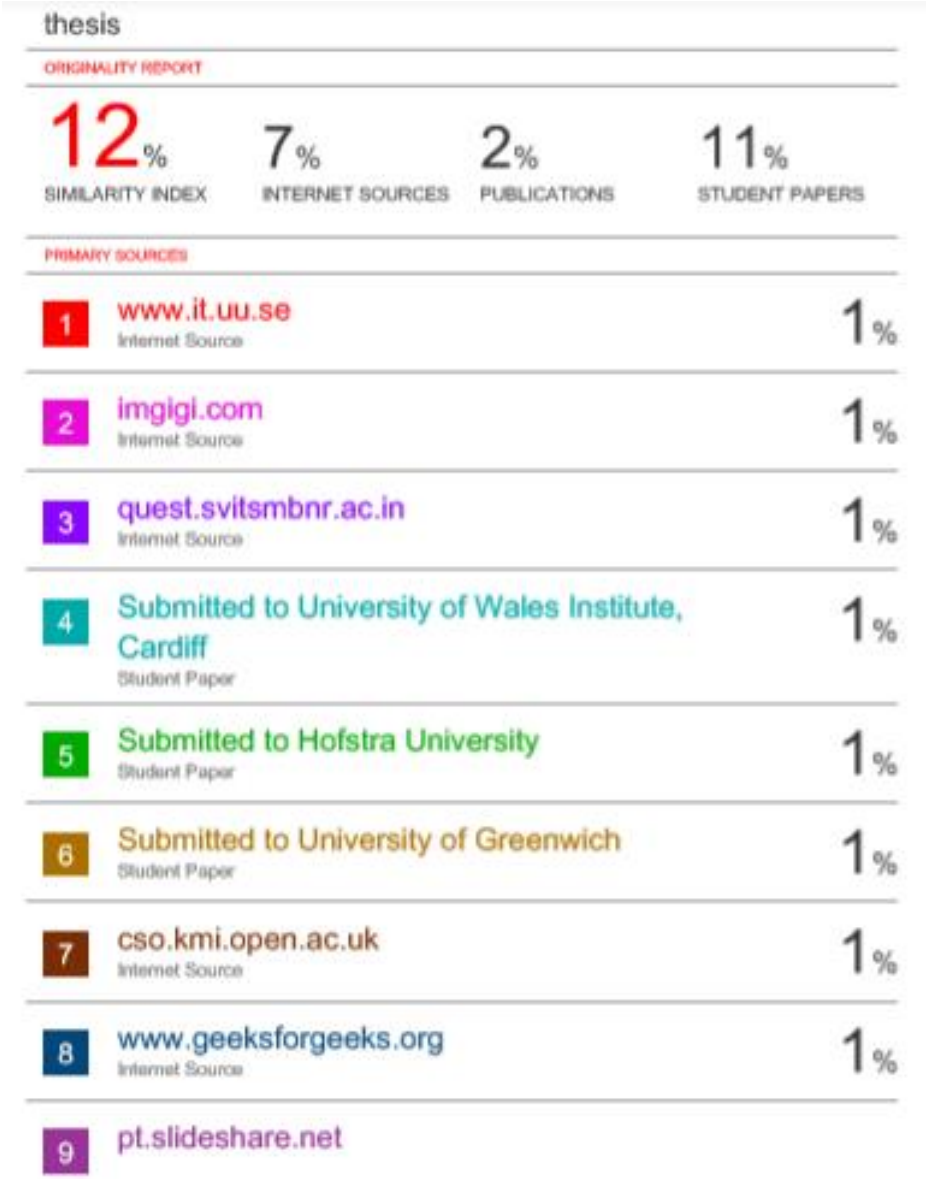
Chapter 9. Reference Materials

a) Use Case Modeling Guidelines, which documents the guidelines used to develop the use case model specifying the functional requirements in this specification. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=787548

b) EXPOSING VIDEO INTER-FRAME FORGERY BASED ON VELOCITY FIELD CONSISTENCY by Yuxing Wu, Xinghao Jiang*, Tanfeng Sun and Wan Wang School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. https://www.researchgate.net/publication/271425160_Exposing_video_interframe_forgery_based_on_velocity_field_consistency..

c) Video Inter-Frame Forgery Identification Based on Consistency of Correlation Coefficients of Gray Values by Qi Wang, Zhaohong Li, Zhenzhen Zhang, Qinglong Ma School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China. https://www.researchgate.net/publication/274109330_Video_InterFrame_Forgery_Identification_Based_on_Consistency_of_Correlation_Coefficients_of_Gray _Values

Plagiarism Report



thesis

ORIGINALITY REPORT

12%	7%	2%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.it.uu.se Internet Source	1%
2	imgigi.com Internet Source	1%
3	quest.svitsmbnr.ac.in Internet Source	1%
4	Submitted to University of Wales Institute, Cardiff Student Paper	1%
5	Submitted to Hofstra University Student Paper	1%
6	Submitted to University of Greenwich Student Paper	1%
7	cso.kmi.open.ac.uk Internet Source	1%
8	www.geeksforgeeks.org Internet Source	1%
9	pt.slideshare.net	

	Internet Source	<1%
10	Amal El Kaid, Karim Baïna, Jamal Baïna. "Reduce False Positive Alerts for Elderly Person Fall Video-Detection Algorithm by convolutional neural network model", Procedia Computer Science, 2019 Publication	<1%
11	Submitted to 8971 Student Paper	<1%
12	www.slideshare.net Internet Source	<1%
13	Submitted to University of Hong Kong Student Paper	<1%
14	Submitted to Queen Mary and Westfield College Student Paper	<1%
15	Submitted to Higher Education Commission Pakistan Student Paper	<1%
16	Submitted to University of Nottingham Student Paper	<1%
17	Submitted to Victoria University Student Paper	<1%
18	Submitted to Hong Kong Baptist University Student Paper	<1%

19	Submitted to Universiti Tenaga Nasional Student Paper	<1%
20	id.scribd.com Internet Source	<1%
21	Submitted to Misr International University Student Paper	<1%
22	dspace.up.edu.ps Internet Source	<1%
23	Submitted to University of Maryland, University College Student Paper	<1%
24	Submitted to University of the West Indies Student Paper	<1%
25	Submitted to Stuyvesant High School Student Paper	<1%
26	Submitted to Intercom Programming & Manufacturing Company Limited (IPMC) Student Paper	<1%
27	Submitted to Manipal University Student Paper	<1%
28	profile.typepad.com Internet Source	<1%
29	Submitted to KDU College Sdn Bhd Student Paper	<1%

30	Submitted to Universiti Teknologi MARA Student Paper	<1%
31	Andrew Crabtree, Mark Rouncefield, Peter Tolmie. "Doing Design Ethnography", Springer Science and Business Media LLC, 2012 Publication	<1%
32	Submitted to University of Dammam Student Paper	<1%
33	Submitted to Majan College Student Paper	<1%
34	Submitted to University of Technology, Sydney Student Paper	<1%
35	Submitted to Texas State University- San Marcos Student Paper	<1%
36	Submitted to Informatics Education Limited Student Paper	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		