A PROJECT REPORT ON

ENHANCING PUBLIC SERVICES THROUGH AI AUTOMATION

Major project submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

(2020-2024)

BY

A. LOUKYA SREE	**20241A1201**
U. SUHITHA	**20241A1252**
V. HARIKA	**20241A1255**
B. SATHVIKA	**21245A1204**

Under the Esteemed guidance

of

N. Mounika

Assistant Professor

Dept of IT

DEPARTMENT OF INFORMATION TECHNOLOGY

GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS) HYDERABAD.

CERTIFICATE

This is to certify that it is a bona fide record of Major Project work entitled "**ENHANCING PUBLIC SERVICES THROUGH AI AUTOMATION**" done by **A.LOUKYA SREE (20241A1201), U.SUHITHA (20241A1252), V.HARIKA REDDY (20241A1255), B.SATHVIKA (21245A204))** of **B.Tech (IT)** in the Department of Information Technology, **Gokaraju Rangaraju Institute of Engineering and Technology** during the period2020-2024 in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from GRIET, Hyderabad.

N. Mounika,
Assistant Professor
(Internal guide)

Dr. Y. J. Nagendra Kumar,
(Head of the Department)

(Project External)

ACKNOWLEDGEMENT

We take immense pleasure in expressing gratitude to our internal guide, **N. Mounika, Assistant Professor, Dept. of IT, GRIET.** We express our sincere thanks for her encouragement, suggestions, and support, which provided the impetus and paved the way for the successful completion of the project work.

We wish to express our gratitude to **Dr.Y. Jeevan Nagendra Kumar,** HOD IT**,** and our Project Coordinators, **Dr.Y. Jeevan Nagendra Kumar, G. Vijendar Reddy** for their constant support during the project.

We sincerely thank **Dr.Jandhyala N Murthy**, Director, GRIET, and **Dr. J. Praveen**, Principal, GRIET, for providing us a conductive environment for easily carrying through our academic schedules and project.

We also take this opportunity to convey our sincere thanks to the teaching and non- teaching staff of GRIET College, Hyderabad.

Name: Akula Loukya Sree
Email: loukyaakula7@gmail.com
Contact No: 9052920038
Address: Sangareddy, Telangana.

Name: Suhitha Ulavalapudi
Email: suhitha.rinky@gmail.com
Contact No: 9966961403
Address: Malkajgiri, Hyderabad.

Name: Vanguru Harika Reddy
Email: harika282003@gamil.com
Contact No: 7396401503
Address: Jeedimetla, Hyderabad

Name: Bathula Sathvika
Email: sathvikabathula0@gmail.com
Contact No: 8885755404
Address: Lingampally, Telangana.

DECLARATION

This is to certify that the project entitled "**ENHANCING PUBLIC SERVICES THROUGH AI AUTOMATION**" is a bonafide work done by us in partial fulfilment of the requirements for the award of the degree **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites, books and paper publications are mentioned in the Bibliography.

This work was not submitted earlier at any other University or Institute for the award of any degree.

A. LOUKYA SREE	**20241A1201**
U. SUHITHA	**20241A1252**
V. HARIKA	**20241A1255**
B. SATHVIKA	**21245A1204**

TABLE OF CONTENTS

ABSTRACT

- Artificial Intelligence (AI) has been around for some decades in several theoretical forms and complicated systems. However, only recent advances in computational powers and big data have enabled AI to achieve outstanding results in an ever-growing number of domains.
- It still faces several challenges that hinder its deployment in the e-government applications-both for improving the e-government systems and the e-government-citizens interactions. This project explores the integration of Artificial Intelligence (AI) in e-government systems, aiming to automate and optimize various governmental processes.
- It addresses the challenges of e-government systems and propose a framework that utilizes AI technologies to automate and facilitate e-government services. It outlines a framework for the management of
- e-government information resources. It contains a set of deep learning models that aims to automate several e-government services.
- This system consists of a smart e-government platform architecture that supports the development and implementation of AI applications of e-government. The utilization of AI in e-government also promises to improve data security and privacy measures, addressing concerns related to digital governance.
- This project's significance lies in its potential to revolutionize public administration making government services more efficient, transparent, and user-centric.

Keywords: Artificial Intelligence, Big Data, e-Government, Public Services, AI Automation.

1. INTRODUCTION

Artificial Intelligence (AI) has been around for some decades in several theoretical forms and complicated systems; however, only recent advances in computational powers and big data have enabled AI to achieve outstanding results in an ever-growing number of domains. AI can be defined as the ability of a computer to imitate the intelligence of human behavior while improving its own performance. AI is not only robotics, rather an intelligent behavior of an autonomous machine that describes the brain of the machine and not its body; it can drive a car, play a game, and perform diverse sophisticated jobs. AI is a field that falls at the intersections of several other domains, including Machine Learning, Deep Learning, Natural Languages Processing, Context Awareness, and Data Security and Privacy.

Machine Learning (ML) is the ability of an algorithm to learn from prior data in order to produce a smart behavior and make correct decisions in various situations that it has never faced before. ML algorithms are enabled by training computational model, which is the process of exposing an algorithm to a large dataset (e.g., citizens' demographics) in order to predict future behaviors (e.g., employment rates). The process of learning from prior datasets is known as a supervised learning. Unlike traditional ML algorithms, Deep Learning, a subfield of ML, has emerged to outcome the limitations of prior ML algorithms. Deep learning can be defined as a mapping function that maps raw input data (e.g., a medical image) to the desired output (e.g., diagnosis) by minimizing a loss function using some optimization approach, such as stochastic gradient descent (SGD). Deep learning algorithms, inspired by the neural networks in the human brain, are built with a large number of hierarchical artificial neural networks that map the raw input data (inserted at the input layer) to the desired output (produced at the output layer) through a large number of layers (known as hidden layers), and thus the name deep learning.

Despite the fact that deep learning has improved the state-of-art results in several domains, it is still evident that e-government applications face several challenges regarding adapting deep learning. First, given the recent and rapid advances in the deep learning domain, it is becoming more difficult to and experts of this technology who are capable of developing efficient and reliable AI applications, especially in third world countries. Second, the development life cycle of AI

projects, especially deep learning, has introduced a new set of development challenges. In particular, traditional software development focuses on meeting a set of required functional and non-functional requirements; in contract, deep learning development focuses on optimizing a specific metric based on a large set of parameters, which is done in a unsystematic search approach. Third, integrating AI and deep learning applications in e-government services requires strong policies and measures on data security and privacy.

1.1 Objective:

The overarching objective of the project is to harness the transformative potential of artificial intelligence (AI) to enhance e-government systems. This involves addressing longstanding challenges in deploying AI within the context of public administration while leveraging recent advancements in computational power and big data.

1. **Explore Integration of AI in E-Government Systems:** Investigate how AI technologies can be effectively integrated into e-government frameworks to automate and optimize various governmental processes. This includes improving interactions between government systems and citizens.

2. **Address Challenges in E-Government:** Identify and tackle obstacles that have historically hindered the deployment of AI in e-government applications. This encompasses challenges related to system complexity, data management, and citizen engagement.

3. **Propose a Framework for AI-Driven E-Government:** Develop a comprehensive framework that leverages AI technologies to automate and facilitate e-government services. This framework should address the specific needs and challenges of e-government systems, providing a roadmap for implementation.

4. **Design a Smart E-Government Platform:** Develop an architecture for a smart e-government platform that supports the development and implementation of AI applications. This platform should provide a flexible and scalable infrastructure for deploying AI-driven solutions across various government services.

5. **Improve Data Security and Privacy:** Explore how the integration of AI can enhance data security and privacy measures within e-government systems. This includes developing mechanisms for secure data handling and addressing concerns related to digital governance.

6. **Revolutionize Public Administration:** Ultimately, the project aims to revolutionize public administration by making government services more efficient, transparent, and user-centric. By harnessing the power of AI, the project seeks to empower governments to better serve their citizens and adapt to the evolving digital landscape.

2. REQUIREMENT ENGINEERING

User Interface:

The user interface of this system is a user-friendly python Graphical User Interface.

Hardware Interfaces:

The interaction between the user and the console is achieved through python capabilities.

Software Interfaces:

The required software is python.

Operating Environment:

Windows XP.

HARDWARE REQUIREMENTS:

System	:	Pentium IV 2.4 GHz.
Hard Disk	:	40 GB.
Floppy Drive	:	1.44 Mb.
Monitor	:	14’ Color Monitor.
Mouse	:	Optical Mouse.
Ram	:	512 Mb.

SOFTWARE REQUIREMENTS:

Operating system	:	Windows 7 Ultimate.
Coding Language	:	Python.
Front-End	:	Python.
Designing	:	Html,css,javascript.
Data Base	:	MySQL.

3. LITERATURE SURVEY

[1] This research investigated the issues associated with using Internet of Things and Artificial Intelligence for smart governance in the public sector. The roadmap is developed based on the suggested framework using Internet of Things and Artificial Intelligence for smart governance. This study includes some significant research areas for the future, namely, undertaking domain-specific studies, extending beyond adoption studies to explore the implementation and assessment of these systems, and concentrating on particular problems that prevent effective Internet of Things and Artificial Intelligence system installation.

[2] This paper develops a research agenda for GovTech based on a conceptual framework. This framework reveals four interrelated design areas for GovTech: institutional, governance, technical and human-centered design. Governments can employ the conceptual framework to further align and develop their strategies by focusing on GovTech governance, referring to the ability to manage the various interdependencies between the four design areas.

[3] This is a large-scale randomized controlled experiment, conducted in a real-world setting with actual taxpayers, to estimate the average causal effect of a concrete policy instrument on the rate of e-filing adoption by slower adopters. The largest estimated effect more than doubles the rate of use of the service, relative to the (non-treated) control group. This finding illustrates the potency of nudges as a practical tool to address the underutilization of important digital public services.

[4] The strategic approach to digitalization as a means to address this dilemma. It underscores the importance of timely awareness, motivation, and the formulation of new strategic goals, alongside goal-directed policy actions. Ultimately, the abstract suggests that these governments offer insights into navigating the challenges of social acceleration while preserving democratic principles.

[5] Through a mixed-method design involving participants from various backgrounds, the research identifies distinct user groups with diverse expectations and requirements. This leads to the proposal of a novel typology of technological frames for online community policing, offering valuable insights for design considerations. Furthermore, the study contributes to the broader

understanding of technological frames by showcasing a sequential mixed-method approach applicable beyond specific context.

[6] By proposing a framework that integrates AI technologies, it aims to automate and enhance various e-government services. The outlined framework emphasizes the management of information resources and the development of deep learning models for service automation. A smart e-government platform architecture to support the implementation of AI applications. Ultimately, the overarching goal is to leverage trustworthy AI techniques to streamline processes, lower costs, and enhance citizen satisfaction within e-government services.

[7] It delineates the prevalent use of AI-analytical models and automation systems at the organizational level, while highlighting the deployment of recommender systems and chatbots in citizen services. The findings underscore the tangible benefits of AI for governments, including cost reduction and enhanced decision-making, alongside the value of personalized experiences for citizens. However, the study also identifies significant challenges, ranging from organizational obstacles to citizen-related dilemmas such as ambiguity and privacy concerns. As a result, the paper advocates for further research to address these challenges and maximize the potential of AI in government and public sector contexts.

4. SYSTEM ANALYSIS

4.1 Existing System:

Recently, many countries have adopted e-government services in various departments and many autonomous applications. While there are several studies conducted for enhancing e-government services, only a few of them address utilizing recent advances in AI and deep learning in the automation of e-government services. Therefore, there is still an urgent need to utilize state-of-the-art AI techniques and algorithms to address e-government challenges and needs. In contrast, implementing e-government applications still faces several challenges, including the following:

Trust: trusting online services depends heavily on a couple of factors including, the citizens trust in the government itself, the quality of the online services, and the personal believes (e.g., there still a large number of citizens who prefer to handle paper applications rather than web services).

Lack of experts: implementing high-quality online services requires the establishment of the right team of experts that covers all involved practice areas from web development to security and privacy. Inaccessibility: several third world countries still face significant issues on accessing the internet and its services.

Security: state-of-the-art security measures are required to secure e-government applications and the citizen's privacy.

4.2 Proposed System:

In this paper author describing concept to automate government services with Artificial Intelligence technology such as Deep Learning algorithm called Convolution Neural Networks (CNN). Government can introduce new schemes on internet and peoples can read news and notifications of such schemes and then peoples can write opinion about such schemes and these opinions can help government in taking better decisions. To detect public opinions about schemes automatically we need to have software like human brains which can easily understand the opinion which peoples are writing is in favor of positive or negative. To build such automated opinion detection author is suggesting to build CNN model which can work like human brains. This CNN model can be generated for any services and we can make it to work like automated decision making without any human interactions. To suggest this technique author already describing concept to implement multiple models in which one model can detect or recognize human hand written digits and second model can detect sentiment from text sentences which can be given by

human about government schemes. In our extension model we added another model which can detect sentiment from person face image. Person face expressions can describe sentiments better than words or sentences. So, our extension work can predict sentiments from person face images.

Modules:

USER:

Generate Hand Written Digits Recognition Deep Learning Model: using this model we are building CNN based hand written model which take digit image as input and then predict the name of digit. CNN model can be generated by taking two types of images called train (train images contain all possible shapes of digits human can write in all possible ways) and test (Using test images train model will be tested whether its giving better prediction accuracy). Using all train images CNN will build the training model. While building model we will extract features from train images and then build a model. While testing also we will extract features from test image and then apply train model on that test image to classify it. Generate Text & Image Based Sentiment Detection Deep Learning Model: using this module we will generate text and image-based sentiment detection model. All possible positive and negative words will be used to generate text-based sentiment model. All different types of facial expression images will be used to generate image-based sentiment model. Whenever we input text or image then train model will be applied on that input to predict its sentiments.

Upload Test Image & Recognize Digit: By using this module we will upload text image and apply train model to recognize digit.

Write Your Opinion About Government Policies: using this module we will accept user's opinion and then save that opinion inside application to detect sentiment from opinion.

View Peoples Sentiments From Opinions: using this module user can see all users opinion and their sentiments detected through CNN model.

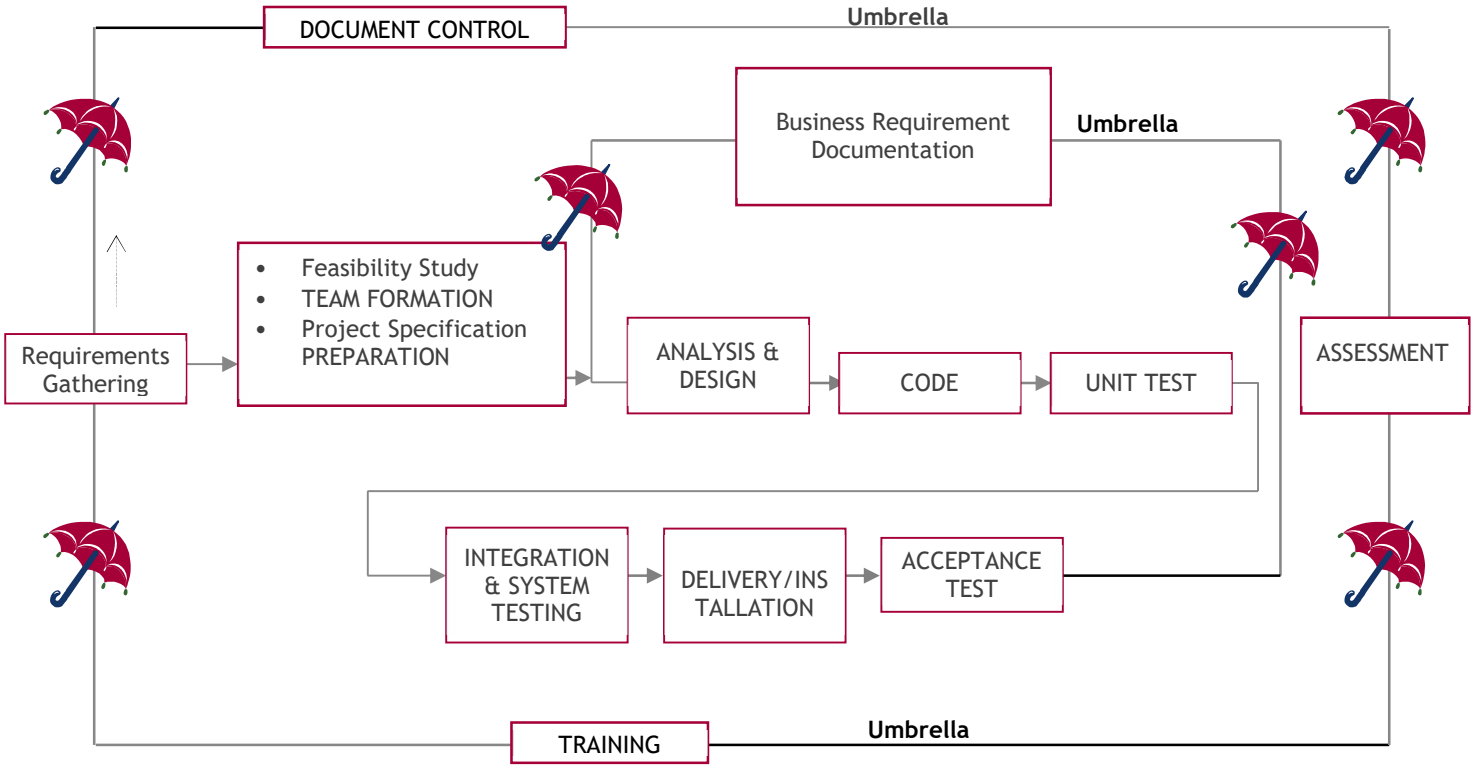
Upload Your Face Expression Photo About Government Policies: using this module user will upload his image with facial expression which indicates whether user is satisfy with this scheme

or not.

Detect Sentiments From Face Expression Photo: using this module different users can see the facial expression image and detected sentiment which is uploaded by past users.

4.3. PROCESS MODEL USED WITH JUSTIFICATION:

SDLC (Umbrella Model):



SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

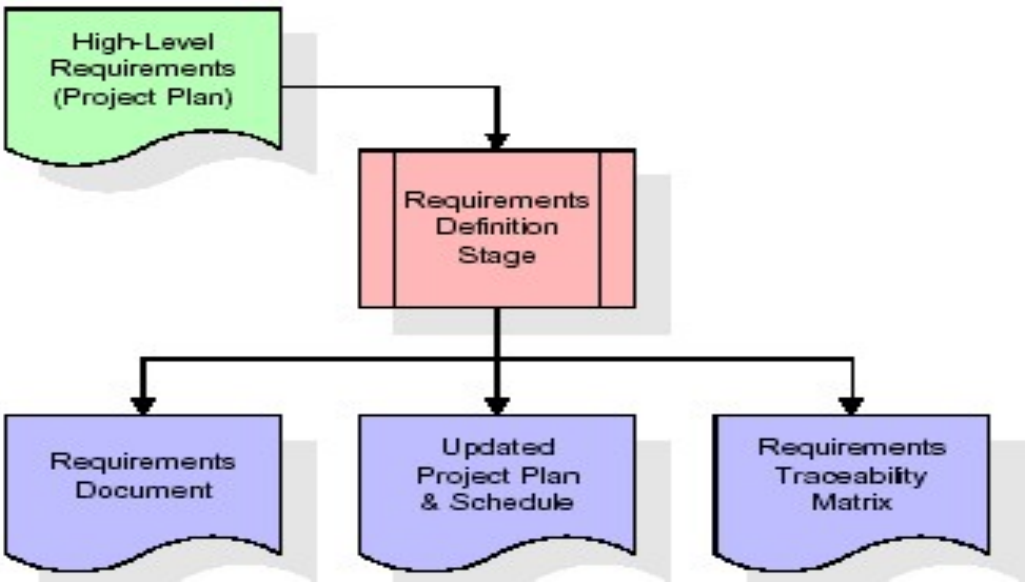
Stages in SDLC:

- Requirement Gathering
- Analysis
- Designing
- Coding

- Testing
- Maintenance

Requirements Gathering stage:

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components

developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

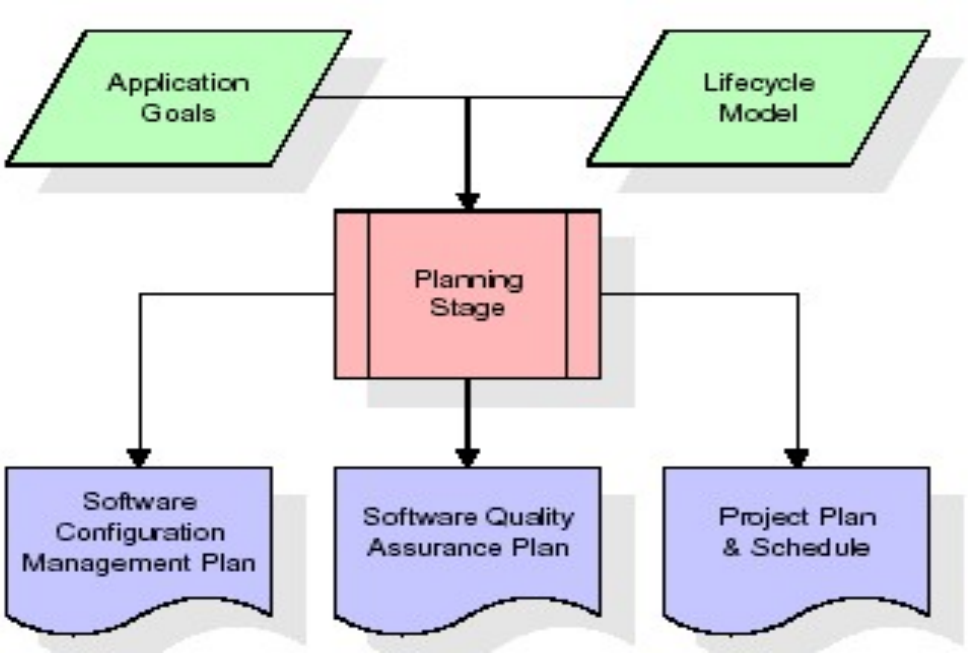
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

- Feasibility study is all about identification of problems in a project.
- No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
- Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

Analysis Stage:

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

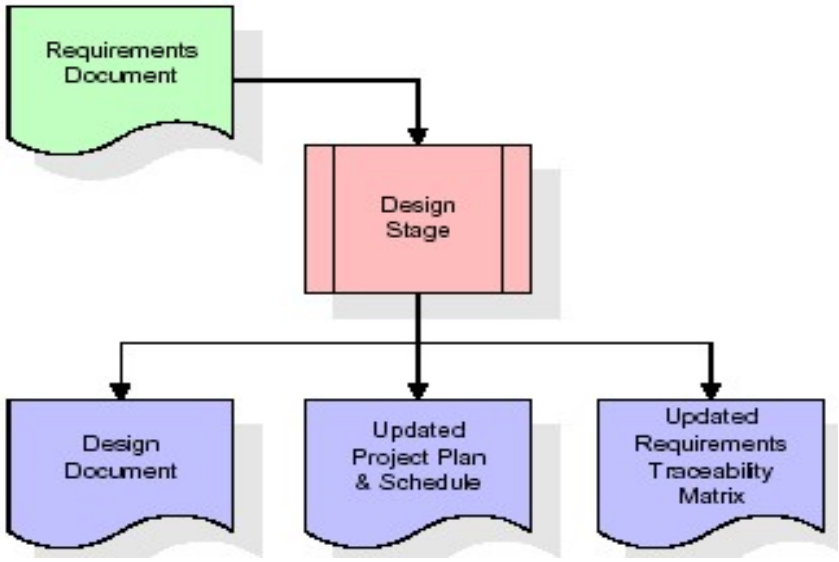


The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high-level estimates of effort for the out stages.

Designing Stage:

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal

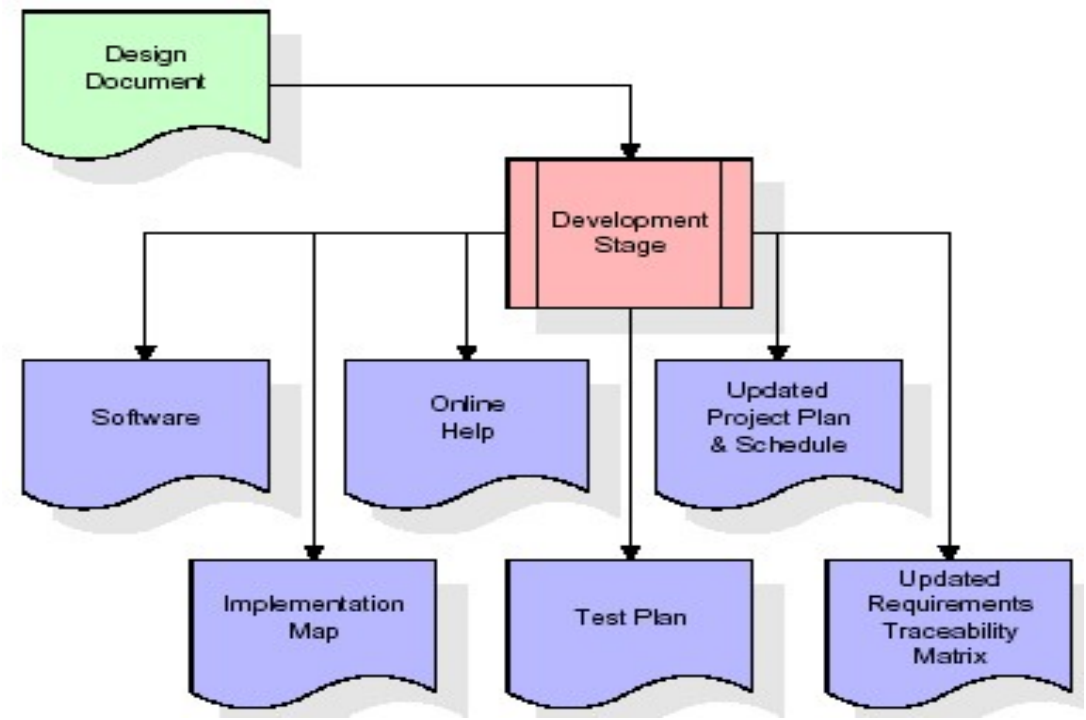
additional input.



When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development (Coding) Stage:

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

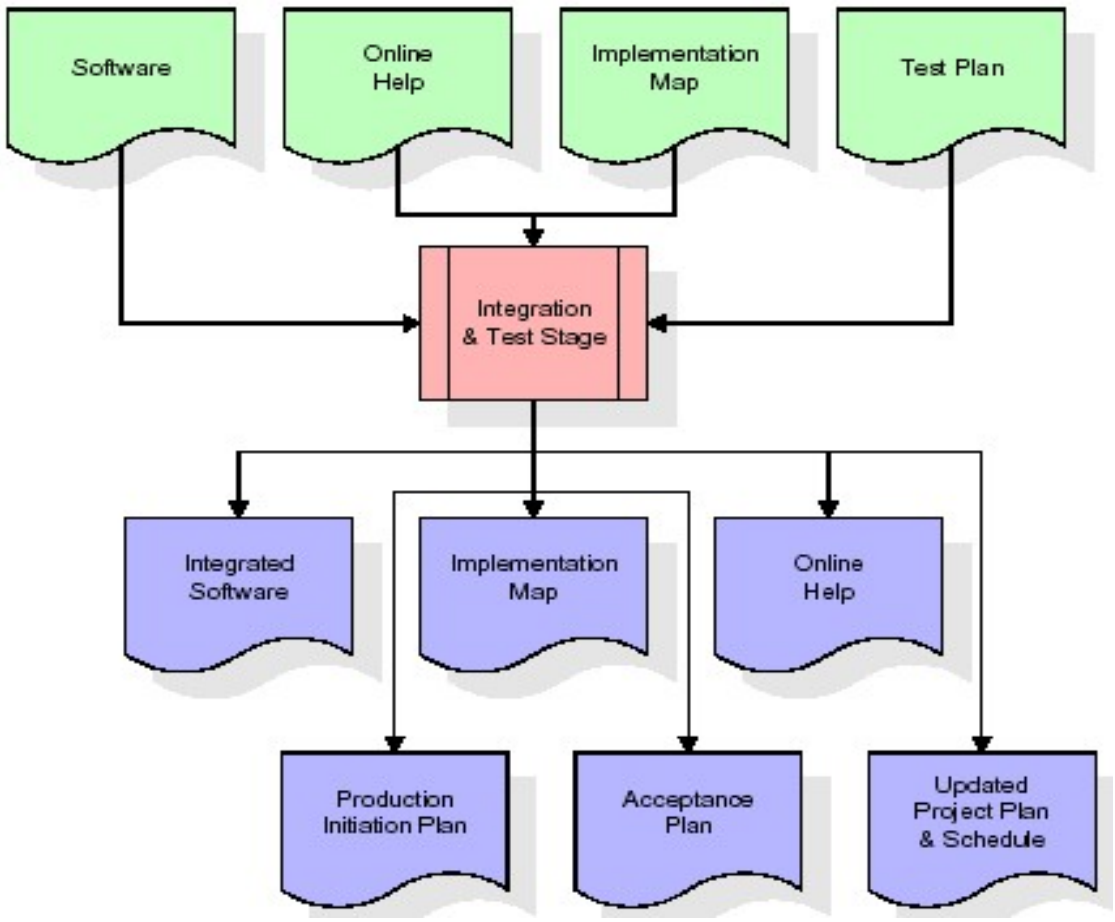


The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

Integration & Test Stage:

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test

suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

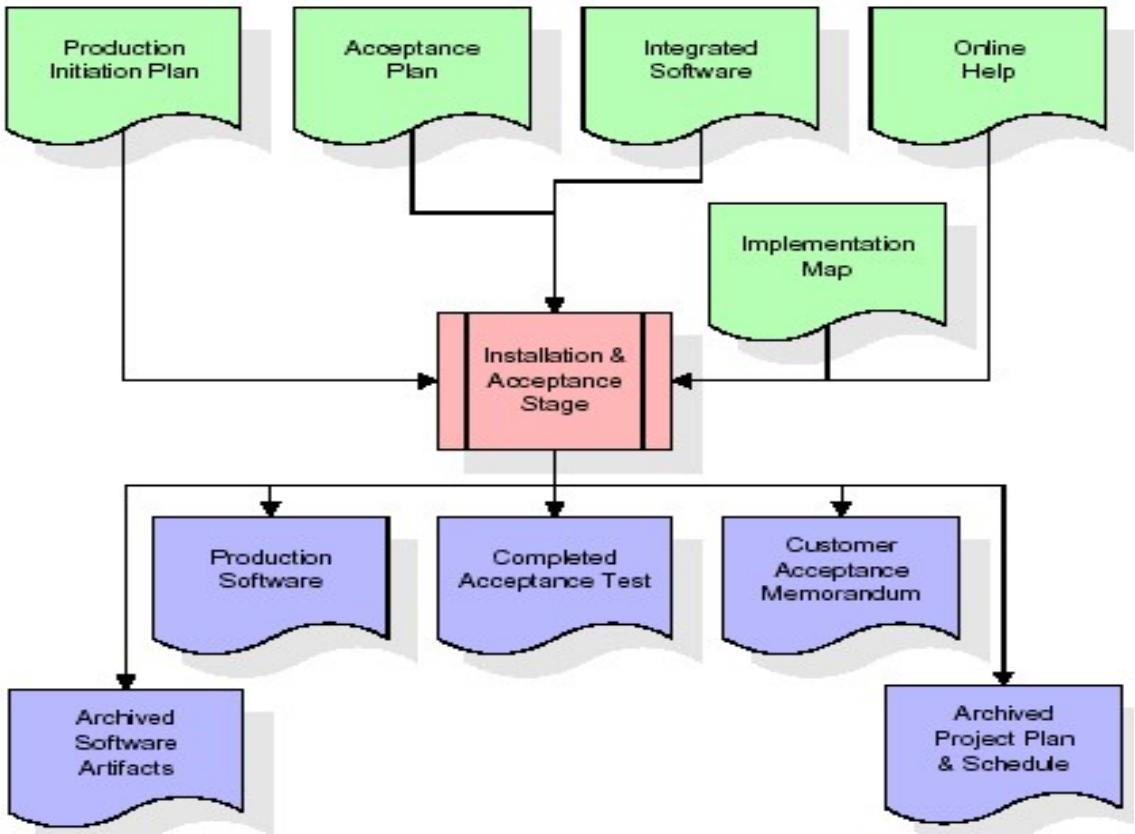


The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

Installation & Acceptance Test:

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future

reference.

Maintenance:

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

4.4. Software Requirement Specification

4.4.1. Overall Description:

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development lifecycle domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- Business requirements describe in business terms *what* must be delivered or accomplished to provide value.
- Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
- Process requirements describe activities performed by the developing organization. For instance, process requirements could specify. Preliminary investigation examine project

feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, there is nominal expenditure and economic feasibility for certain.

- **OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

- **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

5. DESIGN REQUIREMENT ENGINEERING

5.1 UML DIAGRAMS

The Unified Modeling Language (UML) is an extensively adopted and standardized framework utilized for the specification, visualization, construction, and documentation of various elements within software systems. It serves as a specialized diagramming language, diverging from conventional programming languages like C++, Java, and COBOL. UML is meticulously designed to provide a comprehensive set of visual tools, enabling the creation of detailed blueprints or models that depict the intricacies of software applications.

Unlike programming languages that delve into coding syntax, UML places a primary emphasis on the visual representation of a software system's architecture, design, and interactions. This visual modeling approach is instrumental in conveying complex concepts and structures in a more accessible manner, fostering effective communication and collaboration among stakeholders involved in the software development process.

UML's versatility lies in its ability to transcend the intricacies of coding languages, making it accessible to a broad audience, including software architects, designers, developers, and other project stakeholders. The language achieves this by offering a standardized set of symbols, notations, and diagrams, which collectively form a visual vocabulary for expressing various aspects of software systems.

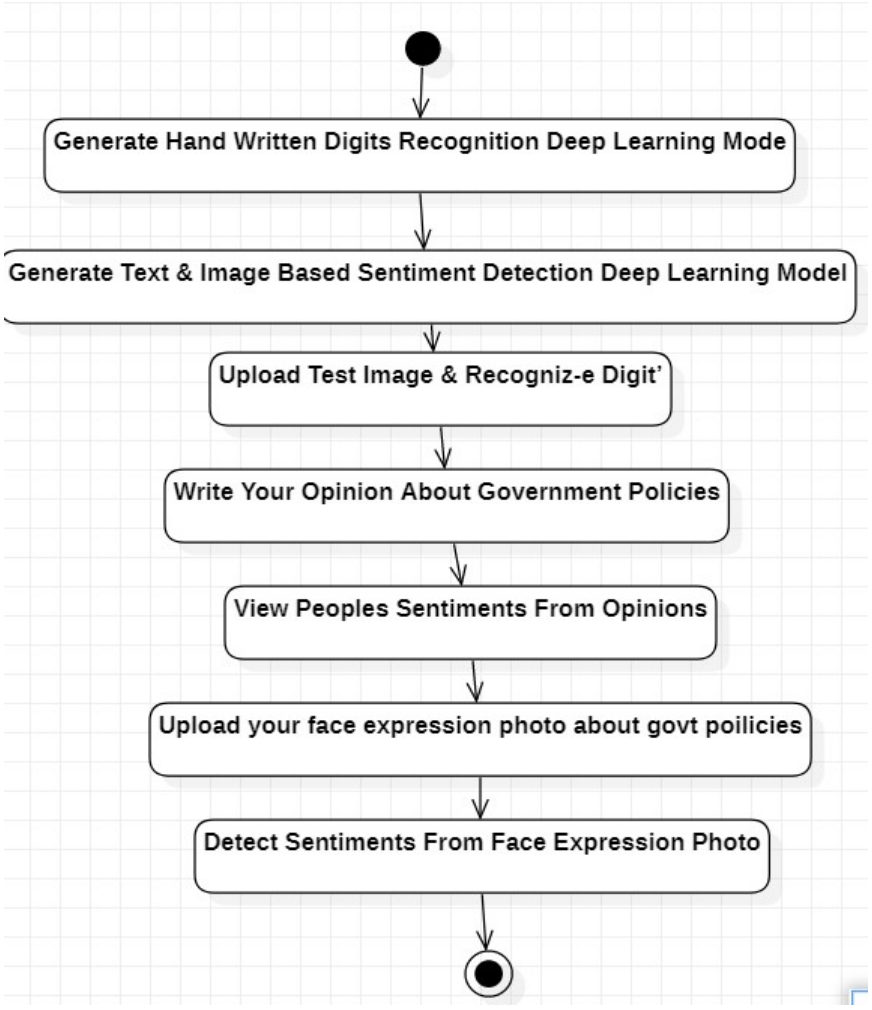
The key objective of UML is to establish a modeling language that is both adaptable and easily understandable. This adaptability ensures that UML can be effectively applied across diverse software development methodologies and project scenarios. The language is designed to accommodate the needs of individuals with varying technical backgrounds, promoting a shared understanding of the software system throughout the development lifecycle.

In essence, UML serves as a crucial tool for facilitating a common language among multidisciplinary teams, allowing them to collaboratively create, analyze, and refine software models. This collaborative modeling process enhances the precision and coherence of software

representations, ultimately contributing to the successful development and implementation of sophisticated software systems.

5.1.1 Activity Diagram:

An activity diagram holds crucial importance as a behavioral diagram employed in UML diagrams, serving the purpose of showcasing the dynamic attributes of a system. It serves as a more intricate version of a flowchart, providing a visual representation of the information flow between various activities. Activity diagrams effectively portray the dynamic behavior of a system.



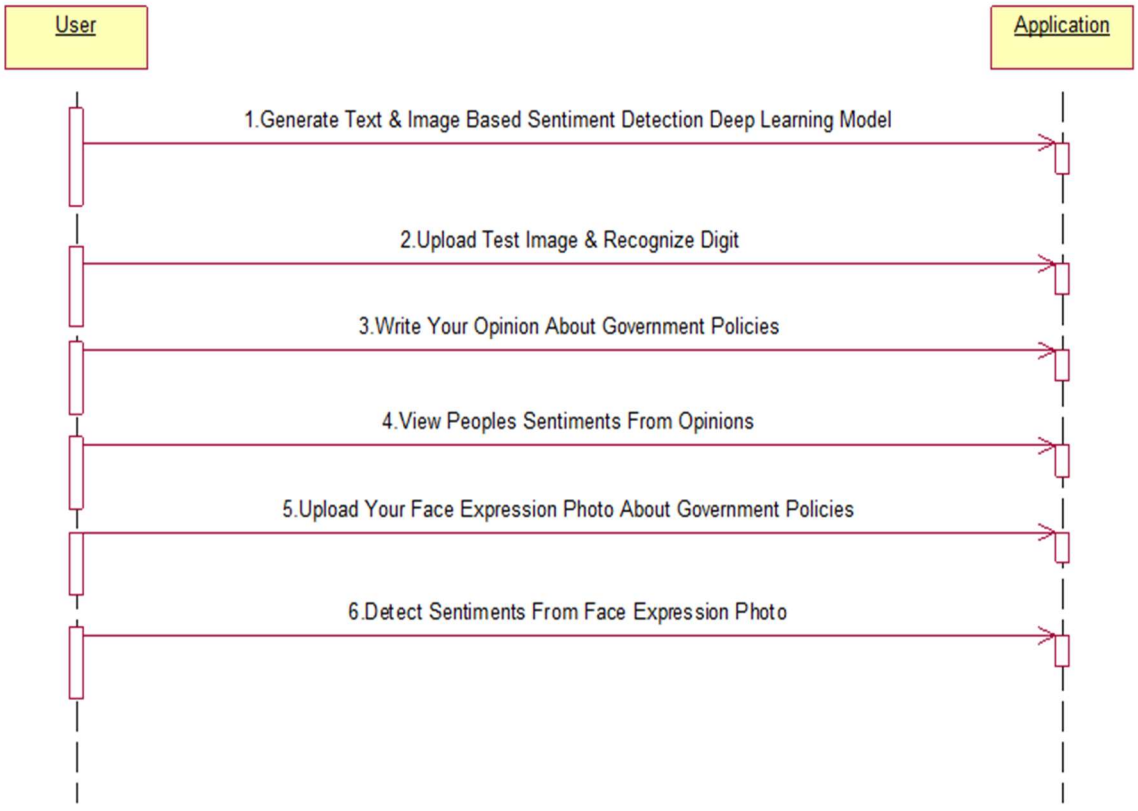
5.1.2. Sequence Diagram:

A sequence diagram, categorized as an interaction diagram in UML, illustrates the sequenceand order of interactions among system processes.

It utilizes a message sequence chart to depict these interactions. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

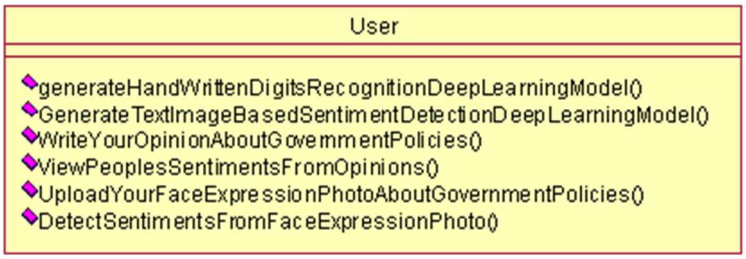
Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

Sequence diagrams are alsoknown as event diagrams or timing diagrams.



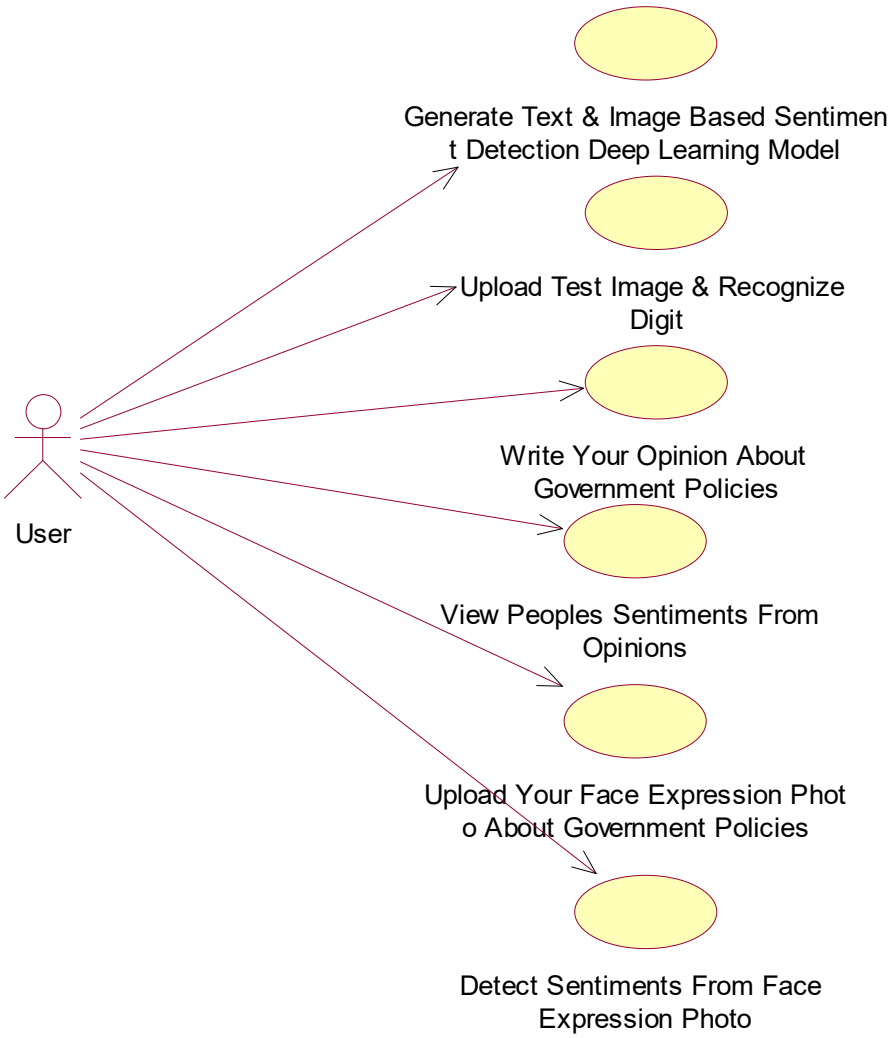
5.1.3 Class Diagram:

The class diagram offers a static perspective of an application, presenting a visual depiction that focuses on its unchanging elements. It serves multiple purposes, including visualization, description, and documentation of various system components. Moreover, it aids in the creation of executable software programs. The class diagram showcases the characteristics and behaviors of classes, as well as the constraints imposed on the system. Due to their direct translation to object-oriented languages, class diagrams find extensive application in the design of object-oriented systems. They present a representation of classes, interfaces, relationships, collaborations, and constraints, earning them the title of structural diagrams. Class diagrams play a critical role in creating component and deployment diagrams, as well as in supporting forward and backward engineering processes.



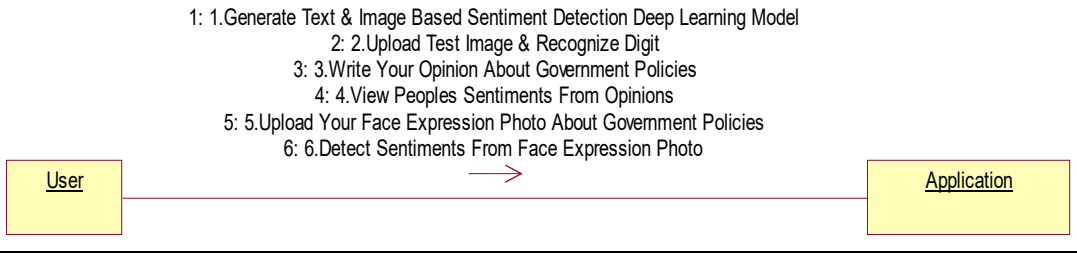
5.1.4. Use Case Diagram:

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



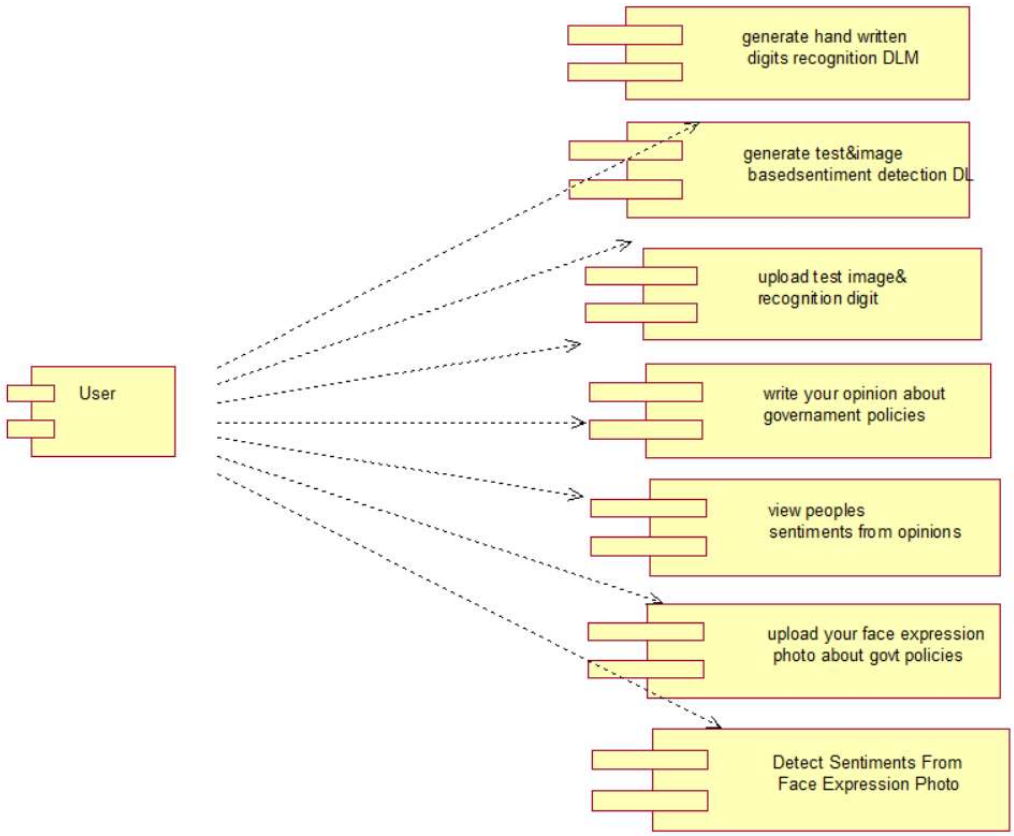
5.1.5 Collaboration Diagram:

A collaboration diagram visually maps out interactions among objects by illustrating the sequence of messages exchanged. It consolidates information from class, sequence, and use case diagrams, offering a comprehensive depiction of a system's static structure and dynamic behavior. By portraying how objects collaborate through message passing, it provides insights into both the relationships between objects and the flow of control within the system, enhancing understanding of its functionality and design.



5.1.6. Component Diagram:

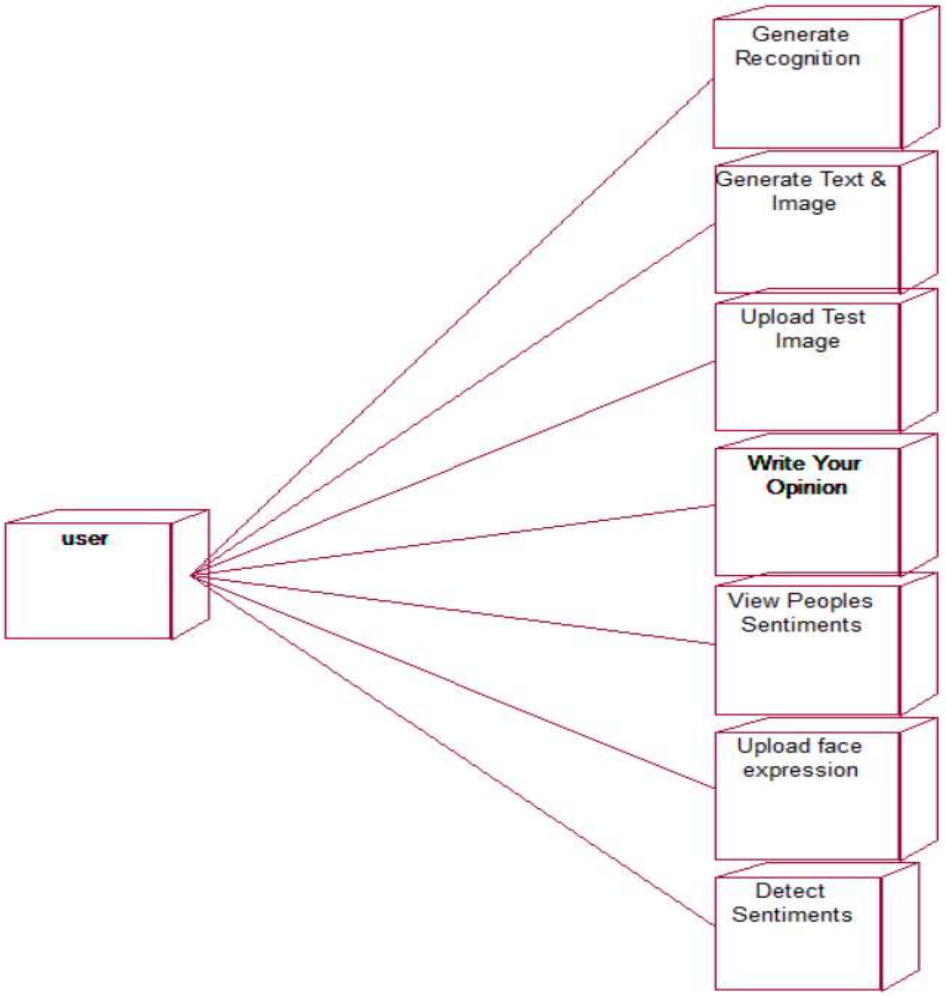
In the Unified Modeling Language (UML), a component diagram illustrates the interconnection of components to build larger components or software systems, portraying the structure of intricate systems. Assembly connectors link components, connecting the required interface of one with the provided interface of another, showcasing the service consumer-service provider dynamic between them.



5.1.7 Deployment Diagram:

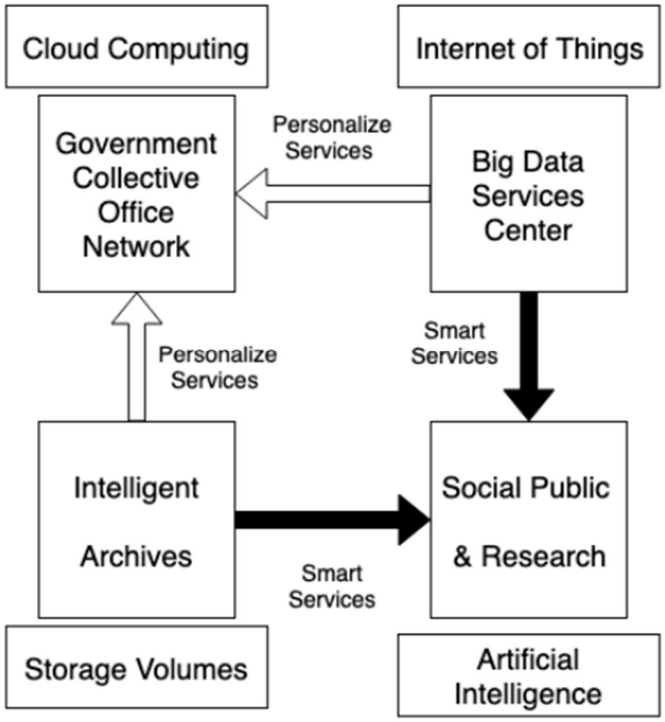
In the Unified Modeling Language, a deployment diagram visualizes how artifacts are physically deployed on nodes. For instance, it would illustrate the hardware components (nodes) like web servers, application servers, and database servers, the software components (artifacts) running on each node such as web applications and databases, and the connections between them like JDBC, REST, or RMI.

Nodes are depicted as boxes, with rectangles inside representing allocated artifacts. Nodes can have sub-nodes shown as nested boxes, and a single node might symbolize multiple physical nodes, like a database server cluster.



5.1.8 Architecture:

An architecture diagram, based on UML, is employed by system designers and developers to illustrate the high-level structure of a system or application. It serves to verify that the system meets user requirements and can also describe design patterns present in the system. The architecture diagram provides a means to abstract the overall framework of a software system, defining boundaries, relationships, and constraints between components. It offers a comprehensive view of the physical deployment and evolution plan of the software system. Developers and designers find this diagram highly useful in their work.



6. IMPLEMENTATION

6.1 Python:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language.

It provides constructs that enable clear programming on both small and large scales. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large & comprehensive standard library.

Over time, Python's ecosystem has developed significantly, particularly in its capacity for statistical analysis. It strikes a fine balance between scalability and elegance, prioritizing efficiency and code readability. Its design emphasizes readability and simplicity, making it beginner-friendly with a concise syntax, notably employing indentation. Key features include dynamic system functions and automatic memory management. Python's extensive libraries, user-friendly nature, and strong community support have solidified its position as a leading language for machine learning and data research, driving its popularity in various fields.

Applications of Python:

Python is used in many application domains. It makes its presence in every emerging field. It is the fastest-growing programming language and may be used to create any type of application.

- Web Applications. We can use Python to develop web applications. ...
- Desktop GUI Applications. ...
- Console-based Application. ...
- Software Development. ...
- Scientific and Numeric. ...
- Business Applications. ...
- Audio or Video-based Applications. ...
- 3D CAD Applications.

6.2 DJANGO:

Django, a top-tier Python Web framework, promotes swift development and emphasizes a clear, practical structure. Crafted by seasoned developers, it handles many tedious aspects of Web development, freeing you to concentrate on app creation without duplicating effort. It's both free and open source. Django simplifies the construction of intricate, database-centric websites, stressing component reusability, rapid development, and the "don't repeat yourself" principle.

6.3 CODE:

EGovernment.py

Invalid Python interpreter selected for the project

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import matplotlib.pyplot as plt
import numpy as np
from sklearn.externals import joblib
from keras.models import load_model
from keras.preprocessing.image import img_to_array
import cv2
from keras.models import model_from_json
from keras.preprocessing import image
from keras.optimizers import Adam
from keras.utils import np_utils
from keras.preprocessing import image
import os
from numpy import dot
from numpy.linalg import norm
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D
import imutils
import nltk

```

```
main = tkinter.Tk()
main.title("Enhancing Public Services through AI Automation")
main.geometry("1300x1200")

global filename
global text_sentiment_model
EMOTIONS = ["angry","disgust","scared", "happy", "sad", "surprised","neutral"]
global face_detection
global image_sentiment_model
global digits_cnn_model

1 usage
def digitModel():
    global digits_cnn_model
    with open('models/digits_cnn_model.json', "r") as json_file:
            loaded_model_json = json_file.read()
            digits_cnn_model = model_from_json(loaded_model_json)

    digits_cnn_model.load_weights("models/digits_cnn_weights.h5")
    digits_cnn_model._make_predict_function()
    print(digits_cnn_model.summary())
    text.insert(END,'Digits based Deep Learning CNN Model generated\n')
```

```
75      output = imutils.resize(orig, width=400)
76      cv2.putText(output, "Digits Predicted As : "+predicted, (10, 25),  cv2.FONT_HERSHEY_SIMPLEX,0.7, (0, 255, 0), 2)
77      cv2.imshow("Predicted Image Result", output)
78      cv2.waitKey(0)
79
80
        1 usage
81  def opinion():
82      user = simpledialog.askstring("Please enter your name", "Username")
83      opinion = simpledialog.askstring("Government Service Opinion", "Please write your Opinion about government services & policies")
84      f = open("Peoples_Opinion/opinion.txt", "a+")
85      f.write(user+"#"+opinion+"\n")
86      f.close()
87      messagebox.showinfo("Thank you for your opinion", "Your opinion saved for reviews")
88
89
        3 usages (2 dynamic)
90  def stem(textmsg):
91      stemmer = nltk.stem.PorterStemmer()
92      textmsg_stem = ''
93      textmsg = textmsg.strip("\n")
94      words = textmsg.split(" ")
95      words = [stemmer.stem(w) for w in words]
96      textmsg_stem = ' '.join(words)
97      return textmsg_stem
98
        1 usage
```

```
def uploadPhoto():
    filename = filedialog.askopenfilename(initialdir="expression_images_to_upload")
    user = simpledialog.askstring("Please enter your name", "Username")
    policy = simpledialog.askstring("Please enter Government Policy name related to Facial Expression", "Please enter Government Policy name relate
    img = cv2.imread(filename)
    cv2.imwrite("sentimentImages/"+user+"-"+policy+".jpg",img);
    messagebox.showinfo("Your facial expression image accepted for reviews", "Your facial expression image accepted for reviews")

1 usage
def photoSentiment():
    filename = 'sentimentImages'
    for root, dirs, files in os.walk(filename):
      for fdata in files:
          frame = cv2.imread(root+"/"+fdata)
          faces = face_detection.detectMultiScale(frame,scaleFactor=1.1,minNeighbors=5,minSize=(30,30),flags=cv2.CASCADE_SCALE_IMAGE)
          msg = ''
          if len(faces) > 0:
              faces = sorted(faces, reverse=True,key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
              (x, y, w, h) = faces
              cv2.rectangle(frame, (x,y), (x+w,y+h), (0,0,255), 2)
              temp = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
              roi = temp[y:y + h, x:x + w]
              roi = cv2.resize(roi, (48, 48))
              roi = roi.astype("float") / 255.0
              roi = img_to_array(roi)
              roi = np.expand_dims(roi, axis=0)
              preds = image_sentiment_model.predict(roi)[0]
              emotion_probability = np.max(preds)
              label = EMOTIONS[preds.argmax()]
```

```
              msg = "Sentiment detected as : "+label
              img_height, img_width = frame.shape[:2]
              cv2.putText(frame, msg, (50,40), cv2.FONT_HERSHEY_SIMPLEX, 0.5,(0,0,255), 2)
              cv2.imshow(fdata,frame)
              messagebox.showinfo(fdata, "Sentiment predicted from Facial expression as : "+label)
              if cv2.waitKey(10) & 0xFF == ord('q'):
                  break
    cv2.waitKey(0)
    cv2.destroyAllWindows()


font = ('times', 16, 'bold')
title = Label(main, text='Automating E-Government Services With Artificial Intelligence',anchor=W, justify=CENTER)
title.config(bg='yellow4', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)


font1 = ('times', 14, 'bold')
digitButton = Button(main, text="Generate Hand Written Digits Recognition Deep Learning Model", command=digitModel)
digitButton.place(x=50,y=100)
digitButton.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='yellow4', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=50,y=150)
```

```
1 usage
def viewSentiment():
    text.delete('1.0', END)
    with open("Peoples_Opinion/opinion.txt", "r") as file:
        for line in file:
            line = line.strip('\n')
            line = line.strip()
            arr = line.split("#")
            text_processed = stem(arr[1])
            X =  [text_processed]
            sentiment = text_sentiment_model.predict(X)
            predicts = 'None'
            if sentiment[0] == 0:
                predicts = "Negative"
            if sentiment[0] == 1:
                predicts = "Positive"
            text.insert(END,"Username : "+arr[0]+"\n");
            text.insert(END,"Opinion  : "+arr[1]+" : Sentiment Detected As : "+predicts+"\n\n")

1 usage
def uploadPhoto():
    filename = filedialog.askopenfilename(initialdir="expression_images_to_upload")
    user = simpledialog.askstring("Please enter your name", "Username")
    policy = simpledialog.askstring("Please enter Government Policy name related to Facial Expression", "Please enter Government Policy name relate
    img = cv2.imread(filename)
    cv2.imwrite("sentimentImages/"+user+"-"+policy+".jpg",img);
    messagebox.showinfo("Your facial expression image accepted for reviews", "Your facial expression image accepted for reviews")
```

```
sentimentButton = Button(main, text="Generate Text & Image Based Sentiment Detection Deep Learning Model", command=sentimentModel)
sentimentButton.place(x=50,y=200)
sentimentButton.config(font=font1)

recognizeButton = Button(main, text="Upload Test Image & Recognize Digit", command=digitRecognize)
recognizeButton.place(x=50,y=250)
recognizeButton.config(font=font1)

opinionButton = Button(main, text="Write Your Opinion About Government Policies", command=opinion)
opinionButton.place(x=50,y=300)
opinionButton.config(font=font1)

viewButton = Button(main, text="View Peoples Sentiments From Opinions", command=viewSentiment)
viewButton.place(x=50,y=350)
viewButton.config(font=font1)

photoButton = Button(main, text="Upload Your Face Expression Photo About Government Policies", command=uploadPhoto)
photoButton.place(x=50,y=400)
photoButton.config(font=font1)

photosentimentButton = Button(main, text="Detect Sentiments From Face Expression Photo", command=photoSentiment)
photosentimentButton.place(x=50,y=450)
photosentimentButton.config(font=font1)


font1 = ('times', 12, 'bold')
text=Text(main,height=15,width=80)
scroll=Scrollbar(text)
```

```
198
199     font1 = ('times', 12, 'bold')
200     text=Text(main,height=15,width=80)
201     scroll=Scrollbar(text)
202     text.configure(yscrollcommand=scroll.set)
203     text.place(x=600,y=100)
204     text.config(font=font1)
205
206
207     main.config(bg='gray1')
208     main.mainloop()
209
```

7. TESTING

Testing aims to uncover errors by systematically examining every possible fault or weakness in a product. It verifies the functionality of individual components, sub-assemblies, and the final product, ensuring it meets requirements and user expectations without unacceptable failures. Different test types cater to specific testing needs, collectively ensuring thorough evaluation and validation of the software system.

7.1 TYPES OF TESTS

7.1.1 Unit testing:

Unit testing involves creating test cases to verify the correct functioning of internal program logic and to validate that inputs yield valid outputs. It encompasses testing individual software units after their completion but before integration, examining decision branches and code flow. This form of structural testing relies on an understanding of the unit's construction and is thorough. Unit tests assess specific business processes, applications, or system configurations, ensuring accurate performance according to documented specifications with clear inputs and expected outcomes.

7.1.2 Integration testing:

Integration tests are designed to assess the functionality of integrated software components, ensuring they operate seamlessly as a unified program. These tests are event-triggered and primarily focus on verifying the overall performance of screens or fields. They validate that although each component passed individual unit tests, their combination is cohesive and accurate. Integration testing is specifically targeted at identifying issues that arise from the combination of components.

7.1.3 Functional test:

Functional tests systematically demonstrate that the tested functions adhere to specified business and technical requirements, system documentation, and user manuals. They focus on various aspects:

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 emphasize requirements, key functions, and special test cases. It also ensures systematic coverage of business process flows, data fields, predefined processes, and successive processes. Additional tests are identified and the effectiveness of current tests is evaluated before completing functional testing.

7.1.4 System Test:

System testing validates whether the entire integrated software system meets specified requirements, ensuring known and predictable outcomes. It typically involves configuration-oriented system integration tests, emphasizing pre-defined process descriptions and flows, along with integration points.

7.1.5 White Box Testing:

White Box Testing involves testers having insight into the inner workings, structure, or purpose of the software. It explores areas inaccessible at the black box level.

7.1.6 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.* The test provides inputs and responds to outputs without considering how the software works.

7.1.7 Unit Testing:

Unit testing usually occurs alongside coding, forming a part of the software lifecycle. It ensures individual units' function correctly, often integrated with code development.

Test strategy and approach

Field testing is manual, while functional tests are meticulously scripted.

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.

7.1.8 Integration Testing:

Integration testing combines multiple software components incrementally to identify interface defects. It verifies seamless interaction between components or applications without errors.

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.

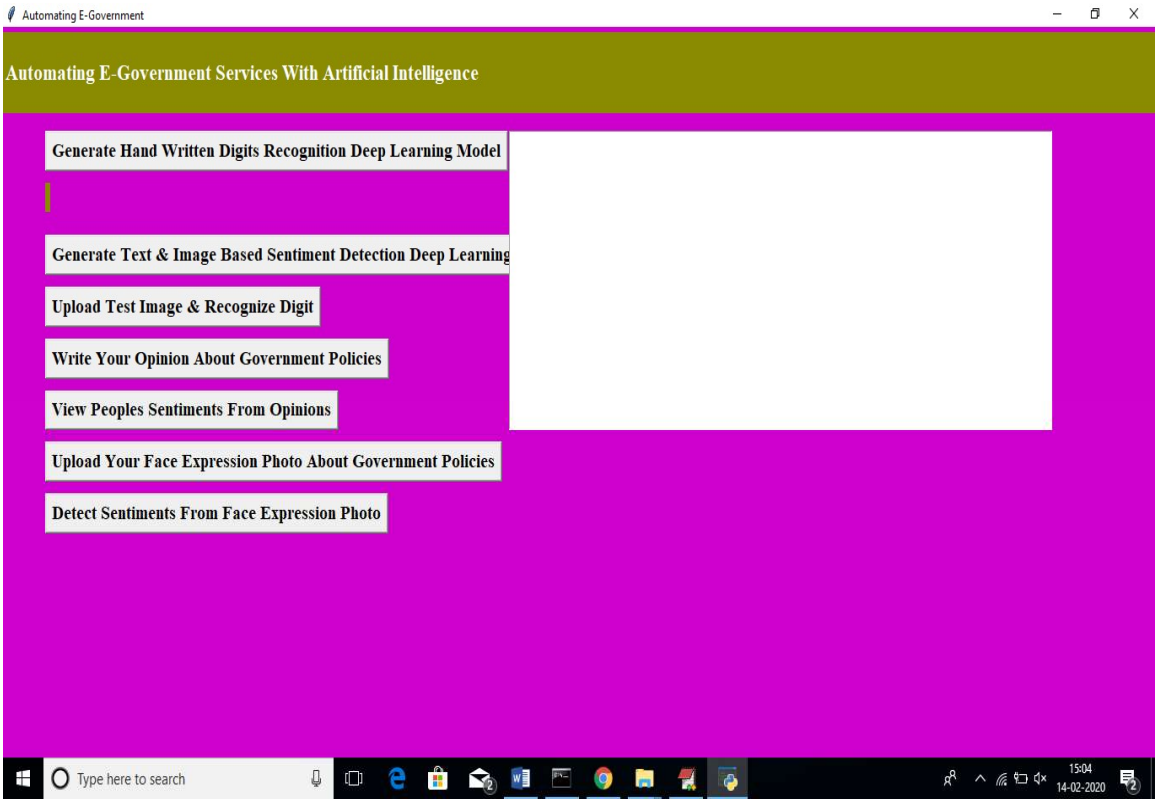
7.1.9 Acceptance Testing:

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.

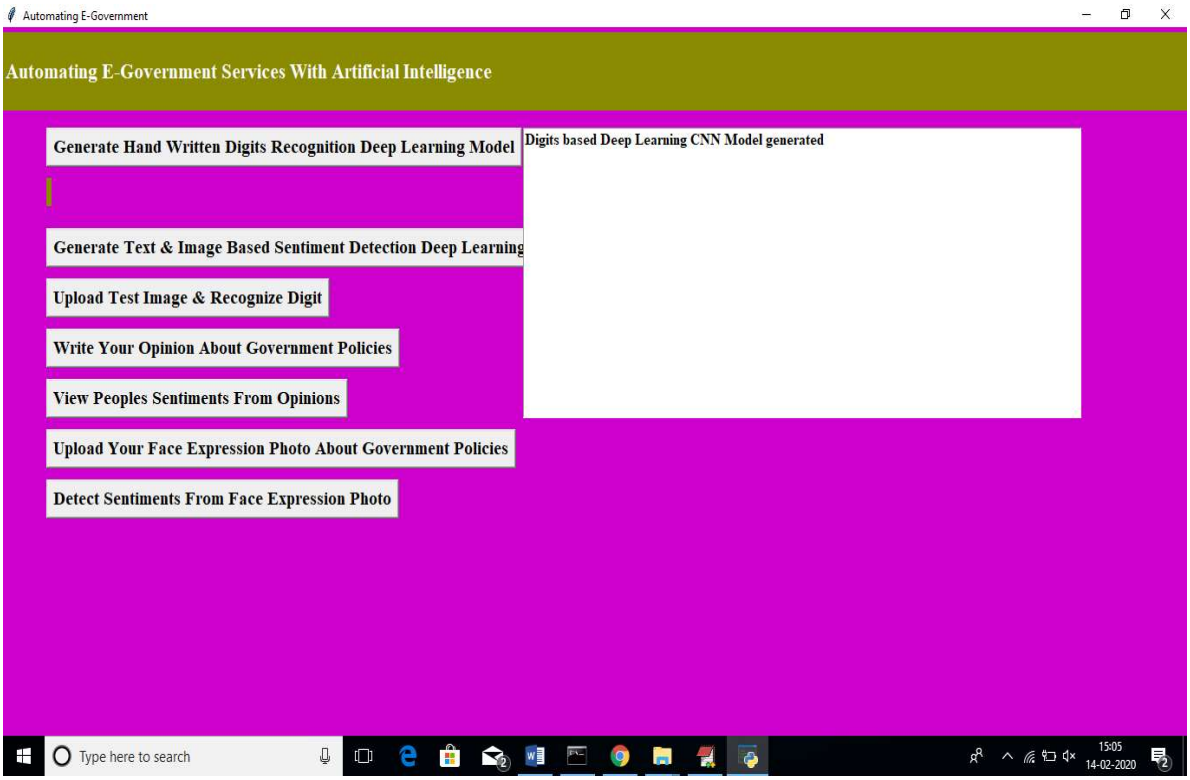
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8. WORKING OF FRONT-END AND BACK-END

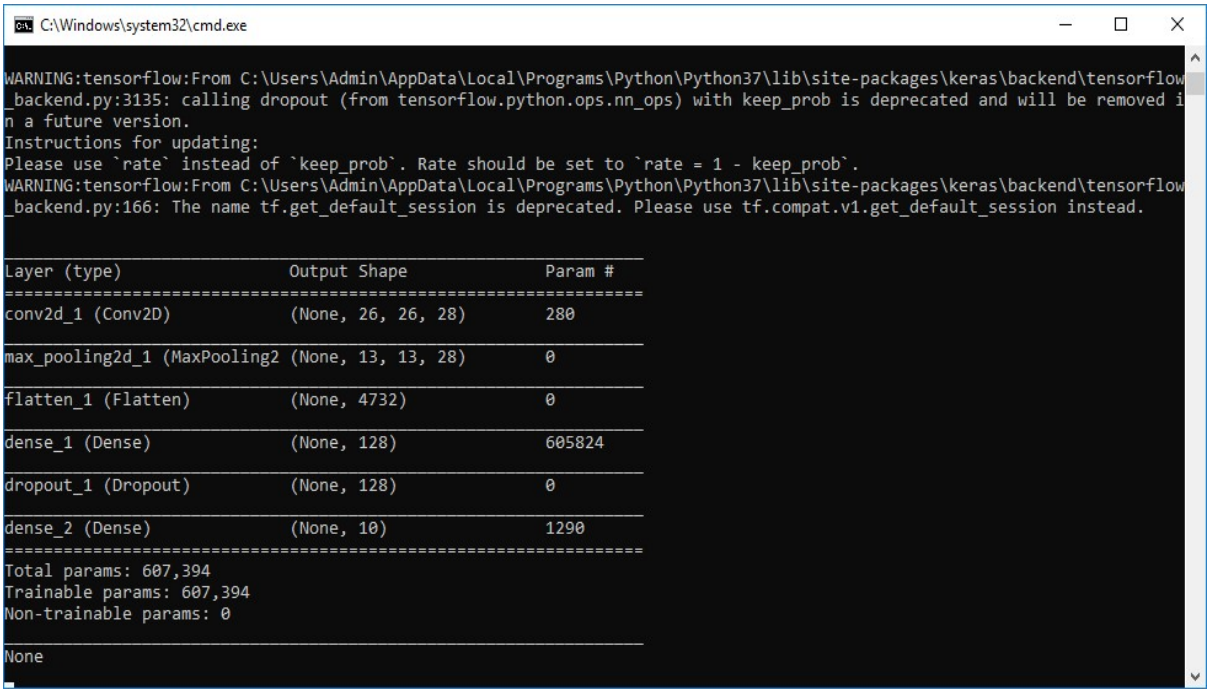
Home Page:



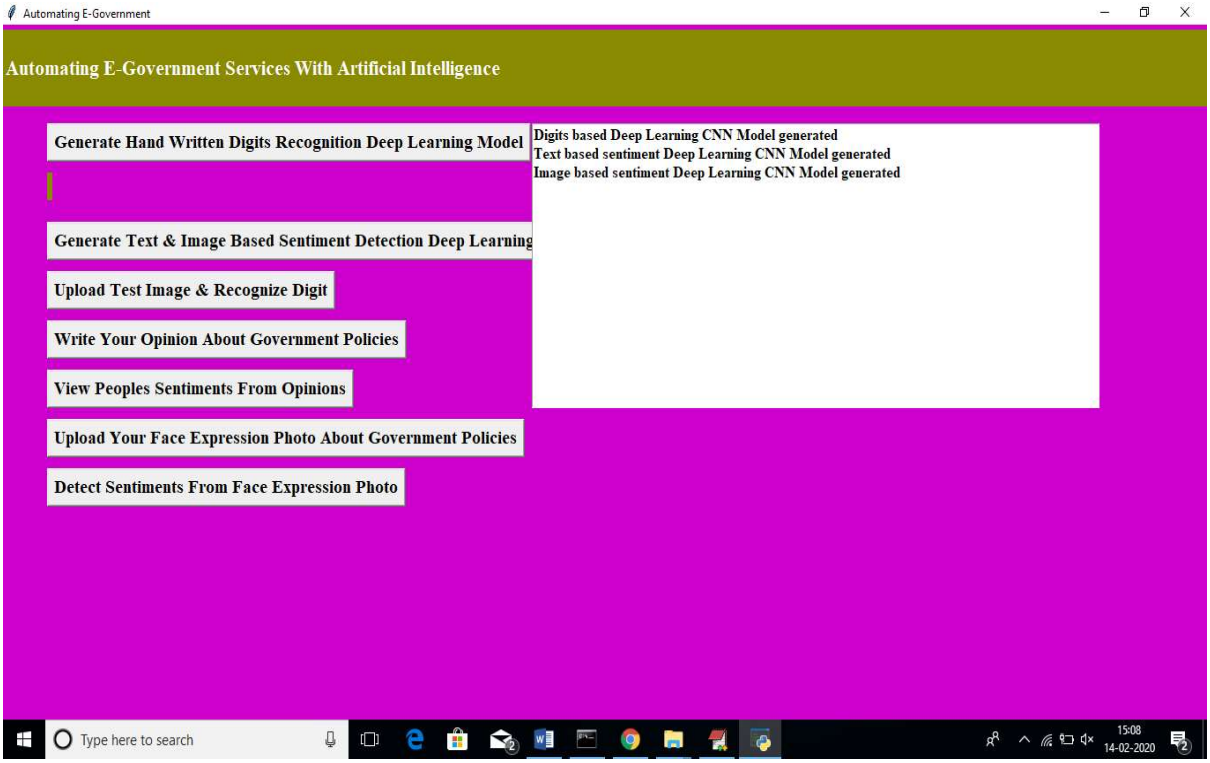
In above screen, click on 'Generate Hand Written Digits Recognition Deep Learning Model' button to generate CNN digits recognition model.



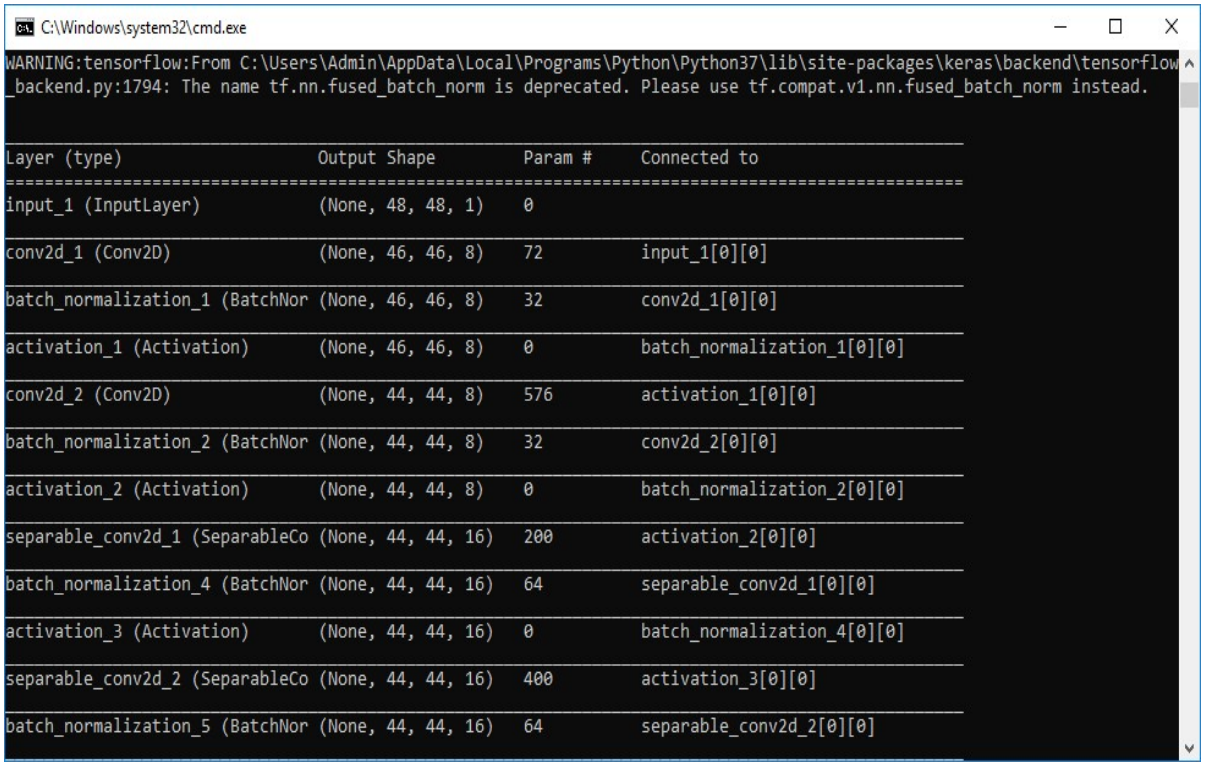
In above screen we can see digits model generated and CNN layer details.



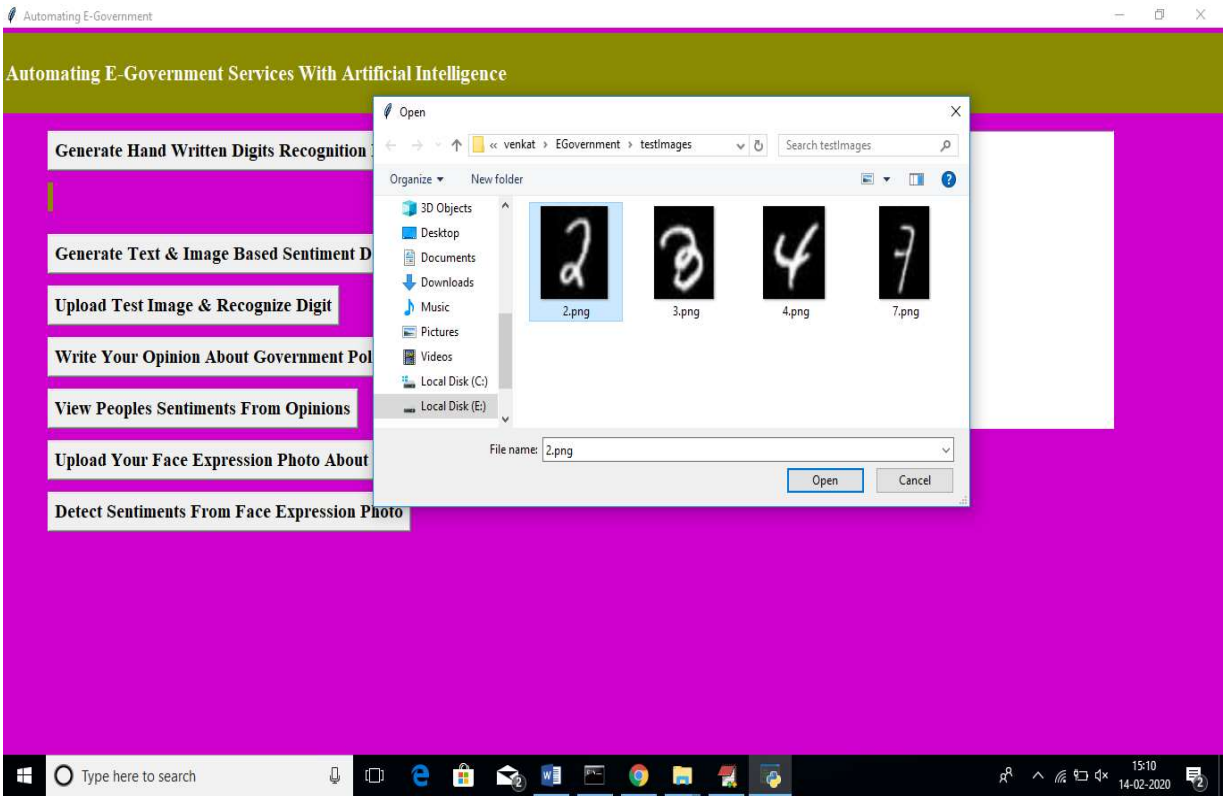
In above screen, we can see Conv2d means convolution or CNN generate image features layer from different size as first layer generate with image size 26, 26 and second generated with 13 and 13 and goes on. Now click on 'Generate Text & Image Based Sentiment Detection Deep Learning Model' button to generate CNN for text and image-based sentiment detection model.



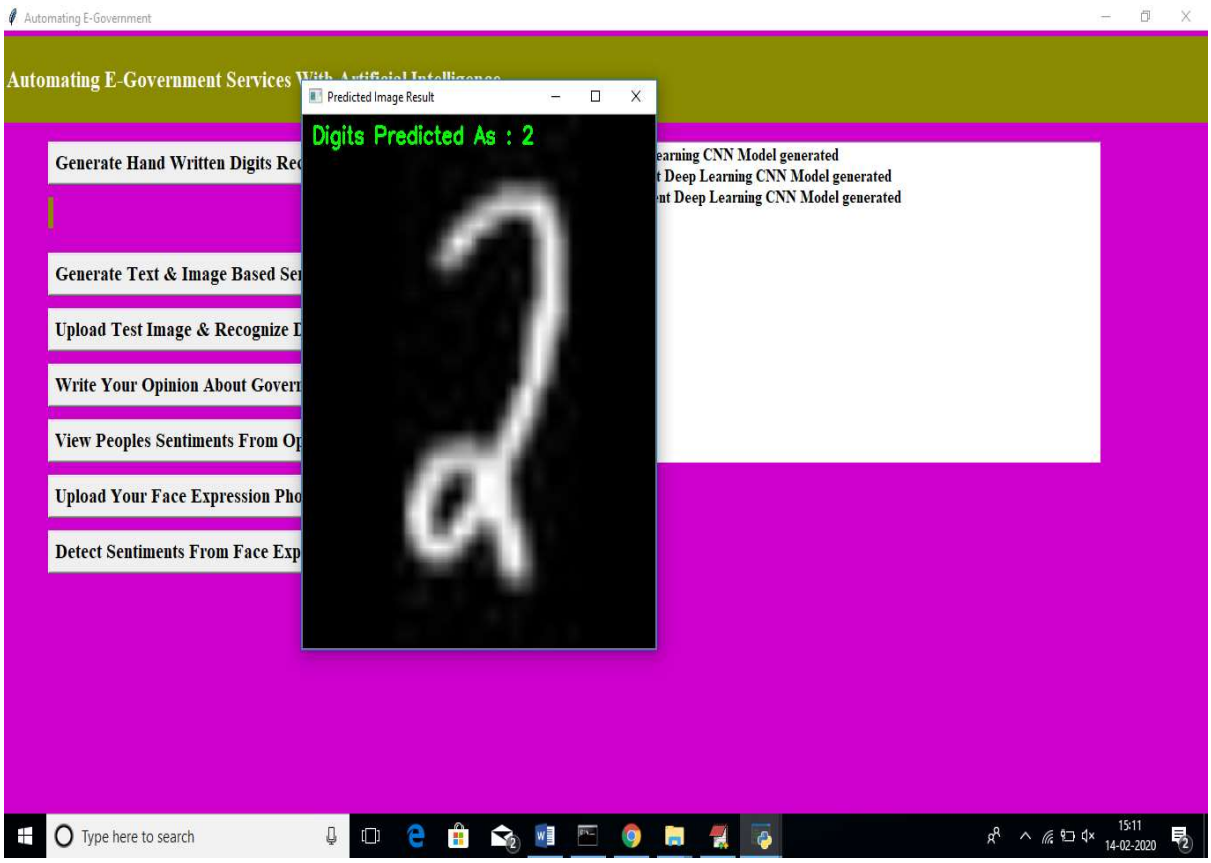
In above screen, we can see text and image-based CNN model generated. See backend for more details.



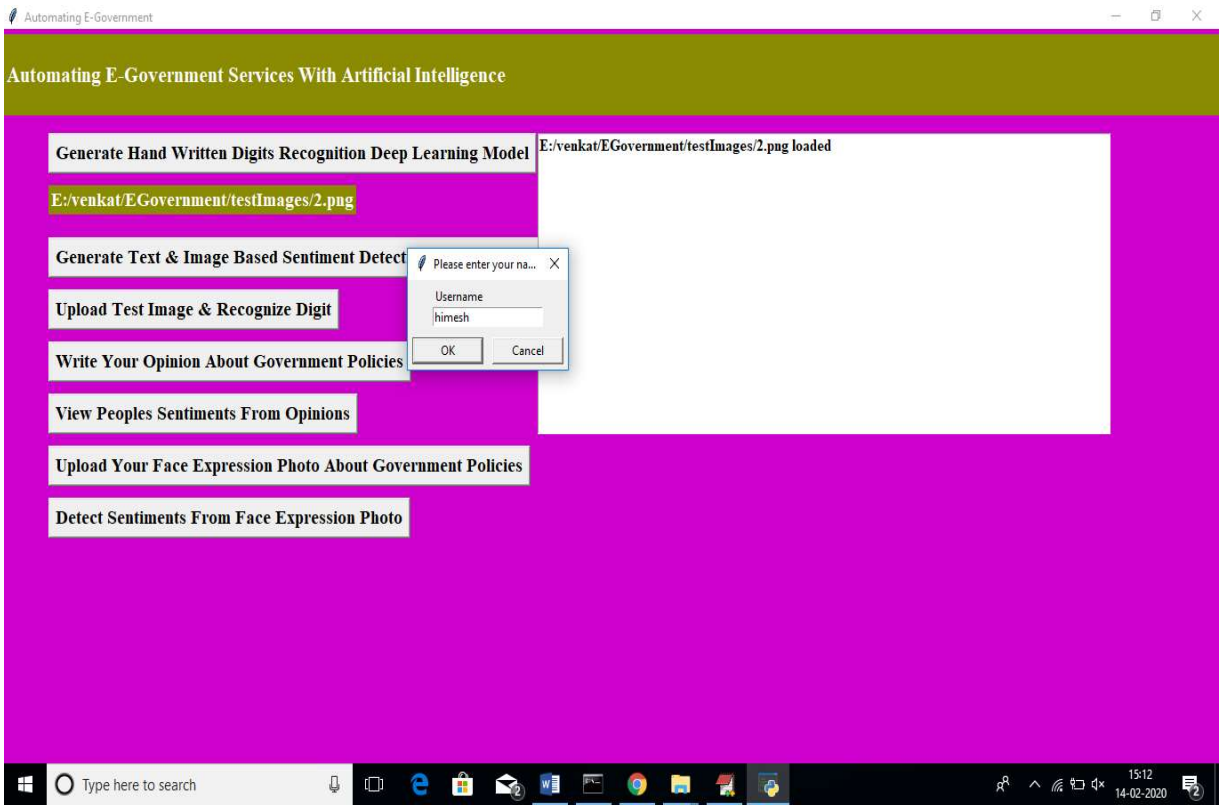
Now click on ‘Upload Test Image & Recognize Digit’ button to upload digit images and to get name of that digit. All digit images saved inside testImages folder.



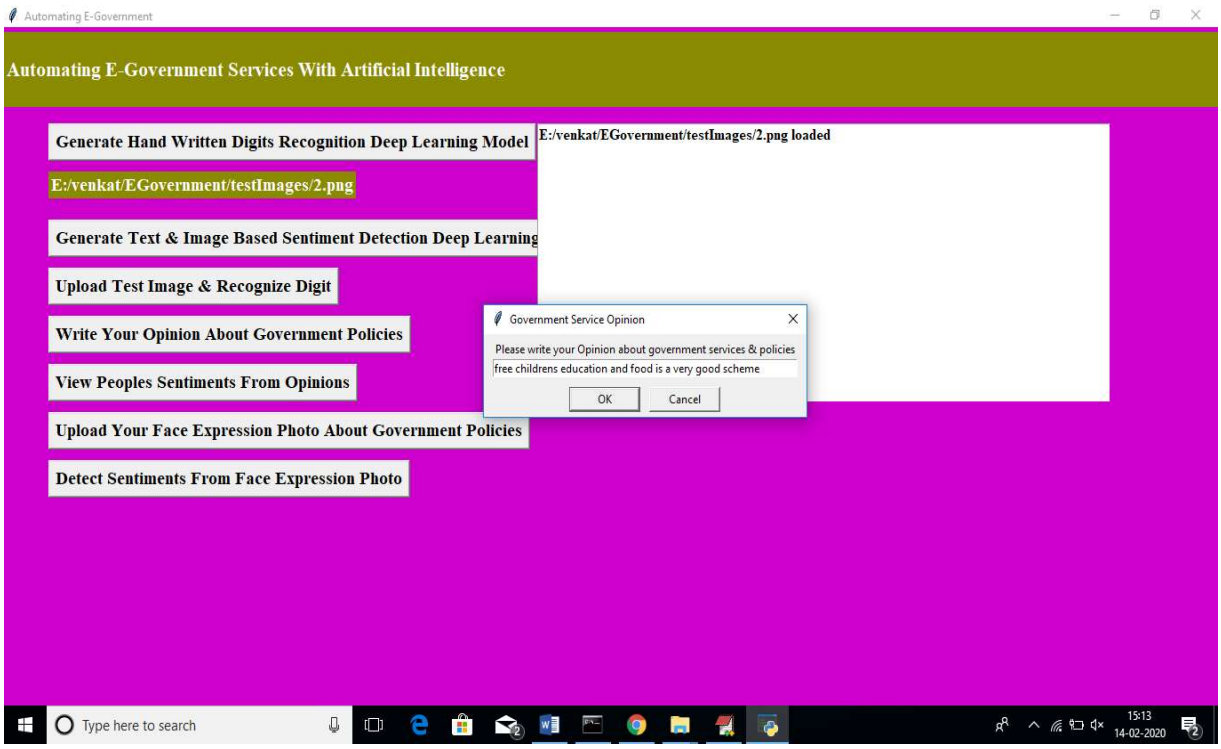
In above screen, we are uploading an image which contain digit 2 and below is the output of detection.



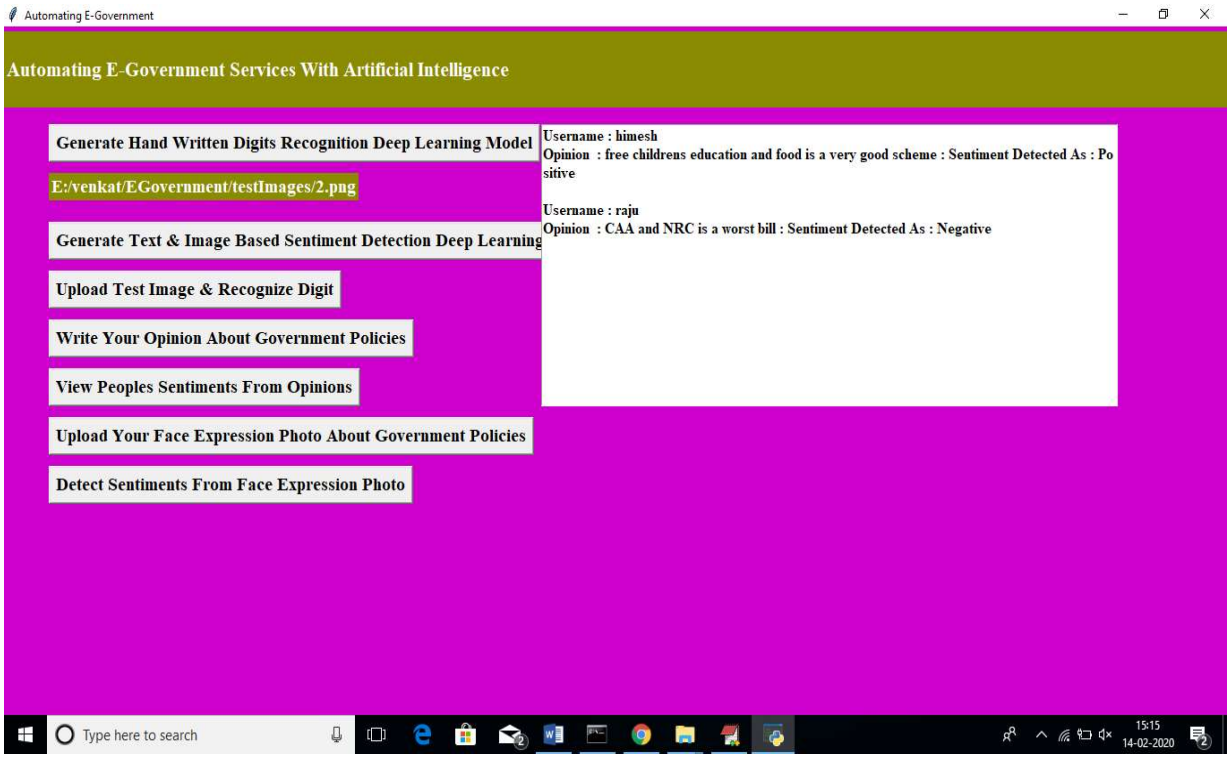
In above screen, we can see Digits Predicted as: 2. Now click on 'Write Your Opinion About Government Policies' button to write some comments on government policy.



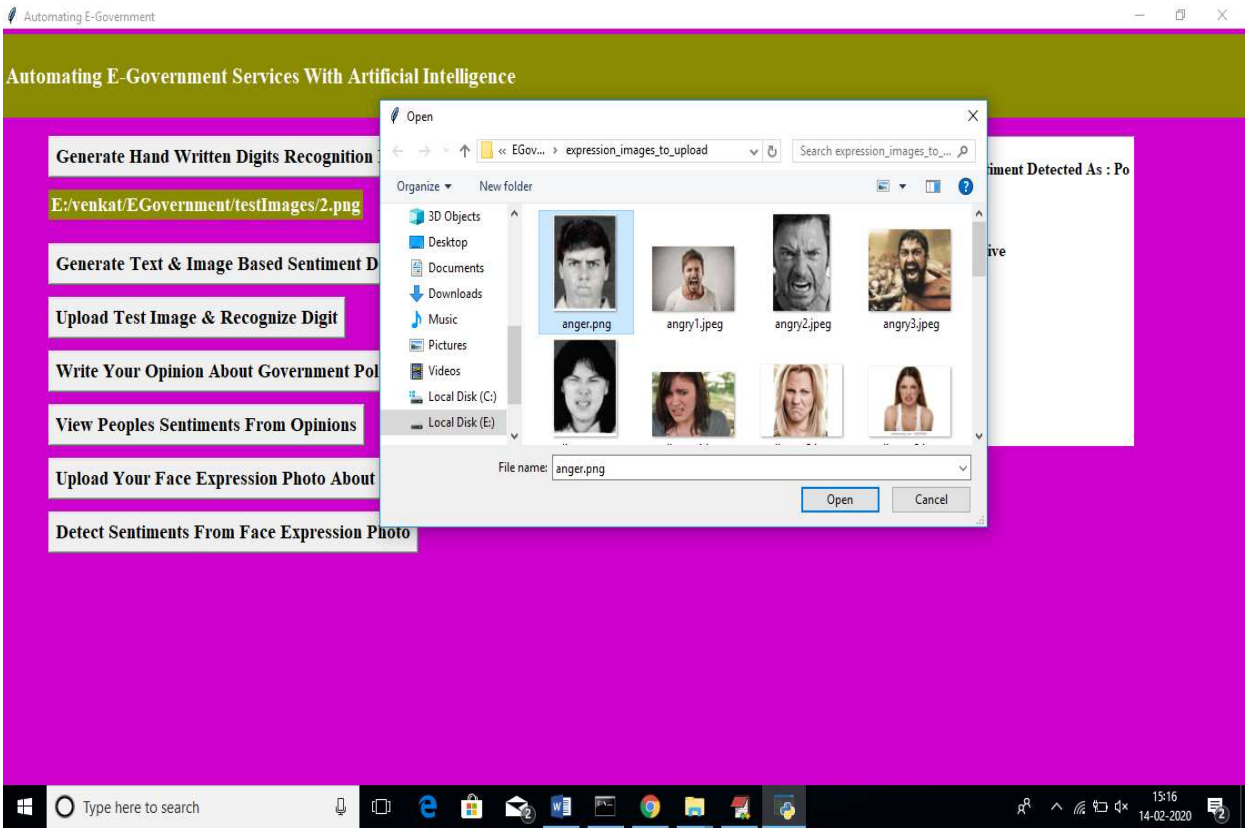
In above screen before writing opinions we need to write username after writing username click ok button to get below screen



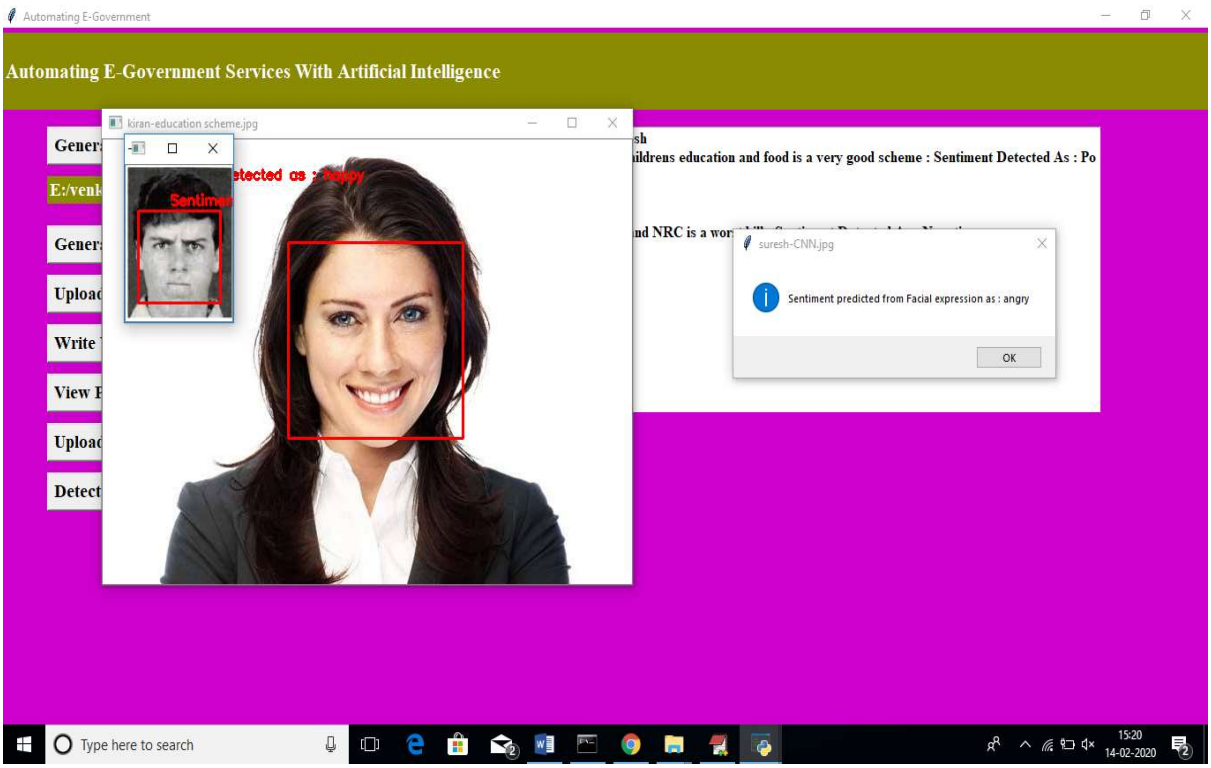
In above screen I wrote some comment on some scheme and application detect sentiment from it as positive or negative. Now click on 'View Peoples Sentiments From Opinions' button to view all opinions from past users.



In above screen text area, we can see opinions from all users and in first opinion we got sentiment detected as positive which means user is satisfy with that scheme and for second opinion, we got sentiment as negative which means user not happy. Similarly, user can upload their image with facial expression which describe whether user is happy or angry.



In above screen I am uploading one anger face image and then application ask to write username and referring scheme name. Similarly, any number of users can upload their images. Now click on 'Detect Sentiments From Face Expression Photo' button to get all images and its detected sentiments



In above screen we can see all images with facial expression are identified with their sentiments. In dialog box also we can see sentiment result.

Similarly, you can enter any number of comments or facial images to detect their sentiments.

9. CONCLUSION

With the recent strides in AI and deep learning technologies, governmental agencies increasingly embrace these tools to enhance their systems and services. Nonetheless, several challenges impede their adoption, such as the scarcity of expertise, computational resources, trust issues, and the interpretability of AI.

In this paper, we define artificial intelligence and e-government, briefly review global e-government indices, and propose solutions to enhance e-government, focusing on the Gulf Countries. We introduce a comprehensive framework for managing government information resources to oversee the entire e-government lifecycle. Additionally, we suggest employing deep learning techniques to streamline various e-government services and present a smart AI platform for development and implementation within e-government.

The primary objective of this paper is to introduce novel frameworks and platforms that amalgamate recent advancements in AI techniques into e-government systems and services, aiming to bolster trust, transparency, and efficiency in e-government operations.

10. FUTURE ENHANCEMENTS

1. Advanced AI Algorithms: Integrate cutting-edge AI algorithms such as reinforcement learning, natural language processing, and generative adversarial networks to improve the accuracy and efficiency of automated e-government services.

2. **Personalized Services**: Develop AI-driven systems capable of providing personalized public services tailored to individual citizens' needs and preferences, enhancing user experience and satisfaction.

3. Predictive Modeling: Utilize predictive modeling and machine learning algorithms to anticipate citizen demands, detect emerging trends, and proactively address potential service disruptions or issues before they occur.

4. **Multi-channel Integration**: Enhance service accessibility by integrating AI-powered virtual assistants and chatbots across various communication channels, including websites, mobile apps, social media platforms, and voice-enabled devices.

5. Blockchain Integration: Explore the integration of blockchain technology with AI automation to enhance data security, transparency, and integrity in e-government transactions and processes.

6. Continuous Innovation: Foster a culture of continuous innovation and experimentation, encouraging government agencies to explore and implement emerging AI technologies and best practices to further enhance public service delivery and citizen satisfaction.

By incorporating these future enhancements, the project can significantly advance the goal of leveraging AI automation to enhance public services, foster greater efficiency, transparency, and responsiveness in government operations, and ultimately improve the quality of life for citizens.

11. BIBLIOGRAPHY

1. Abdulaziz Al-Besher, Kailash Kumar "Use of artificial intelligence to enhance e-government services" College of Computing and Informatics, Saudi Electronic University, Riyadh, 11673, Saudi Arabia. https://www.sciencedirect.com/science/article/pii/S2665917422001180

2. "The rise of GovTech: Trojan horse or blessing in disguise? "A research agenda Nitesh Bharosa Delft University of Technology, The Netherlands.

   https://www.sciencedirect.com/science/article/pii/S0740624X22000259#ab0005

3. "Enhancing the adoption of digital public services: Evidence from a large-scale field experiment "Ari Hyytinen, Jarno Tuimala , Markus Hammar Hanken and Helsinki GSE, Finland b Finnish Tax Administration, Finland , 2021.

   https://www.sciencedirect.com/science/article/pii/S0740624X2200020X

4. "Digitalization as a policy response to social acceleration: Comparing democratic problem solving in Denmark and the Netherlands" Kees van Kersbergen , Barbara Vis . Aarhus University, Department of Political Science, School of Business and Social Sciencest, Netherlands 2020. https://www.sciencedirect.com/science/article/pii/S0740624X22000405

5. "Designing for successful online engagement: Understanding technological frames of citizen and police users of community policing platforms " Mark van der Giessen , Petra Saskia Bayerl b a Aalto University School of Business, Ekonominaukio 1, 02150 Espoo, Finland b Centre of Excellence in Terrorism, Resilience, Intelligence and Organised Crime Research (CENTRIC), Sheffield Hallam University, 20 Furnival Street, Sheffield S1 2NU, United Kingdom, 2022. https://www.sciencedirect.com/science/article/pii/S0740624X22000442

6. "Automating E-Government Services With Artificial Intelligence" Omar Saeed Al-Mushayt Department of MIS, King Khalid University, Abha 61421, Saudi Arabia ,2019 https://ieeexplore.ieee.org/document/8862835/authors#authors

7. J.R. Gil-Garcia, J. Zhang, G. Puron-Cid "Conceptualizing smartness in government: an integrative and multi-dimensional view"

https://www.sciencedirect.com/science/article/abs/pii/S0740624X16300284

8. "E-Governance in Smart Cities": Global Trends and Key Enablers by Aleksandra Kuzior 1,*ORCID,Olena Pakhnenko 2ORCID,Inna Tiutiunyk 2 and Serhiy Lyeonov 3, 2023 https://www.mdpi.com/2624-6511/6/4/78

9. "Artificial intelligence and effective governance: A review, critique and research agenda" Gagan Deep Sharma PhD, Anshita Yadav, Ritika Chopra. Artificial intelligence Environmental sustainability Governance Policy making Public Administration ICT, 2021 https://www.sciencedirect.com/science/article/pii/S2666188819300048#abs0001

10. " Artificial Intelligence in E-Governance in India: Exploring the potential applications, benefits, and ethical implications of integrating AI technologies in electronic governance systems to enhance service delivery and efficiency in India. Iqra Amin, Innabat Amin, Research Scholar Public Administration, University School of Social Sciences Rayat Bahra University, Punjab.

https://www.ijfans.org/uploads/paper/581a32d653ee14181122186eb1831433.pdf

11. "Governance of artificial intelligenc"e Araz TaeihaghPolicy and Society, Volume 40, Issue 2, June 2021, Pages 137–157, 04 June 2021.

https://academic.oup.com/policyandsociety/article/40/2/137/6509315?login=false

12. "THE CONCEPT AND IMPLEMENTATION IN THE EMERGING ECONOMY" Dhruba Lal Pandey, Nischal Risal, Tribhuvan University, P.O.Box 148, Lalitpur, Code-44700. https://www.researchgate.net/publication/344650302_E-governance_A_study_of_the_concept_and_implementation_in_the_emerging_economy