

A MINI PROJECT REPORT
ON
WEB-BASED COLLABORATIVE BIG DATA ANALYTICS ON BIG DATA

Submitted to Sri Indu College of Engineering & Technology, Hyd
In partial fulfilment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

K.JONATHAN RAJ	(18D41A0595)
K.JAGADISH	(18D41A05A0)
K.SAI RAM	(18D41A0592)
K.RESHMA	(18D41A05A4)

Under the esteemed guidance of
Dr.T.KUMARESAN



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution under UGC, Accredited by NBA&NAAC, Affiliated to JNTUH)

Sheriguda, Ibrahimpatnam (2020-2021)

SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution under UGC, Accredited by NBA&NAAC, Affiliated to JNTUH)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that the mini project work entitled

“WEB-BASED COLLABORATIVE BIG DATA ANALYTICS ON BIG DATA”

Submitted by

K.JONATHAN RAJ

(18D41A0595)

K.JAGADISH

(18D41A05A0)

K.SAI RAM

(18D41A0592)

K.RESHMA

(18D41A05A4)

In partial fulfilment for the award of Bachelor of Technology in Computer Science and Engineering to the SICET, is a record of Bonafide work carried out by them under my guidance and supervision during academic year **2020-2021**.

Internal Guide

HOD

(Dept of CSE)

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We are thankful to principal **Dr. G. SURESH**, for giving us the permission to carry out this project and for providing necessary infrastructure and labs. We are highly indebted to **Dr. T. CHARAN SINGH**, Head of the Department of Computer Science Engineering, for providing valuable guidance at every stage of this project.

We are grateful to our internal project guide **Dr. T. KUMARESAN** for his constant motivation and guidance given by him during the execution of this project work.

We would like to thank the Teaching & Non-Teaching staff of Department of Computer Science and engineering for sharing their knowledge with us, last but not least we express our sincere thanks to everyone who helped directly or indirectly for the completion of this project.

K.JONATHAN RAJ

K.JAGADISH

K.SAI RAM

K.RESHMA

ABSTRACT

Big data describes the large volume of data both structured and unstructured data that helps a business on a day to day basis. Big data basically consists of capturing data, searching, sharing, and storing the data. Big data analytics refers to the strategy of analysing large volume of data or big data. This big data is gathered from a wide variety of sources, including social networks, videos, digital images. A service is a discrete unit of functionality that can be accessed remotely and acted upon and updated independently. Amazon and Flipkart are the examples of services.

Now a days, the data has rapidly increased with cloud computing and social websites. So it is difficult to manage and process the data in traditional technologies such as relational data base and scale up infrastructure. Relational data base is designed to run on single server can not handle huge amount of data. So new collaborative big data analytics platform has introduced for more efficient services, with collaborating data owners, data scientists and service developers.

A collaborative analytics platform is designed which focus on big data analytics software as a service. It provides all the three commonly required services, big data infrastructure as a service, big data platform as a service, big data software as a service; in a single encapsulated service named big data as a service. This project works on implementing the collaborative big data analytics on big data as a service platform.

CONTENTS

S.NO.	CHAPTERS	pg.no
	ABSTRACT	4
	LIST OF FIGURES	6
	LIST OF SCREENSHOTS	7
1. INTRODUCTION		8 - 9
2. SYSTEM ANALYSIS		10
	2.1 EXISTING SYSTEM	
	2.2 PROPOSED SYSTEM	
	2.3 SYSTEM SPECIFICATION	
3. SYSTEM STUDY		12
	3.1 FEASIBILITY STUDY	
	3.2 TECHNICAL FEASIBILITY	
	3.3 SOCIAL FEASIBILITY	
	3.4 ECONOMICAL FEASIBILITY	
4. SYSTEM DESIGN		14
	4.1 ARCHITECTURE	
	4.2 UML DIAGRAMS	
	4.2.1 USECASE DIAGRAM	
	4.2.2 CLASS DIAGRAM	
	4.2.3 SEQUENCE DIAGRAM	
	4.2.4 COLLABORATIVE	
	4.3 INPUT AND OUTPUT DESIGN	
5. IMPLEMENTATION		72
	5.1. SOFTWARE ENVIRONMENT	
	5.1. NETBEANS	
6. SCREENS		94
7. TESTING		99
8. CONCLUSION		103
	REFERENCES	104

LIST OF FIGURES

Fig No	Name	Pg. No.
Fig. 1	Architecture diagram	12
Fig. 2	Use case diagram	14
Fig. 3	Class diagram	15
Fig. 4	Activity diagram	17
Fig. 5	Sequence diagram	18
Fig. 6	Dataflow diagram	20
Fig. 7	Java Programming Language	24
Fig. 8	Java Platform	25
Fig. 9	Java 2SDK	26
Fig.10	TCP/IP Stack	32

1 : INTRODUCTION

Social media is drastically advancing their feature day by day while making themselves from social networks to Geosocial Networks. It empowers people to make their content public along with their geographical information. This has resulted in an increase in the use of Geosocial Networks by providing users with the ability to voice opinions, report events, and share views, anger, or love while connecting with others, which was unthinkable in the pre-Internet age. The information shared in any media is geosocial because: 1) the posts have extensive content that represents geographical information with specific locations that are either entered explicitly (by check-in) or added implicitly (by Earth coordinates, such as latitude or altitude), and 2) the views shared on social media reveal social knowledge and strengthen relationship and communication. Advances in technology have allowed the use of GPS systems in smartphones, which made location data more dominant. The location of people posting, commenting, or uploading pictures on social media is recorded. Thus, by aggregating such type of location data from all network users, social networks produce warehouses of geosocial data. Another method for generating geosocial data is by crowdsourcing while providing self-developed applications intended for various purposes or causes. This takes geosocial data from volunteers or paid users who provide data or information for that cause. For example, during the Haiyan typhoon in the Philippines in 2013, a virtual community composed of a large number of supporters, volunteers, and IT professionals created online street maps for emergency relief [1]. Such type of online information gathered by crowdsourcing was given the name “volunteer geographic information” (VGI) [2]. Today, many platforms and software have been developed to use crowdsourcing for geosocial data harvesting in order to maximize business, promote causes, or for other commercial purposes. An example of one such software platform is Ushahidi, which allows report creation by matching a specific keyword on a Geosocial Network that corresponds to various specific locations. The report can then be used for social awareness and help in the case of emergencies or disasters [3]. There are also some other applications with limited functionalities, such as Hootsuite [4] and 140kit [5]. Currently, researchers are more interested in Geosocial Networks because they consider them new data assets [6, 7]. Crooks et al. [8] used Twitter data (geolocated tweets) in their work to map earthquakes in the United States. Among 21,362 geolocated tweets, the first tweet that announced an earthquake appeared 1 minute after the earthquake [8]. Similarly, Chow [9] and Papadimitriou [10] proposed architectures using geotagged recommendations while creating a new social network. GeoLife 2.0 [11] performed similarity detection, trajectory analysis, and recommendation selection. There are few other systems that perform analysis on the content of tweets, e.g., [12]—some use locations. Some research has also been performed to detect hotspots and hyper-local events in a city using tweeter data [13, 14]. Social network data could be beneficial for many fields if well analyzed. By analyzing the social behavior of a community [15] in a particular area by filtering and profile matching, one can recommend people a shop, hotels, cheap markets, banking systems, advertisements, etc., based on their likes and constraints [16]. Similarly, based on people and vehicles movement, the authorities can perform better city plans [17-19] and recommend suitable traffic routes to people based on current circumstances [20, 21]. In addition, Social network data analysis, such as Twitter is used in many healthcare applications [22-24], to monitor and control fatal diseases and infections. There are several other works that have been performed using

Geosocial Networks that employ limited geolocated information. However, all of the existing systems do not consider the processing aspects of real-time, high-speed geosocial traffic. There are few other high-speed big data processing system aimed at establishing healthcare system [25], processing M2M data using data fusion model [26], establishing smart city [27], etc. On the other hand, These system are not suitable to process social network data because of its geosocial nature. Using geosocial network data is not only beneficial to governments, but it can also have a major impact on human life. Geosocial Network data can provide benefits to normal citizens and business people. However, when harvesting geosocial data from networks such as Twitter or Facebook, it should be noted that these networks have millions of users who post thousands of tweets and statuses with an hour. Therefore, it can be easily reasoned that all the users of various social networks generate a significant amount of data: such data might range in the terabytes within minutes. Consequently, harvesting such realtime geo-social data is a very challenging task. We need a special computational environment and advanced computing techniques with intelligent management in order to provide intime/real-time analysis. All the aforementioned techniques do not consider more than one social network at a time, and their analyses are scalable in terms of data size. Therefore, in order to meet these computational challenges, in this paper, we propose an advanced geosocial data analytical system that not only processes offline data efficiently within a time limit but also provides real-time data analysis for various social networks, including Twitter, Flickr, Facebook, YouTube, etc. The system deploys a Hadoop ecosystem for data processing and analysis with Spark at the top it as a third-party tool for real-time analysis. We tested the system by taking two social networks, Twitter and Flickr. The rest of the document describes the proposed system, including data analysis, implementation, and evaluation.

2 : SYSTEM ANALYSIS

Introduction to Modules:

User

Double click on run.bat file to get below screen

Click on 'Upload Video File' button to upload video

Now click on 'Start CCTV Object Detection From Frames' button to start detecting objects from video

In above screen we can see while video playing application will start detecting objects and all those objects details we can see in black console also

Now click on 'Upload CCTV Metadata to Hadoop' button to save metadata of video to be saved on hadoop

Now we can see this data in Hadoop with file name as date and time and format of file is in xml

From Hadoop we can download and see that file data

In above screen click on download link to download data

In browser status bar we can see data is downloaded, now we can open and see that video data

In above screen we can see xml data displaying video name, points of frame, frame id and detected objects.

2.1 EXISITNG SYSTEM:

Literally, big data is a large volume of data so that it is difficult to collect, store, and analyze. However, big data is not just about a large volume of data. It is a concept that provides an opportunity to find useful knowledge into existing data.

2.2 PROPOSED SYSTEM:

Big data is characterized by 3Vs: Volume, Velocity, and Variety[4]. First of all, there has been the exponential growth in the data storage as data format is diversifying. We can find various data formats such as video, music, and large image files on the social network service channels. It is common to have the storage system of which size is Terabytes or Peta bytes. Secondly, velocity is often identified with real-time analytics. However, velocity is also about the rate of changes. There was a time when data had been hardly changed or the rate of changes is very low. Yet, today people reply on SNS (Social Network Service) to update their messages. Sometimes, a few seconds old messages is discarded or updated because users are not interested in any more. They often remove

old messages and pay attention to recent updates. This high velocity data illustrates big data. Lastly, data can be stored in multiple formats. For example, the data would be stored in a simple text file or stored in the form of video, SMS, pdf or user defined format. In this case, we should consider how to process a variety formats of data. This variety of data describes big data.

2.3 SYSTEM SPECIFICATION:

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

- Processor - Pentium –III
- RAM - 256 MB (min)
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

SOFTWARE REQUIREMENTS:

- Operating System : WindowsXP
- Front End : HTML, Jsp
- Scripts : JavaScript.
- Server side Script : awt, swings.
- Database : MySQL 5.0
- DatabaseConnectivity : JDBC

3 : SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are :

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4 : SYSTEM DESIGN

UML DIAGRAMS :

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

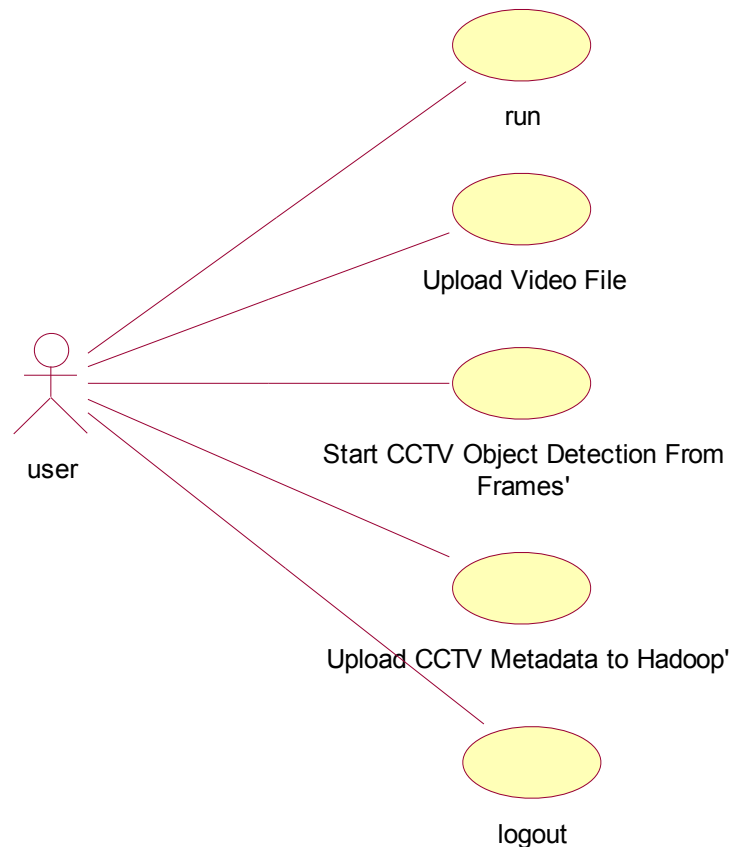
GOALS :

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM :

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



USECASE:

```
(object Petal
  version      42
  _written     "Rose 4.5.8054a"
  charSet      0)
(object Design "Logical View"
  is_unit      TRUE
  is_loaded    TRUE
  quid         "5CFEDEB10313"
  defaults     (object defaults
    rightMargin 0.250000
```

```

leftMargin    0.250000
topMargin     0.250000
bottomMargin  0.500000
pageOverlap   0.250000
clipIconLabels TRUE
autoResize    TRUE
snapToGrid    TRUE
gridX         16
gridY         16
defaultFont   (object Font
  size        10
  face        "Arial"
  bold        FALSE
  italics     FALSE
  underline   FALSE
  strike      FALSE
  color       0
  default_color TRUE)
showMessageNum 1
showClassOfObject TRUE
notation       "Unified")
root_usecase_package (object Class_Category "Use Case View"
  quid            "5CFEDEB10315"
  exportControl   "Public"
  global          TRUE
  logical_models  (list unit_reference_list)
  logical_presentations (list unit_reference_list
    (object UseCaseDiagram "Main"
      quid            "5CFEDEB2001E"
      title           "Main"
      zoom            100
      max_height      28350
      max_width       21600
      origin_x        0
      origin_y        0
      items            (list diagram_item_list))))
root_category   (object Class_Category "Logical View"
  quid          "5CFEDEB10314"
  exportControl "Public"
  global        TRUE
  subsystem     "Component View"
  quidu         "5CFEDEB10316"
  logical_models (list unit_reference_list
    (object Class "user"
      quid        "5CFEE0B200F1"
      stereotype  "Actor"))

```

```

(object UseCase "run"
  quid      "5CFEE0B70391")
(object UseCase "Upload Video File"
  quid      "5CFEE0C003BF")
(object UseCase "Start CCTV Object Detection From Frames"
  quid      "5CFEE0D8014F")
(object UseCase "Upload CCTV Metadata to Hadoop"
  quid      "5CFEE0E40209")
(object UseCase "logout"
  quid      "5CFEE0F40120")
(object Association "$UNNAMED$0"
  quid      "5CFEE0FC00E1"
  roles     (list role_list
    (object Role "$UNNAMED$1"
      quid      "5CFEE0FC0344"
      supplier   "Logical View::run"
      quidu     "5CFEE0B70391"
      is_navigable TRUE)
    (object Role "$UNNAMED$2"
      quid      "5CFEE0FC0345"
      supplier   "Logical View::user"
      quidu     "5CFEE0B200F1"
      is_navigable TRUE)))
(object Association "$UNNAMED$3"
  quid      "5CFEE0FE03D1"
  roles     (list role_list
    (object Role "$UNNAMED$4"
      quid      "5CFEE0FF0227"
      supplier   "Logical View::Upload Video File"
      quidu     "5CFEE0C003BF"
      is_navigable TRUE)
    (object Role "$UNNAMED$5"
      quid      "5CFEE0FF0228"
      supplier   "Logical View::user"
      quidu     "5CFEE0B200F1"
      is_navigable TRUE)))
(object Association "$UNNAMED$6"
  quid      "5CFEE10200CE"
  roles     (list role_list
    (object Role "$UNNAMED$7"
      quid      "5CFEE1030023"
      supplier   "Logical View::Start CCTV Object Detection From Frames"
      quidu     "5CFEE0D8014F"
      is_navigable TRUE)
    (object Role "$UNNAMED$8"
      quid      "5CFEE1030024"

```

```

        supplier      "Logical View::user"
        quidu         "5CFEE0B200F1"
        is_navigable  TRUE)))
(object Association "$UNNAMED$9"
  quid              "5CFEE104013E"
  roles             (list role_list
    (object Role "$UNNAMED$10"
      quid          "5CFEE10500B0"
      supplier      "Logical View::Upload CCTV Metadata to Hadoop"
      quidu         "5CFEE0E40209"
      is_navigable  TRUE)
    (object Role "$UNNAMED$11"
      quid          "5CFEE10500B1"
      supplier      "Logical View::user"
      quidu         "5CFEE0B200F1"
      is_navigable  TRUE)))
(object Association "$UNNAMED$12"
  quid              "5CFEE10601BC"
  roles             (list role_list
    (object Role "$UNNAMED$13"
      quid          "5CFEE107019C"
      supplier      "Logical View::logout"
      quidu         "5CFEE0F40120"
      is_navigable  TRUE)
    (object Role "$UNNAMED$14"
      quid          "5CFEE107019D"
      supplier      "Logical View::user"
      quidu         "5CFEE0B200F1"
      is_navigable  TRUE))))
logical_presentations (list unit_reference_list
  (object ClassDiagram "Main"
    quid              "5CFEDEB20025"
    title             "Main"
    zoom              100
    max_height        28350
    max_width         21600
    origin_x          0
    origin_y          0
    items             (list diagram_item_list
      (object UseCaseView "Logical View::run" @1
        location       (912, 160)
        label          (object ItemLabel
          Parent_View  @1
          location      (912, 298)
          anchor_loc    1
          nlines        2

```



```

        max_width 633
        justify    0
        label      "run")
    icon_style     "Icon"
    line_color     3342489
    fill_color     12058623
    quidu          "5CFEE0B70391"
    height         118
    autoResize     TRUE)
(object UseCaseView "Logical View::Upload Video File" @2
    location       (912, 416)
    label          (object ItemLabel
        Parent_View @2
        location    (912, 554)
        anchor_loc 1
        nlines      2
        max_width   633
        justify     0
        label       "Upload Video File")
    icon_style     "Icon"
    line_color     3342489
    fill_color     12058623
    quidu          "5CFEE0C003BF"
    height         118
    autoResize     TRUE)
(object UseCaseView "Logical View::Start CCTV Object Detection From Frames"
@3
    location       (912, 688)
    label          (object ItemLabel
        Parent_View @3
        location    (912, 826)
        anchor_loc 1
        nlines      2
        max_width   633
        justify     0
        label       "Start CCTV Object Detection From Frames")
    icon_style     "Icon"
    line_color     3342489
    fill_color     12058623
    quidu          "5CFEE0D8014F"
    height         118
    autoResize     TRUE)
(object UseCaseView "Logical View::Upload CCTV Metadata to Hadoop" @4
    location       (912, 1008)
    label          (object ItemLabel
        Parent_View @4

```

```

        location      (912, 1146)
        anchor_loc    1
        nlines        2
        max_width     633
        justify       0
        label         "Upload CCTV Metadata to Hadoop")
    icon_style        "Icon"
    line_color        3342489
    fill_color        12058623
    quidu             "5CFEE0E40209"
    height            118
    autoResize        TRUE)
(object ClassView "Class" "Logical View::user" @5
    ShowCompartmentStereotypes    TRUE
    IncludeAttribute               TRUE
    IncludeOperation               TRUE
    location                       (176, 688)
    label                         (object ItemLabel
        Parent_View               @5
        location                   (176, 848)
        anchor_loc                 1
        nlines                     2
        max_width                 327
        justify                    0
        label                     "user")
    icon                         "Actor"
    icon_style                    "Icon"
    line_color                    3342489
    fill_color                    12058623
    quidu                         "5CFEE0B200F1"
    annotation                    8
    autoResize                     TRUE)
(object AssociationViewNew "$UNNAMED$0" @6
    location                      (529, 429)
    stereotype                     TRUE
    line_color                     3342489
    quidu                         "5CFEE0FC00E1"
    roleview_list (list RoleViews
        (object RoleView "$UNNAMED$1" @7
            Parent_View            @6
            location                (273, -227)
            stereotype              TRUE
            line_color              3342489
            quidu                  "5CFEE0FC0344"
            client                  @6
            supplier                @1

```

```
line_style    0)
(object RoleView "$UNNAMED$2" @8
  Parent_View @6
  location    (273, -227)
  stereotype  TRUE
  line_color  3342489
  quidu       "5CFEE0FC0345"
  client      @6
  supplier    @5
  line_style  0)))
```

Collaboration:

(object Petal

version 42
_written "Rose 4.5.8054a"
charSet 0)

(object Design "Logical View"

is_unit TRUE
is_loaded TRUE
quid "5CFEE1B300FD"
defaults (object defaults
rightMargin 0.250000
leftMargin 0.250000
topMargin 0.250000
bottomMargin 0.500000
pageOverlap 0.250000
clipIconLabels TRUE
autoResize TRUE
snapToGrid TRUE
gridX 16
gridY 16
defaultFont (object Font
size 10
face "Arial"
bold FALSE
italics FALSE
underline FALSE
strike FALSE
color 0
default_color TRUE)
showMessageNum 1
showClassOfObject TRUE
notation "Unified")

root_usecase_package (object Class_Category "Use Case View"

quid "5CFEE1B300FF"
exportControl "Public"
global TRUE
logical_models (list unit_reference_list)
logical_presentations (list unit_reference_list
(object UseCaseDiagram "Main"
quid "5CFEE1B3016B"
title "Main"
zoom 100
max_height 28350
max_width 21600

```

        origin_x      0
        origin_y      0
        items          (list diagram_item_list)))
root_category        (object Class_Category "Logical View"
  quid                "5CFEE1B300FE"
  exportControl       "Public"
  global              TRUE
  subsystem           "Component View"
  quidu               "5CFEE1B30100"
  logical_models      (list unit_reference_list
    (object Mechanism @1
      logical_models   (list unit_reference_list
        (object Object "User"
          quid          "5CFEE1BF0321"
          collaborators (list link_list
            (object Link
              quid       "5CFEE1E10001"
              supplier    "Database"
              quidu       "5CFEE1D600F0"
              messages    (list Messages
                (object Message "run"
                  quid        "5CFEE1E10002"
                  frequency    "Aperiodic"
                  synchronization "Simple"
                  dir          "FromClientToSupplier"
                  sequence      ""
                  ordinal       0)
                (object Message "uploadVideoFile"
                  quid        "5CFEE1F202D3"
                  frequency    "Aperiodic"
                  synchronization "Simple"
                  dir          "FromClientToSupplier"
                  sequence      ""
                  ordinal       1)
                (object Message "startCCTVobjectDetection"
                  quid        "5CFEE20C00CE"
                  frequency    "Aperiodic"
                  synchronization "Simple"
                  dir          "FromClientToSupplier"
                  sequence      ""
                  ordinal       2)
                (object Message "uploadCCTVMetadataToHadoop"
                  quid        "5CFEE21F0324"
                  frequency    "Aperiodic"
                  synchronization "Simple"
                  dir          "FromClientToSupplier"

```

```

sequence      ""
ordinal        3)))
(object Link
  quid          "5CFEE2390393"
  supplier      "User"
  quidu         "5CFEE1BF0321"
  messages      (list Messages
    (object Message "logout"
      quid        "5CFEE2390394"
      frequency    "Aperiodic"
      synchronization "Simple"
      dir          "FromClientToSupplier"
      sequence      ""
      ordinal        4))))
  persistence    "Transient"
  multi          FALSE)
(object Object "Database"
  quid          "5CFEE1D600F0"
  persistence    "Transient"
  multi          FALSE))))
(object ObjectDiagram "sequence"
  mechanism_ref  @1
  quid          "5CFEE2560304"
  title         "sequence"
  zoom          100
  max_height     28350
  max_width      21600
  origin_x       0
  origin_y       0
  items          (list diagram_item_list
    (object ObjectView "User" @23
      location     (400, 560)
      font          (object Font
        underline   TRUE)
      label         (object ItemLabel
        Parent_View @23
        location     (319, 509)
        fill_color    12058623
        nlines        2
        max_width     162
        justify       0
        label         "User")
      icon_style    "Icon"
      line_color     3342489
      fill_color     12058623
      quidu         "5CFEE1BF0321"

```

```

height      126
annotation  4
autoResize  TRUE
subobjects  0
x_offset    0.000000
y_offset    0.000000)
(object ObjectView "Database" @24
  location   (1648, 560)
  font       (object Font
    underline TRUE)
  label      (object ItemLabel
    Parent_View @24
    location    (1567, 509)
    fill_color  12058623
    nlines     2
    max_width   162
    justify    0
    label      "Database")
  icon_style "Icon"
  line_color  3342489
  fill_color  12058623
  quidu       "5CFEE1D600F0"
  height      126
  annotation  4
  autoResize  TRUE
  subobjects  0
  x_offset    0.000000
  y_offset    0.000000)
(object LinkSelfView "" @25
  location   (400, 347)
  stereotype TRUE
  line_color  3342489
  quidu       "5CFEE2390393"
  client      @23
  supplier    @23
  line_style  0)
(object MessView "" @26
  location   (400, 303)
  line_color  3342489
  label      (object SegLabel @27
    Parent_View @26
    location    (400, 259)
    quidu       "5CFEE2390394"
    anchor_loc  1
    nlines     1
    max_width   150

```

```

        justify      0
        label        "logout"
        pctDist      0.500000
        height       45
        orientation  0)
    object_arc      @25
    pctDist          0.500000
    height           45
    orientation      0
    dir              1
    origin            (340, 303)
    terminus         (460, 303))
(object LinkView "" @28
    stereotype      TRUE
    line_color       3342489
    quidu            "5CFEE1E10001"
    client           @23
    supplier         @24
    line_style       0)
(object MessView "" @29
    location          (1024, 516)
    line_color        3342489
    label             (object SegLabel @30
        Parent_View  @29
        location      (1024, 331)
        quidu         "5CFEE1E10002"
        anchor_loc    1
        nlines        1
        max_width     100
        justify       0
        label         "run"
        pctDist       0.500000
        height        45
        orientation   0)
    label            (object SegLabel @31
        Parent_View  @29
        location      (1024, 378)
        quidu         "5CFEE1F202D3"
        anchor_loc    1
        nlines        1
        max_width     322
        justify       0
        label         "uploadVideoFile"
        pctDist       0.500000
        height        45
        orientation   0)

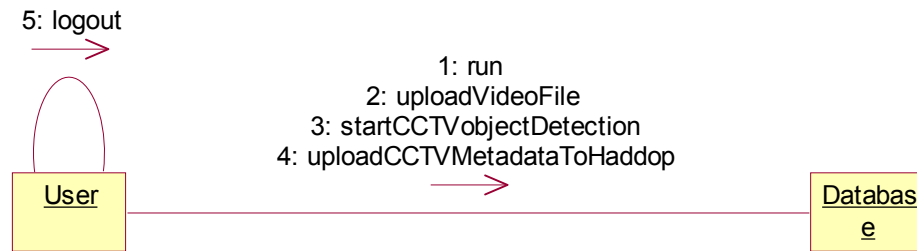
```



```

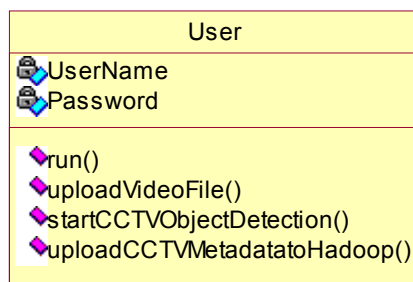
        label      (object SegLabel @32
          Parent_View      @29
          location  (1024, 425)
          quidu     "5CFEE20C00CE"
          anchor_loc 1
          nlines    1
          max_width 497
          justify   0
          label     "startCCTVobjectDetection"
          pctDist   0.500000
          height    45
          orientation 0)
        label      (object SegLabel @33
          Parent_View      @29
          location  (1024, 472)
          quidu     "5CFEE21F0324"
          anchor_loc 1
          nlines    1
          max_width 600
          justify   0
          label     "uploadCCTVMetadataToHaddop"
          pctDist   0.500000
          height    45
          orientation 0)
        object_arc  @28
        pctDist     0.500000
        height      45
        orientation  0
        dir         1
        origin      (964, 516)
        terminus    (1084, 516))))))
root_subsystem  (object SubSystem "Component View"
  quid          "5CFEE1B30100"
  physical_models (list unit_reference_list)
  physical_presentations (list unit_reference_list
    (object Module_Diagram "Main"
      quid      "5CFEE1B3016A"
      title     "Main"
      zoom      100
      max_height 28350
      max_width  21600
      origin_x   0
      origin_y   0
      items      (list diagram_item_list))))))

```



CLASS DIAGRAM:

In software engineering, a 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 the classes. It explains which class contains information.



Class:

(object Petal

version 42
 _written "Rose 4.5.8054a"
 charSet 0)

(object Design "Logical View"

is_unit TRUE
 is_loaded TRUE
 quid "5CFEE1320222"
 defaults (object defaults
 rightMargin 0.250000
 leftMargin 0.250000
 topMargin 0.250000
 bottomMargin 0.500000
 pageOverlap 0.250000

```

clipIconLabels      TRUE
autoResize          TRUE
snapToGrid          TRUE
gridX               16
gridY               16
defaultFont          (object Font
  size              10
  face              "Arial"
  bold              FALSE
  italics           FALSE
  underline         FALSE
  strike            FALSE
  color             0
  default_color     TRUE)
showMessageNum      1
showClassOfObject   TRUE
notation            "Unified")
root_usecase_package (object Class_Category "Use Case View"
  quid              "5CFEE1320224"
  exportControl     "Public"
  global            TRUE
  logical_models     (list unit_reference_list)
  logical_presentations (list unit_reference_list
    (object UseCaseDiagram "Main"
      quid          "5CFEE1320281"
      title         "Main"
      zoom          100
      max_height    28350
      max_width     21600
      origin_x      0
      origin_y      0
      items         (list diagram_item_list))))
root_category        (object Class_Category "Logical View"
  quid              "5CFEE1320223"
  exportControl     "Public"
  global            TRUE
  subsystem         "Component View"
  quidu            "5CFEE1320225"
  logical_models     (list unit_reference_list
    (object Class "User"
      quid          "5CFEE13403C9"
      operations     (list Operations
        (object Operation "run"
          quid        "5CFEE140012E"
          concurrency  "Sequential"
          opExportControl "Public"

```

```

uid          0)
(object Operation "uploadVideoFile"
  quid        "5CFEE1440100"
  concurrency  "Sequential"
  opExportControl "Public"
  uid          0)
(object Operation "startCCTVObjectDetection"
  quid        "5CFEE15D007C"
  concurrency  "Sequential"
  opExportControl "Public"
  uid          0)
(object Operation "uploadCCTVMetadatatoHadoop"
  quid        "5CFEE17C000D"
  concurrency  "Sequential"
  opExportControl "Public"
  uid          0))
class_attributes (list class_attribute_list
  (object ClassAttribute "UserName"
    quid        "5CFEE13900BF")
  (object ClassAttribute "Password"
    quid        "5CFEE13B0303"))))
logical_presentations (list unit_reference_list
  (object ClassDiagram "Main"
    quid        "5CFEE1320241"
    title        "Main"
    zoom         100
    max_height    28350
    max_width     21600
    origin_x      0
    origin_y      0
    items         (list diagram_item_list
      (object ClassView "Class" "Logical View::User" @1
        ShowCompartmentStereotypes TRUE
        IncludeAttribute TRUE
        IncludeOperation TRUE
        location (816, 736)
        label (object ItemLabel
          Parent_View @1
          location (504, 530)
          fill_color 12058623
          nlines 1
          max_width 624
          justify 0
          label "User")
          icon_style "Icon"
          line_color 3342489

```

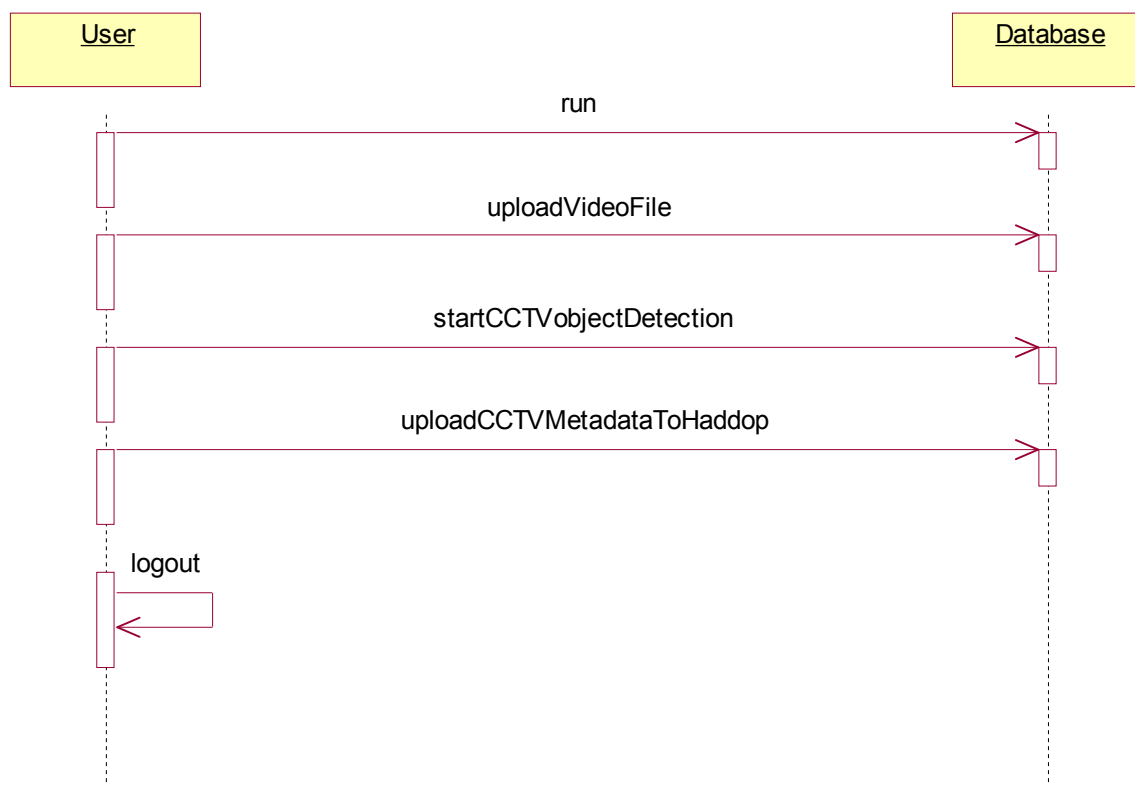
```

        fill_color      12058623
        quidu           "5CFEE13403C9"
        compartment     (object Compartment
            Parent_View   @1
            location      (504, 590)
            icon_style    "Icon"
            fill_color    12058623
            anchor        2
            nlines        7
            max_width     599)
        width           642
        height          436
        annotation       8
        autoResize      TRUE))))))
root_subsystem         (object SubSystem "Component View"
    quid                "5CFEE1320225"
    physical_models      (list unit_reference_list)
    physical_presentations (list unit_reference_list
        (object Module_Diagram "Main"
            quid          "5CFEE1320280"
            title         "Main"
            zoom          100
            max_height    28350
            max_width     21600
            origin_x      0
            origin_y      0
            items          (list diagram_item_list))))))
process_structure       (object Processes
    quid                "5CFEE1320226"
    ProcsNDevs          (list
        (object Process_Diagram "Deployment View"
            quid          "5CFEE1320228"
            title         "Deployment View"
            zoom          100
            max_height    28350
            max_width     21600
            origin_x      0
            origin_y      0
            items          (list diagram_item_list))))))
properties             (object Properties
    quid                "5CFEE1320227"))

```

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



SEQUENCE:

(object Petal

version 42
_written "Rose 4.5.8054a"
charSet 0)

(object Design "Logical View"

is_unit TRUE
is_loaded TRUE
quid "5CFEE1B300FD"
defaults (object defaults
rightMargin 0.250000

```

leftMargin    0.250000
topMargin     0.250000
bottomMargin  0.500000
pageOverlap   0.250000
clipIconLabels TRUE
autoResize    TRUE
snapToGrid    TRUE
gridX         16
gridY         16
defaultFont   (object Font
  size        10
  face        "Arial"
  bold        FALSE
  italics     FALSE
  underline   FALSE
  strike      FALSE
  color       0
  default_color TRUE)
showMessageNum 1
showClassOfObject TRUE
notation       "Unified")
root_usecase_package (object Class_Category "Use Case View"
  quid            "5CFEE1B300FF"
  exportControl   "Public"
  global          TRUE
  logical_models  (list unit_reference_list)
  logical_presentations (list unit_reference_list
    (object UseCaseDiagram "Main"
      quid            "5CFEE1B3016B"
      title           "Main"
      zoom            100
      max_height      28350
      max_width       21600
      origin_x        0
      origin_y        0
      items            (list diagram_item_list))))
root_category   (object Class_Category "Logical View"
  quid            "5CFEE1B300FE"
  exportControl   "Public"
  global          TRUE
  subsystem       "Component View"
  quidu           "5CFEE1B30100"
  logical_models  (list unit_reference_list
    (object Mechanism @1
      logical_models  (list unit_reference_list
        (object Object "User"

```

```

quid          "5CFEE1BF0321"
collaborators (list link_list
  (object Link
    quid          "5CFEE1E10001"
    supplier      "Database"
    quidu         "5CFEE1D600F0"
    messages      (list Messages
      (object Message "run"
        quid          "5CFEE1E10002"
        frequency      "Aperiodic"
        synchronization "Simple"
        dir            "FromClientToSupplier"
        sequence        ""
        ordinal         0)
      (object Message "uploadVideoFile"
        quid          "5CFEE1F202D3"
        frequency      "Aperiodic"
        synchronization "Simple"
        dir            "FromClientToSupplier"
        sequence        ""
        ordinal         1)
      (object Message "startCCTVObjectDetection"
        quid          "5CFEE20C00CE"
        frequency      "Aperiodic"
        synchronization "Simple"
        dir            "FromClientToSupplier"
        sequence        ""
        ordinal         2)
      (object Message "uploadCCTVMetadataToHadoop"
        quid          "5CFEE21F0324"
        frequency      "Aperiodic"
        synchronization "Simple"
        dir            "FromClientToSupplier"
        sequence        ""
        ordinal         3)))
Focus_Of_Control (object Focus_Of_Control "" @5
  location      (528, 688)
  line_color    3342489
  InterObjView  @2
  height        120
  y_coord       60
  Nested        FALSE)
Focus_Of_Control (object Focus_Of_Control "" @6
  location      (528, 848)
  line_color    3342489
  InterObjView  @2

```



```

        height      120
        y_coord     60
        Nested      FALSE)
Focus_Of_Control   (object Focus_Of_Control "" @7
        location    (528, 1040)
        line_color  3342489
        InterObjView @2
        height      152
        y_coord     92
        Nested      FALSE))
(object InterObjView "Database" @8
        location    (2000, 224)
        font        (object Font
        underline   TRUE)
        label       (object ItemLabel
        Parent_View @8
        location    (2000, 224)
        fill_color  12058623
        anchor_loc  1
        nlines      2
        max_width   282
        justify     0
        label       "Database")
        icon_style  "Icon"
        line_color  3342489
        fill_color  12058623
        quidu       "5CFEE1D600F0"
        width       300
        height      1057
        icon_height  0
        icon_width  0
        annotation  1
Focus_Of_Control   (object Focus_Of_Control "" @9
        location    (2000, 352)
        line_color  3342489
        InterObjView @8
        height      60
        y_coord     0
        Nested      FALSE)
Focus_Of_Control   (object Focus_Of_Control "" @10
        location    (2000, 512)
        line_color  3342489
        InterObjView @8
        height      60
        y_coord     0
        Nested      FALSE)

```

```

Focus_Of_Control    (object Focus_Of_Control "" @11
  location    (2000, 688)
  line_color  3342489
  InterObjView    @8
  height       60
  y_coord      0
  Nested       FALSE)
Focus_Of_Control    (object Focus_Of_Control "" @12
  location    (2000, 848)
  line_color  3342489
  InterObjView    @8
  height       60
  y_coord      0
  Nested       FALSE))
(object InterMessView "" @13
  location    (0, 352)
  label       (object SegLabel @14
    Parent_View    @13
    location    (1264, 308)
    quidu       "5CFEE1E10002"
    anchor_loc  1
    nlines      1
    max_width   53
    justify     0
    label       "run"
    pctDist     0.500000
    height      45
    orientation  0)
  line_color  3342489
  client      @2
  supplier    @8
  Focus_Src   @3
  Focus_Entry @9
  origin      (544, 352)
  terminus    (1984, 352)
  ordinal     0)
(object InterMessView "" @15
  location    (0, 512)
  label       (object SegLabel @16
    Parent_View    @15
    location    (1262, 470)
    quidu       "5CFEE1F202D3"
    anchor_loc  1
    nlines      1
    max_width   278
    justify     0

```

```

        label      "uploadVideoFile"
        pctDist    0.499306
        height     43
        orientation 0)
    line_color     3342489
    client          @2
    supplier        @8
    Focus_Src       @4
    Focus_Entry     @10
    origin          (544, 512)
    terminus       (1984, 512)
    ordinal         1)
(object InterMessView "" @17
    location        (0, 688)
    label           (object SegLabel @18
        Parent_View @17
        location     (1264, 644)
        quidu        "5CFEE20C00CE"
        anchor_loc   1
        nlines       1
        max_width    450
        justify      0
        label        "startCCTVObjectDetection"
        pctDist      0.500000
        height       45
        orientation  0)
    line_color     3342489
    client          @2
    supplier        @8
    Focus_Src       @5
    Focus_Entry     @11
    origin          (544, 688)
    terminus       (1984, 688)
    ordinal         2)
(object InterMessView "" @19
    location        (0, 848)
    label           (object SegLabel @20
        Parent_View @19
        location     (1264, 804)
        quidu        "5CFEE21F0324"
        anchor_loc   1
        nlines       1
        max_width    553
        justify      0
        label        "uploadCCTVMetadataToHadoop"
        pctDist      0.500000

```

```

        height      45
        orientation  0)
        line_color   3342489
        client        @2
        supplier      @8
        Focus_Src     @6
        Focus_Entry   @12
        origin        (544, 848)
        terminus      (1984, 848)
        ordinal       3)
    (object SelfMessView "" @21
        location      (0, 1072)
        label          (object SegLabel @22
            Parent_View @21
            location     (619, 1028)
            quidu        "5CFEE2390394"
            anchor_loc   1
            nlines       1
            max_width    106
            justify      0
            label         "logout"
            pctDist      0.500000
            height       45
            orientation  0)
            line_color   3342489
            client        @2
            supplier      @2
            Focus_Src     @7
            origin        (544, 1072)
            terminus      (694, 1072)
            ordinal       4))))
    root_subsystem  (object SubSystem "Component View"
        quid        "5CFEE1B30100"
        physical_models (list unit_reference_list)
        physical_presentations (list unit_reference_list
            (object Module_Diagram "Main"
                quid        "5CFEE1B3016A"
                title       "Main"
                zoom        100
                max_height   28350
                max_width    21600
                origin_x     0
                origin_y     0
                items        (list diagram_item_list))))
    process_structure (object Processes
        quid          "5CFEE1B30101"

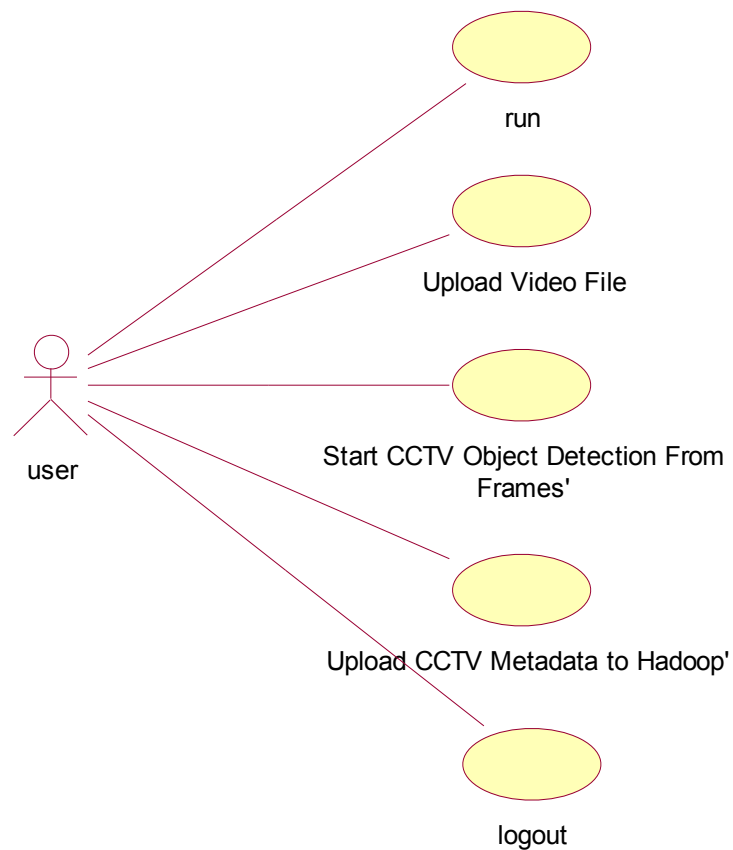
```

```

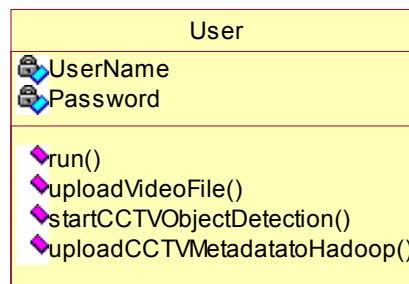
ProcsNDevs (list
  (object Process_Diagram "Deployment View"
    quid "5CFEE1B30103"
    title "Deployment View"
    zoom 100
    max_height 28350
    max_width 21600
    origin_x 0
    origin_y 0
    items (list diagram_item_list)))
properties (object Properties quid "5CFEE1B30102"))

```

4.2.1 USE CASE DIAGRAM:



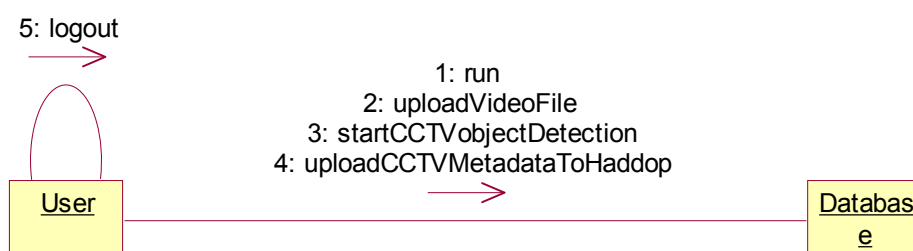
4.2.2 CLASS DIAGRAM:



4.2.3 SEQUENCE DIAGRAM:



4.2.4 COLLABORATIVE DIAGRAM:



4.3. INPUT AND OUTPUT DESIGN

INPUT DESIGN

Input design is part of overall system design that requires special attention designing input data is to make the data entered easy and free from errors. The input forms are designed using the controls available in .NET framework. Validation is made for each and every data that is entered. Help information is provided for the users during when the customer feels difficult.

Input design is the process of converting the user originated inputs to a computer based format. A system user interacting through a workstation must be able to tell the system whether to accept the input to produce reports. The collection of input data is considered to be most expensive part of the system design. Since the input has to be planned in such a manner so as to get relevant information, extreme care is taken to obtain pertinent information.

This project first will entered to the input of allocation forms it will be created on student details form and subject entry form, time table form .it will helps to calculate subject wise attendance system. Next one if u wants any verification on your data's also available in details show forms. Attendance to entered single subject wise or all subject wise attendance system available in this project

OUTPUT DESIGN

Output design this application “Student Attendance management system”generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

As the outputs are the most important sources of information to the users, better design should improve the system's relationships with us and also will help in decisionmaking. Form design elaborates the way output is presented and the layout available for capturing information.

One of the most important factors of the system is the output it produces. This system refers to the results and information generated. Basically the output from a computer system is used to communicate the result of processing to the user.

Attendance management system to show the report subject wise attendance maintaining by staffs. Taken as a whole report obtain on a administrator privileges only. this forms will show weekly report and consolidate report generated date, batch, and class wise to our end user. we want to change our report to convert Excel format .if you want change any modification.

5 : IMPLEMENTATION

5. SOFTWARE ENVIRONMENT

5.1 .NETBEANS:

What is Netbeans?

This document provides a list of *NetBeans APIs* with a short description of what they are used for, and a table describing different types of interfaces (see [What is an API?](#) to understand why we list DTDs, file formats, etc.) and with a stability category (stable and official, [under development](#), [deprecated](#), [friend or private](#); see [API stability](#) for more info). The aim is to provide as detailed a definition of NetBeans module external interfaces as possible and give other developers a chance to decide whether they want to depend on a particular API or not.

Some of these APIs are part of the [NetBeans Platform](#) (for example lookup, loaders, utilities, nodes, explorer, window systems, multiview, etc.), some of them are specific to [NetBeans IDE](#) (projects, javacore, diff, etc.) and some are not included in the release at all and are just provided for download (usually via autoupdate). Basically when building an application based on *NetBeans* one is free to choose the set of modules and their APIs to satisfy one's needs.

To get the API of your module listed here, see the documentation for the Javadoc building [infrastructure](#).

Can't find what you're looking for? Try the [Apache NetBeans website](#).

- [Actions APIs](#) - Actions provides system of support and utility classes for 'actions' usage in NetBeans.
- [Ant](#) - The Ant integration module recognizes Ant build scripts, facilitates editing them as text or structurally, lets the user run targets or create shortcuts to them, etc.
- [Ant-Based Project Support](#) - Provides the basic infrastructure by which Ant-based projects can be created, read and write configuration parameters and properties from/to disk, satisfy common queries and interfaces, etc.
- [Ant-Based Project Support UI](#) - Provides the basic infrastructure by which Ant-based projects can be created, read and write configuration parameters and properties from/to disk, satisfy common queries and interfaces, etc.
- [Auto Update Services](#) - [org.netbeans.api.autoupdate](#) [org.netbeans.spi.autoupdate](#) .
- [Auto Update UI](#) - [AutoUpdateUI](#) .
- [Base Project UI API](#) - The module supplies the APIs for the basic, generic UI infrastructure for projects: list of opened projects, main project, basic project-sensitive actions, template wizards, etc.
- [Base Utilities API](#) - Described in the overall answer.
- [Classpath APIs](#) - Models basic aspects of the metadata surrounding list of source roots, such as the classpath.
- [Command Line Parsing API](#) - GetOpts like infrastructure to parse command line arguments with the cooperative participation of various modules.

- **Common Annotations** - Provides common annotations serving as a documentation element and for static code analysis.
- **Common Palette** - The project implements a new component palette that will be reused by other projects.
- **Core - IDE** - **core.ide** .
- **Database Explorer** - This project provides access to objects defined in the Database Explorer.
- **Datatypes API** - In summary, the **LoadersAPI** is responsible for scanning files in a directory on disk, weeding out irrelevant files of no interest to the IDE, and grouping the rest into logical chunks, or just determining what type of data each represents.
- **Debugger Core API** - The debuggercore/api module (Debugger Core API) allows to install different debugger implementation to one IDE.
- **Debugger Core - UI** - The debuggercore module (Debugger Core UI) contains shared UI components for all debugger implementations, and defines some SPI for sharing of them.
- **Dialogs API** - The **DialogsAPI** allows creating a user notification, a dialog's description and also permits it to be displayed.
- **Diff** - The diff module provides the ability to visualize differences between source files.
- **Editor Code Completion** - Code Completion provides users with a list of suggested completions for partially typed texts in the editor and various dialog input fields.
- **Editor Code Folding** - The Code Folding is part of the editor module functionality and it's responsible for hiding of the portions of the code that are less important for the user at the given time.
- **Editor Code Templates** - Code Templates allow to paste various code snippets by using parametrized text.
- **Editor Error Stripe** - The Error Stripe shows an overview of important information of an edited source code.
- **Editor Guarded Sections** - Guarded Sections protects user from modifying document content.
- **Editor Library 2** - The Editor Library 2 module is a set of official APIs and SPIs, designed to replaces the original Editor Library with legacy APIs that are not properly structured and do not conform to the rules implied on the current NB APIs.
- **Editor Settings** - The legacy settings system in the editor module is complicated, error prone and hard to use.
- **Execution API** - The IDE uses a small interface to describe the execution of Java-based classes together with arguments.
- **Explorer & Property Sheet API** - The **ExplorerAPI** is build around Explorer - solely a user-interface device: it has no particular knowledge of the structure of the IDE.
- **External Execution API** - Provides common APIs to execute external process in the IDE to handle its streams and present the output to the user.
- **External Libraries** - Permits libraries to be defined, customized, and stored by the user for reuse in multiple projects.
- **External Libraries UI** - XXX no answer for arch-what .
- **File System API** - The Filesystems API provides a common API to access files in a uniform manner.
- **File Templates** - This utility standardizes the process to use files as blueprints to create new files.
- **General Queries API** - General kinds of queries between modules.
- **GraalVM SDK API** - Bridge that plugs GraalVM languages into Scripting.

- **HTML UI API** - NetBeans Platform specific bindings over standard HTML for Java API.
- **I/O API and SPI** - The Input/Output API and SPI is a small module which contains InputOutput and related interfaces used in driving the Output Window.
- **I/O APIs - Swing** - The Input/Output API is a small API module which contains InputOutput and related interfaces used in driving the Output Window.
- **Intent API** - This module provides a contract between API clients that can express some intention to invoke an operation and SPI providers that can handle that intention.
- **JavaHelp Integration** - The JavaHelp integration API wraps the standard JavaHelp extension library.
- **Java Platform** - Many Java-based project types need to be able to configure the version and location of Java to be used when building and running the project.
- **Java Platform UI** - Many Java-based project types need to be able to configure the version and location of Java to be used when building and running the project.
- **Java Project Support** - Provides support infrastructure for projects working with the Java language.
- **Java Project Support UI** - Provides support infrastructure for projects working with the Java language.
- **Java Source Base** - XXX no answer for arch-what .
- **Java Source UI** - **java.sourceui** .
- **Java Support APIs** - Provides java specific queries (javadc, source level) used by other modules like java language infrastructure.
- **JPDA Debugger API** - The debuggerjpda/api (Debugger JPDA API) defines API for NetBeans Java Debugger.
- **Keyring API** - XXX no answer for arch-what .
- **[Knockout Model Provider API](#)** - **api.knockout** .
- **Lexer** - Lexer module provides token lists for various text inputs.
- **Lookup API** - Described in the overall answer.
- **Maven API** - XXX no answer for arch-what .
- **MIME Lookup API** - Each editor provides an EditorKit which controls the policy of specific MIME content type.
- **Module System API** - The Modules API lies at the core of NetBeans and describes how plug-in modules are added and managed.
- **MultiView Windows** - Multi views are general mechanism for displaying several perspectives, or views of data, which will be used consistently across whole IDE window system.
- **Navigator API** - Navigator module is a base API module which provides: A place for modules to show structure/outline of their documents Ability for modules to show their view only when special document(node) is active in the system UI for switching between multiple views available for currently active document(node) Coalescing of fast coming selected node changes to show content for .
- **Nodes API** - Nodes API serves as the main apparatus for visualisation of objects in NetBeans.
- **Options Dialog and SPI** - This module contains implementation of Options Panel and simple SPI.
- **Parsing API** - See Parsing API homepage for project requirements, motivation.
- **Parsing API Indexing** - See Parsing API homepage for project requirements, motivation.
- **Print** - Provides print functionality in NetBeans.

- **Progress API** - The progress API is good for tracking progress of long lasting tasks in the IDE.
- **Progress API - Swing** - XXX no answer for arch-what .
- **Project API** - Provides a generic infrastructure for modelling projects.
- **Project UI API** - The module supplies the APIs for the basic, generic UI infrastructure for projects: list of opened projects, main project, basic project-sensitive actions, template wizards, etc.
- **Quick Search API** - QuickSearch API and its implementations provides way for end user to learn system earlier and easier and also to find various things in the system more quickly, conveniently and in standard way.
- **Scripting API Wrapper** - NetBeans specific extensions to JDK's Scripting API can be found in this module.
- **Search API** - This project is good for implementators of nodes to define how files under this node should be searched.
- **Settings API** - Settings are persistent objects with a set of properties, that should be available via lookup.
- **Task List API** - This module provides SPI for third parties that want to display some sort information for the user in the Task List / Problems window.
- **Text API** - The **EditorAPI** is used for accessing editor-related functionality from within the IDE for use by other modules and the core IDE itself.
- **TreeTableView Model** - The debuggercore/**ViewModel** module (View Model) allows to share one TreeTableView among different modules.
- **UI Utilities API** - The **org.openide.awt** provides API/SPI for UI related aspects of application.
- **Utilities API** - Described in the overall answer.
- **Visual Library API** - The Visual Library 2.0 is the next generation of the original Graph Library 1.0.
- **Window System API** - Window System API is used to display and control application GUI: Main window, frames, components.
- **XML Tools API** - It helps with editing XML files and supports leveraging XML in developed application.
- **org.openide-filesystems-nb** - no API description provided (see [how to do it](#))

Ant

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The Ant integration module recognizes Ant build scripts, facilitates editing them as text or structurally, lets the user run targets or create shortcuts to them, etc.

Usage:

Nothing.

Common Annotations

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Provides common annotations serving as a documentation element and for static code analysis.

Usage:

Nothing.



Debugger Core API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The debuggercore/api module (Debugger Core API) allows to install different debugger implementation to one IDE. It allows to share some common UI components.

Usage:

Nothing.

Interface Name	Stability Classification	Specified in What Document?
DebuggerCoreAPI	Official	https://bits.netbeans.org/12.0/javadoc/org-netbeans-api-debugger

JPDA Debugger API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The debuggerjpda/api (Debugger JPDA API) defines API for NetBeans Java Debugger.

Usage:

Nothing.

HTML UI API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

NetBeans Platform specific bindings over standard HTML for Java API.

Usage:

The HTML for Java & NetBeans API is a classical NetBeans module. However it depends on other modules provided by the **HTML for Java** project and those are OSGi bundles. As such, when one decides to use this module, one needs to turn on an OSGi container inside of the NetBeans Platform. It can be either **Felix** or NetBeans version of **Equinox**.

Interface Name	Stability Classification	Specified in What Document?
HTMLUI	Official	NetBeans Platform specific bindings over standard HTML for Java API.

Intent API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

This module provides a contract between API clients that can express some intention to invoke an operation and SPI providers that can handle that intention. This is useful in client-server environments, where the intention can be constructed on server-side, but handled on client-side. The objects that describe the intention should be easy to construct, transfer and interpret.

Usage:

Standard module dependency is sufficient.

Interface Name	Stability Classification	Specified in What Document?
IntentAPI	Official	https://bits.netbeans.org/12.0/javadoc/org-netbeans-api-intent/org/netbeans/api/intent/package-summary.html API for describing and executing intended operations.

IntentHandlerSPI	Official	https://bits.netbeans.org/12.0/javadoc/org-netbeans-api-intent/org/netbeans/spi/intent/package-summary.html SPI for handlers that are able to invoke proper operation for specified intents.
----------------------------------	--------------------------	---

I/O API and SPI

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The Input/Output API and SPI is a small module which contains InputOutput and related interfaces used in driving the Output Window. The normal implementation is `org.netbeans.core.output2`.

Usage:

Normal module dependency is enough.

Availability of some implementation of the SPI is guaranteed by "OpenIDE-Module-Needs: `org.netbeans.spi.io.InputOutputProvider`" in the manifest of this module.

Interface Name	Stability Classification	Specified in What Document?
NbInputOutputAPI	Official	https://bits.netbeans.org/12.0/javadoc/org-netbeans-api-io/org/netbeans/api/io/package-summary.html The module contains APIs for creating output panes (e.g. output tabs in Output Window in the IDE) and for writing data into them. It also supports some advanced techniques, e.g. color text, hyperlinks, code folding, scrolling to stored positions.
NbInputOutputSPI	Official	https://bits.netbeans.org/12.0/javadoc/org-netbeans-api-io/org/netbeans/spi/io/package-summary.html SPI for providing custom implementations of output window is also included in this module, in package <code>org.netbeans.spi.io</code>


Java Support APIs

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Provides java specific queries (javadc, source level) used by other modules like java language infrastructure. More information in the Javadoc.

Usage:

Nothing.



Classpath APIs

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Models basic aspects of the metadata surrounding list of source roots, such as the classpath. More information in the Javadoc.

Usage:

Nothing.




Knockout Model Provider API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

api.knockout

Usage:

XXX no answer for deploy-dependencies



Maven API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

XXX no answer for arch-what

Usage:

No special requirements here.

Interface Name	Stability Classification	Specified in What Document?
MavenArchetypeAPI	Stable	provides miscellaneous APIs related to operations provided by maven-archetype-plugin. See package description for more details.

Progress API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The progress API is good for tracking progress of long lasting tasks in the IDE.

Usage:

Nothing.



Progress API - Swing

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

XXX no answer for arch-what

Usage:

XXX no answer for deploy-dependencies

Scripting API Wrapper

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

NetBeans specific extensions to JDK's Scripting API can be found in this module.

Usage:

XXX no answer for deploy-dependencies

Search API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

This project is good for implementators of nodes to define how files under this node should be searched. For example, if you implement a custom project type, you can define which folders should be searched when the project is in the current search scope. It is also good for people who want to add a custom tab into the "Search in projects" dialog. For example, implementators of platform applications can add form with criteria for searching in a database.

Usage:

No extra declaration is required.

File Templates

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

This utility standardizes the process to use files as blueprints to create new files.

Usage:

No specific deploy dependencies.

Interface Name	Stability Classification	Specified in What Document?
DataSystemsAPI	Official	https://bits.netbeans.org/12.0/javadoc/org-openide-loaders/index.html

Visual Library API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The Visual Library 2.0 is the next generation of the original Graph Library 1.0. It is designed for a general visualization with a support for graph-oriented modeling. Its focus is to become a part of the NetBeans platform and unify the visualization (UI and API) used in NetBeans-Platform-based applications. See <http://graph.netbeans.org/> web-site for details. See documentation for complete set of use-cases and code-snippets.

Usage:

No. Just set a normal project dependency on org.netbeans.api.visual (spec no.: 2.0) module.

XML Tools API

[javadoc](#) | [download](#) | [architecture](#)

It helps with editing XML files and supports leveraging XML in developed application.

Core - IDE

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

core.ide

Usage:

XXX no answer for deploy-dependencies

Interface Name	Stability Classification	Specified in What Document?
----------------	--------------------------	-----------------------------

[core.ide](#)

[Stable](#)

Right now this module provides handy **ServicesTabNodeRegistration** accompanied with few UI elements.

MultiView Windows

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Multi views are general mechanism for displaying several perspectives, or views of data, which will be used consistently across whole IDE window system. Using multi views will enable modules to visually represent data document or object in unified manner at one specific place in the system, yet still allowing for different perspectives of data viewing.

Usage:

Nothing.

GraalVM SDK API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Bridge that plugs GraalVM languages into Scripting.createManager() can be found in this module.

Usage:

XXX no answer for deploy-dependencies

Interface Name	Stability Classification	Specified in What Document?
org.graalvm.polyglot.wrapper	Stable	This module re-exports all found GraalVM languages under <code>GraalVM:lang</code> name and makes them accessible via NetBeans Scripting API.
org.graalvm.polyglot	Third Party	This module re-exports <code>org.graalvm.polyglot</code> APIs. Use them to obtain access to the GraalVM directly, if you only want to work with them and generic Scripting wrapper isn't enough.

Auto Update Services

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

org.netbeans.api.autoupdate org.netbeans.spi.autoupdate

Usage:

Interface Name	Stability Classification	Specified in What Document?
org.netbeans.api.autoupdate	Official	org-netbeans-modules-autoupdate-services/overview-summary.html
org.netbeans.spi.autoupdate	Official	org-netbeans-modules-autoupdate-services/overview-summary.html

Auto Update UI

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

AutoUpdateUI

Usage:

You modules can use the `AutoUpdateUI` API directly (of course you almost certainly need also [Autoupdate Services](#)), just be aware that this module comes with a UI. It is not always practical to depend on modules that provide a UI from low level infrastructure modules - if that is your case, consider separating calls to `AutoUpdateUI` into separate [bridge module](#).

Database Explorer

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

This project provides access to objects defined in the Database Explorer. Documentation is available in the Javadoc.

Usage:

Nothing.

Interface Name	Stability Classification	Specified in What Document?
DatabaseExplorerAPI	Official	org-netbeans-modules-db/index.html
SQLSupportAPI	Official	org-netbeans-modules-db/org/netbeans/api/db/sql/support/package-summary.html

Diff

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The diff module provides the ability to visualize differences between source files. It also has a graphical conflicts resolver tool and built-in patch algorithm. List of the main features: Simple APIs, that provide access to registered diff and merge engines and visualizers. Built in and external diff engines defined. Graphical diff vizualizer and conflicts resolver. Extensible with additional diff and merge engines and vizualizers. Patch algorithm implemented for UNIX-style, unidiff and contextual diff formats.

Usage:

Nothing.

Interface Name	Stability Classification	Specified in What Document?
DiffAPI	Official	
DiffImplementationAPI	Friend	

Editor Code Templates

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Code Templates allow to paste various code snippets by using parametrized text. The parameters of the same name will share the same default value and if that value gets changed by user's typing the new value gets replicated into all the parameter's occurrences. Code Templates replace the original abbreviations functionality. Code template's example for (Iterator `${iterator}` = `${collection instanceof "java.util.Collection"}`).iterator(); `${iterator}`.hasNext();) { `${cursor}${iterator}`.next();" } Each parameter can have additional hints of what values can be assigned to it. The hint has a form `${param hint=value}` or just `${param hint}` which translates to `${param hint="true"}` If necessary the value of the hint can be enclosed in quotes to allow to write whitespace or { or } into the value. The quote can be written by using `\`. Reserved parameter names `${cursor}` defines position where the caret will be located after the editing of the code template default values will finish. Reserved hint names `${param editable=false}` can be used to make the parameter to be skipped from user's editing. This may be useful e.g. with using java-specific type hint (described below). Java: `${ind index}` defines that the default value of the parameter should be an unused variable in the given context named i. If i is already used then j is attempted or then k etc. until z. Then i0, i1 etc. are attempted. `${param type="java.util.Collection"}` defines java type that the parameter must be instance of. Besides class names there can be array e.g. `String[]` or generics `java.util.List<String>` `${param array}` defines parameter of array type (including arrays of primitive data types). `${param type="java.util.Iterator"}` defines that the parameter has the given java type. The template processing infrastructure will use short name `Iterator` and import `java.util.Iterator`.

Usage:

Nothing.



Editor Code Completion

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Code Completion provides users with a list of suggested completions for partially typed texts in the editor and various dialog input fields. The Code Completion module was created to replace the original legacy editor code completion which lacked several key requirements: Support for multiple independent code completion content providers. Implied requirement for ordering and prioritization of the completion items. Direct support for asynchronous completion result computation. Missing separation to the API and SPI and implementation parts.

Usage:

Nothing.



Editor Error Stripe

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The Error Stripe shows an overview of important information of an edited source code. It shows this information for the whole source code (regardless of its size).

Usage:

Nothing.

Interface Name	Stability Classification	Specified in What Document?
TextAPI	Stable	https://bits.netbeans.org/12.0/javadoc/org-openide-text/overview-summary.html OpenIDE Text API
ErrorStripeSPI	Stable	org-netbeans-modules-editor-errorstripe-api/index.html public SPI provided by the Error Stripe.
ErrorStripePrivateSPI	Private	Private mark provider SPI.

Editor Code Folding

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The Code Folding is part of the editor module functionality and it's responsible for hiding of the portions of the code that are less important for the user at the given time.

Usage:

Nothing.



Editor Guarded Sections

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Guarded Sections protects user from modifying document content. The main goal is to simplify work with such a content to module writers and preserve created sections.

Usage:

A module using the Guarded Sections API should also require a proper implementation. Eg in case of java content add to your module manifest file:

```
OpenIDE-Module-Requires: org.netbeans.api.editor.guards.Java
```

A module implementing the Guarded Sections SPI should provide a token in the manifest file. Eg in case of java content add:

```
OpenIDE-Module-Provides: org.netbeans.api.editor.guards.Java
```

Editor Library 2

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The Editor Library 2 module is a set of official APIs and SPIs, designed to replaces the original Editor Library with legacy APIs that are not properly structured and do not conform to the rules implied on the current NB APIs. The APIs currently offered in Editor Library 2 module include: **editor-code-generator editor-highlighting editor-typing-hooks editor-caret**

Usage:

Just normal module dependency.

MIME Lookup API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Each editor provides an EditorKit which controls the policy of specific MIME content type. The policy of content type should be easily registered and found via some lookup mechanism, that will provide convenient way of using it either for kit provider or base editor infrastructure. In addition to this, the policy can be inherited, (e.g. in case of embedded kits like JSP) and the content types need to be merged in this case. MIME Lookup API should provide all mentioned requierements via easy lookup query, so content type policy user need not to solve this searching and merging on its own side.

Usage:

Nothing.

Interface Name	Stability Classificat	Specified in What Document?
MimeLookupAPI	Official	
MimeLookupSPI	Official	

Editor Settings

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

The legacy settings system in the editor module is complicated, error prone and hard to use. It'd been created spontaneously over the years to support immediate needs at that time without paying enough attention to extensibility and interoperability. Historically any module providing editor settings needed to depend on the whole editor module. The main purpose of this project is to define API for editor settings, that is lightweight and easily extensible. The API relies on MimeLookup to provide a way of registering and looking up settings. The aim is NOT to provide an implementation of a storage for editor settings, but to define an interface between this storage and clients like <mime-type> editors, external-editor, etc.

Interface Name	Stability Classificat ion	Specified in What Document?
EditorSettingsAPI	Official	API

External Execution API

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Provides common APIs to execute external process in the IDE to handle its streams and present the output to the user. Input/line processing can be used as separate part.

Usage:

Nothing.

Interface Name	Stability Classification	Specified in What Document?
ExternalExecutionAPI	Stable	org-netbeans-modules-extexecution/org-netbeans/api/extexecution/package-summary.html
ExternalExecutionInputAPI	Stable	org-netbeans-modules-extexecution/org-netbeans/api/extexecution/input/package-summary.html
ExternalExecutionInputPrintingAPI	Stable	org-netbeans-modules-extexecution/org-netbeans/api/extexecution/print/package-summary.html
ExternalExecutionSPI	Stable	org-netbeans-modules-extexecution/org-netbeans/spi/extexecution/package-summary.html
ExternalExecutionDestroySPI	Stable	org-netbeans-modules-extexecution/org-netbeans/spi/extexecution/destroy/package-summary.html
ExternalExecutionOpenSPI	Stable	org-netbeans-modules-extexecution/org-netbeans/spi/extexecution/open/package-summary.html
ExternalExecutionStartupAPI	Stable	org-netbeans-modules-extexecution/org-netbeans/api/extexecution/startup/package-summary.html
ExternalExecutionStartupSPI	Stable	org-netbeans-modules-extexecution/org-netbeans/spi/extexecution/startup/package-summary.html

Java Platform

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Many Java-based project types need to be able to configure the version and location of Java to be used when building and running the project. This API/SPI permits these platforms to be registered and queried, and any customizations made in an appropriate GUI and persisted to disk.

Java Platform UI

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Many Java-based project types need to be able to configure the version and location of Java to be used when building and running the project. This API/SPI permits these platforms to be registered and queried, and any customizations made in an appropriate GUI and persisted to disk.

Usage:

Nothing.

Interface Name	Stability Classification	Specified in What Document?
JavaPlatformUI	Official	<p>The Java Platform API permits access to installed Java platforms (for example, the J2SE JDK, or various mobile-device emulators for J2ME). Particular platform types are registered by modules and can store customized information about the platform to disk.</p> <p>Note that this module concentrates the UI part of API. Its desktop-independent counterpart is in the org.netbeans.modules.java.platform module.</p>

Java Project Support

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Provides support infrastructure for projects working with the Java language.

Java Project Support UI

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

Provides support infrastructure for projects working with the Java language.



Java Source UI

[javadoc](#) | [download](#) | [architecture](#) | [usecases](#)

JAVA. SOURCE UI

Usage:

Interface Name	Stability Classification	Specified in What Document?
java.sourceui	Official	APIs for controlling visual aspects of processing of Java files.

JavaHelp Integration

[javadoc](#) | [download](#) | [architecture](#)

The JavaHelp integration API wraps the standard JavaHelp extension library. It also provides a small additional API for NetBeans modules to supply help sets to the system, add GUI menu items, and request that particular help topics be displayed. **JavaHelpIntegrationAPI**

6 : OUTPUT SCREENS

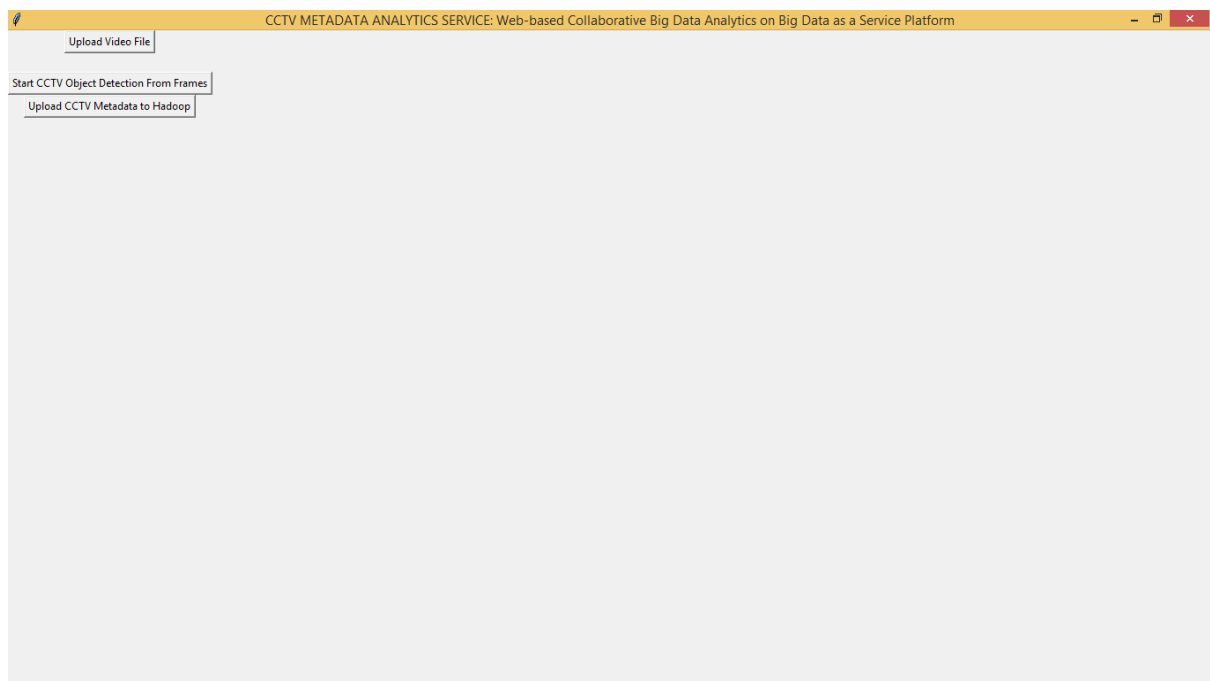
Web-based Collaborative Big Data Analytics on Big Data as a Service Platform

Now-a-days large amount of data is gather on internet and to process such data traditional technique is not suitable and that big data can be process using distributed technology such as HADOOP. In this paper author is describing concept to implement above technique using ROLE based mechanism for web application and by using CCTV real time object detection from videos and those detection will be obtained from video frames and large amount of data will be collected and that data can be process and store at Hadoop.

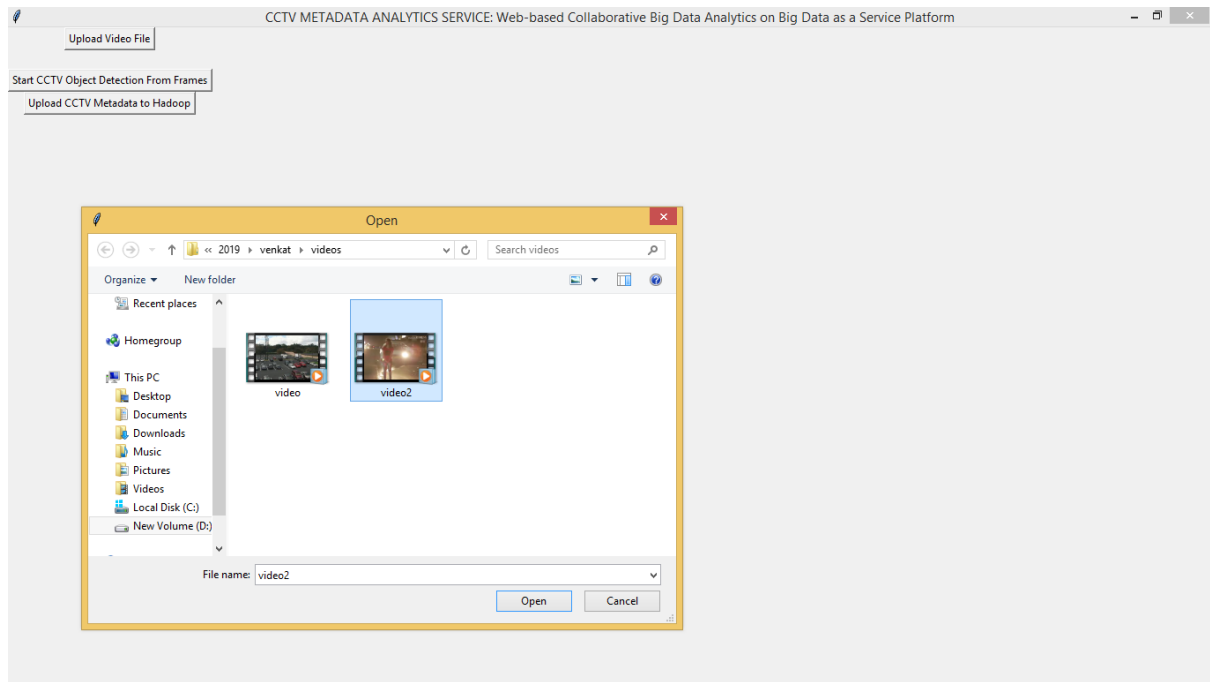
Generated data from video frames can be saved inside xml file. This project implemented using python

Screen shots

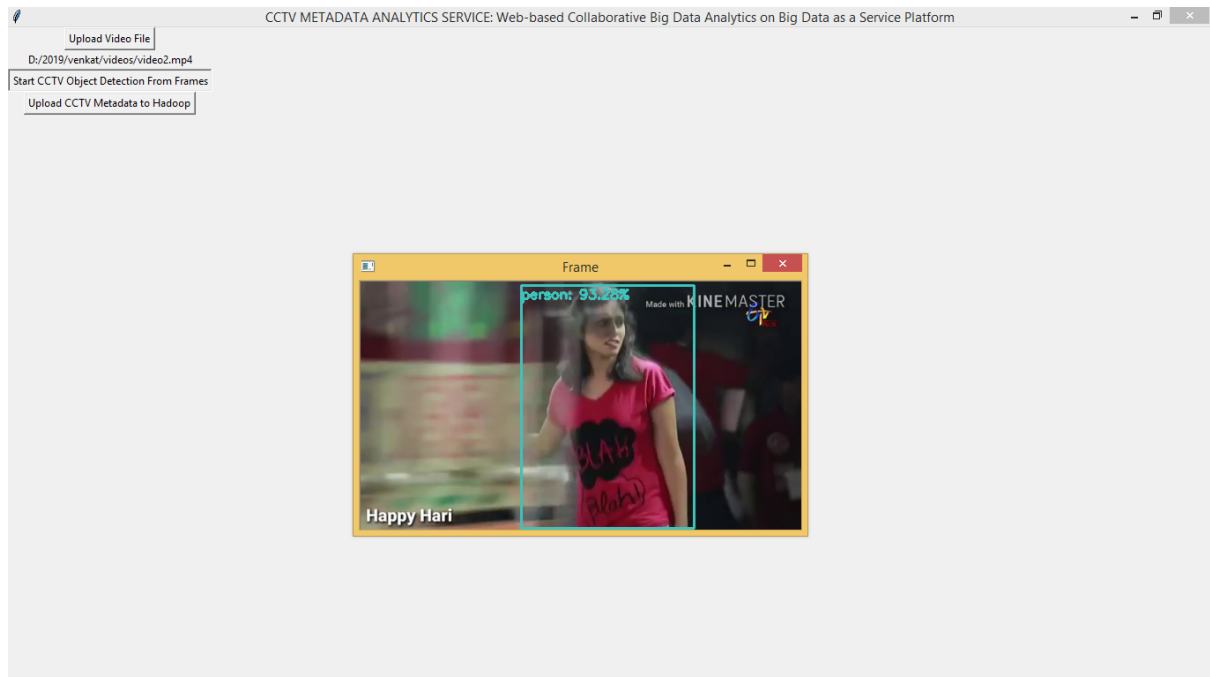
Double click on run.bat file to get below screen



Click on 'Upload Video File' button to upload video



Now click on 'Start CCTV Object Detection From Frames' button to start detecting objects from video

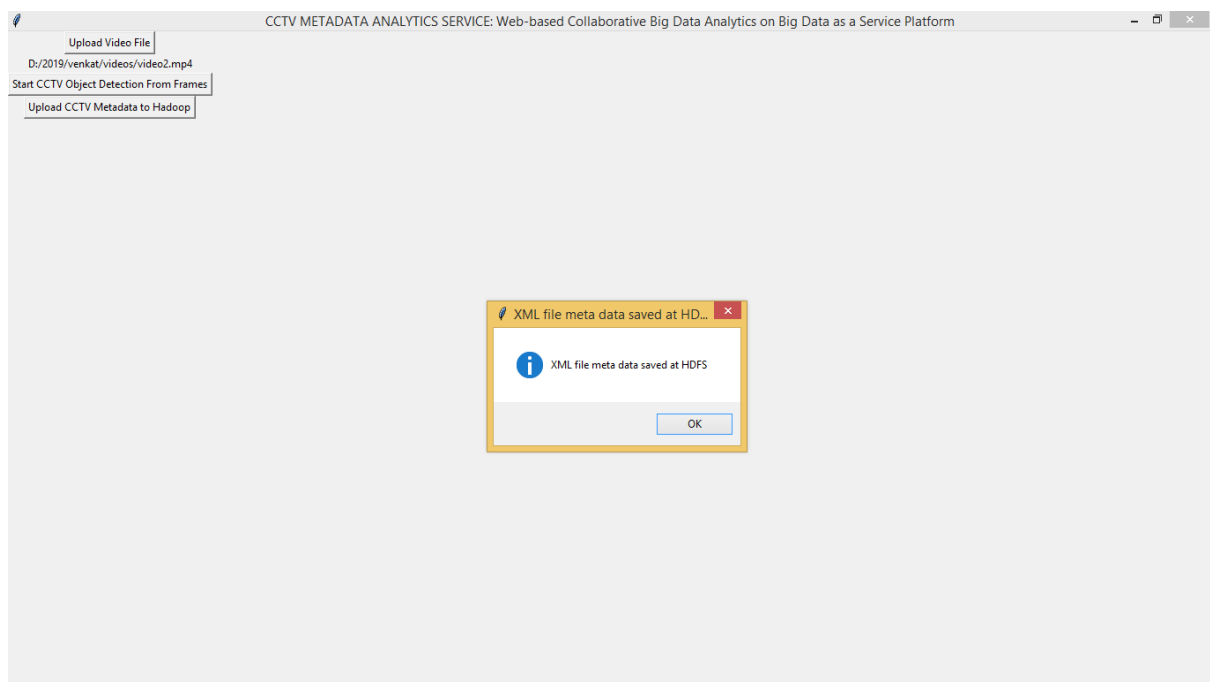


In above screen we can see while video playing application will start detecting objects and all those objects details we can see in black console also

```
C:\Windows\system32\cmd.exe

person: 96.92% 117 frames
person: 91.62% 117 frames
person: 94.77% 118 frames
person: 93.90% 118 frames
person: 96.19% 119 frames
person: 94.93% 119 frames
person: 94.70% 120 frames
person: 92.50% 120 frames
person: 95.31% 121 frames
person: 92.42% 121 frames
person: 95.61% 122 frames
person: 93.48% 122 frames
chair: 28.14% 123 frames
person: 91.80% 123 frames
person: 87.21% 123 frames
person: 97.00% 124 frames
person: 96.88% 125 frames
person: 96.87% 126 frames
person: 96.79% 127 frames
person: 89.98% 128 frames
person: 64.47% 128 frames
person: 95.18% 129 frames
person: 95.09% 130 frames
person: 94.76% 131 frames
person: 36.43% 131 frames
```

Now click on ‘Upload CCTV Metadata to Hadoop’ button to save metadata of video to be saved on Hadoop



Now we can see this data in hadoop with file name as date and time and format of file is in xml

Need this project in Hadoop - k... x Browsing HDFS x How to remove all special chara... x python - Remove all special char... x +

localhost:50070/explorer.html#/user/user

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

/user/user Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
-rwxr-xr-x	user	supergroup	3.01 KB	1	128 MB	20190521_164031172377_xml.txt
-rwxr-xr-x	user	supergroup	20.97 KB	1	128 MB	20190521_165125682334_xml.txt
-rwxr-xr-x	user	supergroup	3.28 KB	1	128 MB	SearchObject.py
-rwxr-xr-x	user	supergroup	1.75 KB	1	128 MB	xml.txt

Hadoop, 2014.

SearchObject.py Show all

From hadoop we can download and see that file data

Need this project in Hadoop - k... x Browsing HDFS x How to remove all special chara... x python - Remove all special char... x +

localhost:50070/explorer.html#/user/user

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

Browse Directory

/user/user Go!

Permission	Owner	Group	Size	Replication	Block Size	Name
-rwxr-xr-x	user	supergroup	3.01 KB	1	128 MB	20190521_164031172377_xml.txt
-rwxr-xr-x	user	supergroup	20.97 KB	1	128 MB	20190521_165125682334_xml.txt
-rwxr-xr-x	user	supergroup	3.28 KB	1	128 MB	SearchObject.py
-rwxr-xr-x	user	supergroup	1.75 KB	1	128 MB	xml.txt

Hadoop, 2014.

SearchObject.py Show all

File information - 20190521_165125682334_xml.txt

[Download](#)

Block information -- Block 0

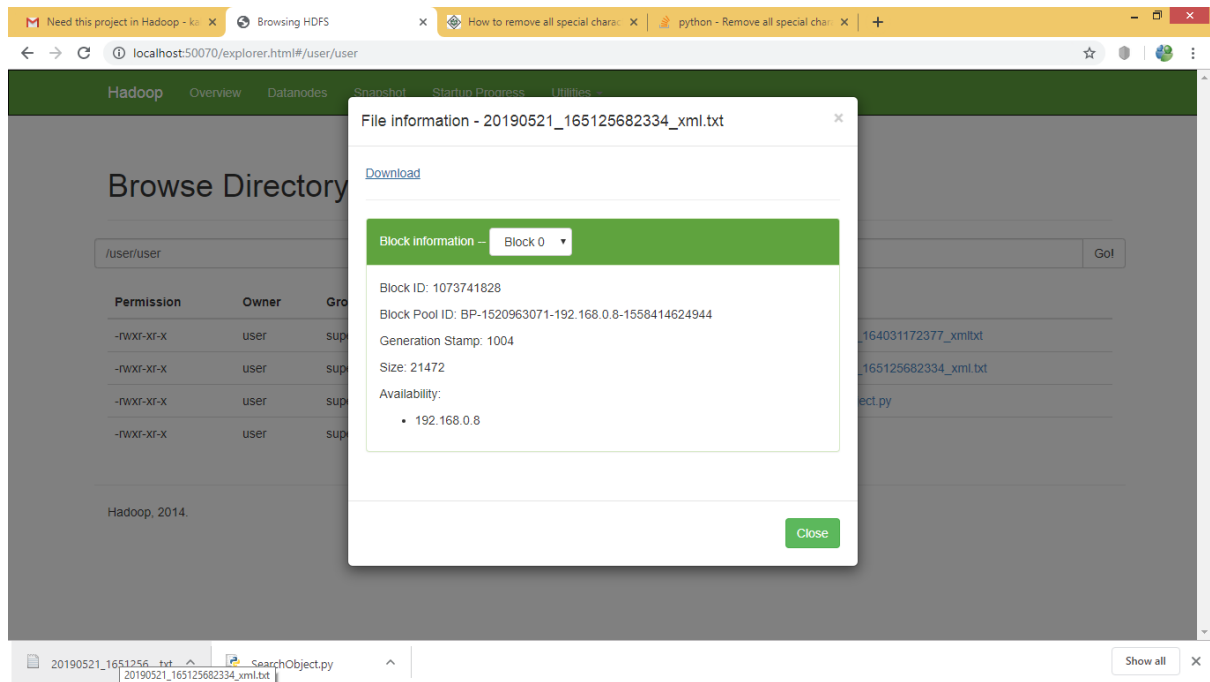
Block ID: 1073741828
 Block Pool ID: BP-1520963071-192.168.0.8-1558414624944
 Generation Stamp: 1004
 Size: 21472
 Availability:
 • 192.168.0.8

Close

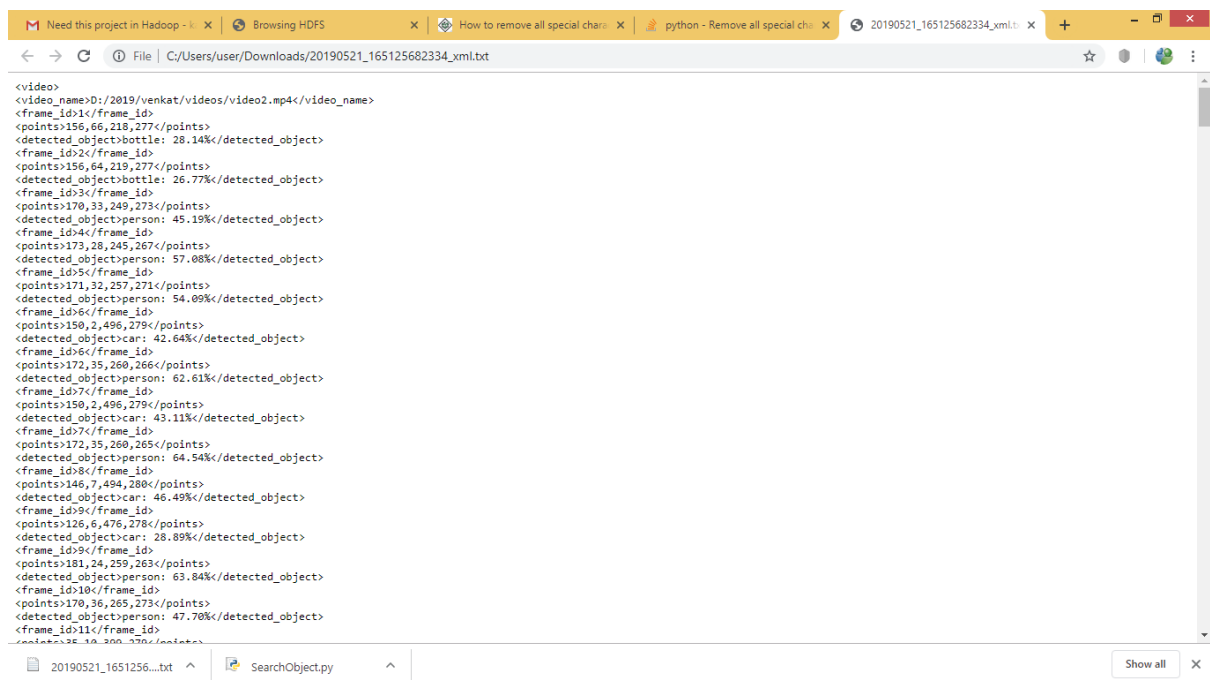
localhost:50070/webhdfs/v1/user/user/20190521_165125682334_xml.txt?op=OPEN

SearchObject.py Show all

In above screen click on download link to download data



In browser status bar we can see data is downloaded, now we can open and see that video data



In above screen we can see xml data displaying video name, points of frame, frame id and detected objects

7 : TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS :

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasising pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must

be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

6.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8 : CONCLUSION

Geosocial Networks can be an asset for governments in terms of providing facilities and safety from disasters through proper management and reduction of the fear of the spread of any infections. Similarly, such networks can benefit to common citizens by providing recommended systems, transport safety, healthcare, etc., and to entrepreneurs for launching new products in various areas by monitoring the geosocial data of a particular area. However, such benefits can only be derived with better analytics that employs a significant amount of data generated from various Geosocial Networks. This is possible with advanced technology and better analytics, and a system with high computing capabilities. Therefore, in this paper, we proposed a system that uses geosocial data for better planning, safety from disasters, and proper management, awareness, etc., based on various geolocations. The system not only can harvest a large amount of data at high-speed from Geosocial Networks, but it can also process, analyze, and make decisions in real time. We analyzed Twitter data for various events using the proposed system. The system was developed using a Hadoop ecosystem with Spark. The system was more efficient when processing a lot of datasets, and showed the advantage of increased throughput with an increase in data volume

REFERENCES

- [1] <http://ajast.net/data/uploads/4ajast-9.pdf>
- [2] <http://ijesc.org/upload/15de67d580745fa9233dd9906e322d67.QR%20Code%20Security%20and%20Solution.pdf>
- [3] <http://academicscience.co.in/admin/resources/project/paper/f201405051399309076.pdf>
- [4] <https://searchsecurity.techtarget.com/definition/one-time-password-OTP>
- [5] https://connect.cognex.com/India-Cognex-Industrial-Barcode-Readers-LP?src=0ebcb667-3333e911-9137-00505693004d&cm_campid=0ebcb667-3333-e911-913700505693004d&gclid=CjwKCAjwkPX0BRBKEiwA7THxiL82xcb7QTpjhbnWReptsAWy_uGGwYQZ5XWEvtIipgKVdKuLHN-ihoCQ84QAvD_BwE
- [6] https://en.wikipedia.org/wiki/One-time_password [7] <https://en.wikipedia.org/wiki/Barcode>
- [8] <https://ieeexplore.ieee.org/document/5711134>