

TRUSTWORTHINESS ASSISSMENT OF USERS IN SOCIAL REVIEWING SYSTEMS

A Project report submitted in partial fulfillment
for the award of the degree of
MASTER OF COMPUTER APPLICATIONS

(2022-2024)

Submitted by

G. GREESHMA SAI
(22G21F0029)

Under the esteemed guidance of

Mrs. M. NARMADHA

Assistant Professor



Department of Master of Computer Applications

AUDISANKARA COLLEGE OF ENGINEERING & TECHNOLOGY
(AUTONOMUS)

(Accredited by NAACA+)

Approved by AICTE, Affiliated to JNTUA,

Ananthapuramu, Tirupati (DT), Andhra Pradesh.

NH-5, Bypass Road, Gudur, Tirupati (Dt.)

www.audisankara.ac.in

2022-24

AUDISANKARA COLLEGE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

(Accredited by NAACA+)

Approved by AICTE, Affiliated to JNTUA,
Ananthapuramu, Tirupati(Dt.), Andhra Pradesh



This is to certify that the project report entitled “**TRUSTWORTHINESS ASSESSMENT OF USERS IN SOCIAL REVIEWING SYSTEMS**” is the bonafide work done by me **G. GREESHMA SAI, REGD.NO-22G21F0029** in partial fulfillment of the requirements for the award of the degree of **Master of Computer Applications**, from Jawaharlal Nehru Technological University, Anantapuramu during the year 2022-2024.

Project Guide

Mrs. M. NARMADHA

Assistant Professor

Department of Master of Computer Applications
AUDISANKARA COLLEGE OF ENGG.&TECH.
GUDUR– TIRUPATI DISTRICT

Head of the Department

Prof. V. CHANDRASEKHAR

Associate Professor

Department of Master of computer Applications
AUDISANKARA COLLEGE OF ENGG.&TECH.
GUDUR– TIRUPATI DISTRICT

Submitted for the viva-voce examination held on _____

Internal Examiner

External Examiner

**AUDISANKARA COLLEGE OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)**

(Accredited by NAACA+)

Approved by AICTE, Affiliated to JNTUA,
Anantapuramu, Tirupati(DT), Andhra Pradesh.



DECLARATION

I, **MS. GREESHMA SAI, Regd.No-22G21F0029**, hereby declare that the Project Work entitle **"TRUSTWORTHINESS ASSESSMENT OF USERS IN SOCIAL REVIEWING SYSTEMS"** done by us under the esteemed guidance of Assistant Professor **Mrs. M. NARMADHA**, and is submitted in partial fulfilment of the requirements for the award of the **masters degree in computer applications**.

Date:

Place:

Signature of the Candidate

G. GREESHMA SAI
(22G21F0029)

ACKNOWLEDGEMENT

The satisfaction and elation that accompany the successful completion of any task would be incomplete without the mention of the people who have made it a possibility. It is my great privilege to express my gratitude and respect to all those who have guided me and inspired me during the course of this project work.

First and fore most, I express my since regratitude to our honorable chairman **Dr. VANKI PENCHALAIAH**, M.A., M.L., Ph.D., who provided all facilities and necessary encouragement during the course of study.

I extend my gratitude and sincere thanks to our beloved Director **Dr. A. MOHANBABU**, principal **Dr. J. RAJA MURUGADOSS** for motivating and providing necessary infrastructure and permitting me to complete the project.

I would like to express the sense of gratitude towards our Head of the Department **Prof.V. CHANDRASEKHAR**, my internal **Mrs. M. NARMADHA**, Assistant Professor and for their support and encouragement for completion of this project.

Finally, I express my sincere thanks to all the teaching and non-teaching staff that guided and helped me to complete the project work successfully.

ABSTRACT

Social Networks represent a cornerstone of our daily life, where the so-called social reviewing systems (SRSs) play a key role in our daily lives and are used to access data typically in the form of reviews. Due to their importance, social networks must be trustworthy and secure, so that their shared information can be used by the people without any concerns, and must be protected against possible attacks and misuses. One of the most critical attacks against the reputation system is represented by mendacious reviews. As this kind of attacks can be conducted by legitimate users of the network, a particularly powerful solution is to exploit trust management, by assigning a trust degree to users, so that people can weigh the gathered data based on such trust degrees. Trust management within the context of SRSs is particularly challenging, as determining incorrect behaviors is subjective and hard to be fully automatized. Several attempts in the current literature have been proposed; however, such an issue is still far from being completely resolved. In this study, we propose a solution against mendacious reviews that combines fuzzy logic and the theory of evidence by modeling trust management as a multicriteria multiexpert decision making and exploiting the novel concept of time-dependent and content-dependent crown consensus. We empirically proved that our approach outperforms the main related works approaches, also in dealing with sockpuppet attacks.

INDEX

S.NO	CONTENTS	PAGE NO
	ABSTRACT	
1	INTRODUCTION	1
2	LITERATURE SURVEY	5
2.1	UNDERSTANDING THE TRUSTWORTHINESS	5
3	SYSTEM ANALYSIS	7
3.1	EXISTING SYSTEM	7
3.2	PROPOSED SYSTEM	7
3.3	SOFTWARE AND HARDWARE REQUIREMENTS	10
4	SYSTEM DESING	11
4.1	SYSTEM ARCHITECTURE	20
4.2	INPUT & OUTPUT REPRESENTETION	20
4.3	DIAGRAMS	22
5	SYSTEM IMPLIMENTATION	32
6	SYSTEM TESTING	53
6.1	INTRODUCTION	53
6.2	TESTING STRATEGIES	53
6.3	TEST CASE	55
7	SCREENSHORTS	56
8	CONCLUSION	66
9	REFERENCE	67

1. INTRODUCTION

ABOUT MACHINE LEARNING

Machine learning ML is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than have human programmers specify every needed step.

The discipline of machine learning employs various approaches to help computers learn to accomplish tasks where no fully satisfactory algorithm is available. In cases where vast numbers of potential answers exist, one approach is to label some of the correct answers as valid. This can then be used as training data for the computer to improve the algorithm it uses to determine correct answers. For example, to train a system for the task of digital character recognition, the MNIST dataset has often been used.

MACHINE LEARNING APPROACHES

Early classifications for machine learning approaches sometimes divided them into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system. These were:

Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself discovering hidden patterns in data or a means towards an end feature learning.

Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal such as driving a vehicle or playing a game against an opponent. As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximise.

Other approaches or processes have since developed that don't fit neatly into this three-fold categorisation, and sometimes more than one is used by the same machine learning system. For example topic modeling, dimensionality reduction or meta learning. 8 As of 2020, deep learning has become the dominant approach for much ongoing work in the field of machine learning.

ABOUT THE PROJECT

As is well known, online social networks are applications that can be accessed via the Internet and are used by individuals to establish social relationships with other people who share personal interests and or activities. Aside from trading individual information, like photos or recordings, basically this multitude of utilizations permit their clients to impart remarks and insights on unambiguous points, to recommend items or spots of interest e.g., Excursion Guide, Foursquare, and so on. or to provide social settings that can help with specific task such as searching for a job on LinkedIn, answering research questions on Research Gate, making purchases on Amazon, etc.

Because of this remark assessment sharing, these social applications, which we will allude to as friendly surveying frameworks srss have been broadly utilized when individuals need to settle on everyday choices, expanding their fame. As a substantial model, a large portion of us admittance to an ideal SRS prior to picking a cafe or purchasing something to get surveys and input. Individuals are continuously and harmoniously subject to them as demonstrated by the high-level assessment displaying and examination, taking advantage of the effect of neighbours on client inclinations or moving toward the current data over-burden in SRS, for example. Hence, the dependability of SRS is especially significant, and a critical worry for powerful assessment elements and trust spread inside a local area of clients. As a matter of fact, srss experience the ill effects of manufactured messages and disguised counterfeit clients that can keep away from people take the ideal choice. This might raise a few issues about protection and security primarily because of the way that few individual and delicate data are shared, and spilled, all through SRS and that an individual might decide to conceal its actual self and goals behind an absolutely misleading virtual character or a Bot short for programming robots may emulate human conduct in SRS. Furthermore, dangers in SRS, for example, information spills, phishing snare, data altering, etc, are never restricted to a given social entertainer, however spread across the organization like a contamination by getting casualties among the companions of the plagued entertainers. Along these lines, a SRS supplier necessities to give appropriate security means to ensure its dependability. A few works in the ongoing writing, for example, generally manage fashioning messages as this can be handily settled by utilizing cryptography. Nonetheless, the second sort of malignant way of behaving brought about by covered counterfeit clients is as yet an open issue. During the last ten years, a few arrangements have been proposed to manage the issue of covered/counterfeit clients. The issue of giving security has prompted the reception of access control implies, while neutralizing producing hubs/personalities and social connections associations requested validation of clients and traded messages. For the most part, such components target moving toward outer aggressors or gate crashers, while upsetting genuine members in the SRS acting in a pernicious way is very difficult. A gullible method for safeguarding against vindictive people is to have clients being cautious while picking with whom to have a relationship.

Two clients in interpersonal organizations might have different sorts of connections: 1 in Face book-like frameworks clients can show others as "companions," or 2 in Instagram-like frameworks a client can "follow" others. Nonetheless, clients are commonly not so cautious while tolerating got joining demands, and choosing different clients to be associated with is normally very troublesome as malevolent clients are likewise specialists in disguising themselves. Not with standing the connections among the social entertainers inside a SRS ought to be founded on the immediate information in the genuine individuals behind such entertainers like previous cohorts, partners, or individual from similar family or gathering of companions, most of the connections are normally made without such a face-to-face information however among clients that have never been met face to face. One of the most common methods for combating such inside attackers is trust management. It comprises to dole out a "trust" worth to clients in light of the immediate examination of their ways of behaving or backhanded trust relationship among social entertainers. This is a soft secure measure that means breaking a social connection with actors who have a low trust level or making protections for actors who have a low trust level stronger by restricting their access to data and functions. Regardless of being a strong security implies trust the board isn't unequivocally given by the fundamental SRS stages, because of the issues connected with its programmed calculation.

OBJECTIVE OF PROJECT

The objective generally involves developing a comprehensive model to predict airfare prices by leveraging machine learning techniques.

2. LITERATURE SURVEY

USER TRUSTWORTHINESS IN ONLINE SOCIAL

The growing popularity of social networks and their easy acceptance of new users have the of fostering an environment where anonymous users can act in malicious ways. Although these platforms have many incentives to prevent such occurrences, they have not been able to cope with the sheer volume of information that must be processed. Moreover, the tendency of attackers to rapidly change strategies in response to defensive measures also poses a challenge. Hence, research on issues related to user trustworthiness on social networks is gaining traction, with many interesting conducted in recent years. In this work, we aim to review the present state of this field and present an analysis of the studies published between 2012 and 2020 that attempt to address this problem using various methodologies. Some of the solutions discussed in the literature can be described as bot identification protocols, while others focus on anti-spam protection, recognition of fake news, or rating the truthfulness of user-generated content. Many of these solutions offer in various respects, however none of them are able to provide comprehensive all-around protection against all possible types of attacks. Monitoring this scientific field is thus a key task, and this review will hopefully lead to a better understanding of the concept of online user trustworthiness by highlighting recent works that deal with this issue.

2.1 UNDERSTANDING THE TRUSTWORTHINESS

Management in the Social Internet of Things: A Survey Subhash Sagar, Adnan Mahmood, Quan Z. Sheng, Jitander Kumar Pabani, and Wei Emma Zhang
Abstract The next generation of the Internet of Things IoT facilitates the integration of the notion of social networking into smart objects i.e., things in a bid to establish the social network of interconnected objects. This integration has led to the evolution of a promising and emerging paradigm of the Social Internet of Things SIoT, wherein the smart objects act as social objects and intelligently impersonate the social behaviour similar to that of humans. These social objects are capable of establishing social relationships with the other objects in the network and can utilize these relationships for service discovery.

Trust plays a significant role to achieve the common goal of trustworthy collaboration and cooperation among the objects and provide systems' credibility and reliability. In SIoT, an untrustworthy object can disrupt the basic functionality of a service by delivering malicious messages and adversely affect the quality and reliability of the service. In this project, we present a holistic review of trustworthiness management for SIoT. The essence of trust in various disciplines has been discussed along with the Trust in SIoT followed by a detailed on trust management components in SIoT. Furthermore, we analyze and compare the trust management schemes by primarily categorizing them into four groups in terms of their strengths, limitations, trust management components employed in each of the referred trust management schemes, and the performance of these vis-a-vis numerous trust evaluation dimensions. Finally, we discuss the future research directions of the emerging paradigm of SIoT particularly for trustworthiness management in SIoT.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

- The existing system for trustworthiness assessment of users in social reviewing systems typically involves analyzing various factors such as user activity, review content, user profile information, and community feedback to determine the reliability and credibility of reviews.
- Analyzing the content of reviews for authenticity and relevance is another method. However, this can be challenging due to the subjective nature of reviews and the difficulty in distinguishing between genuine and fake content.
- Considering information such as user verification, profile completeness, and history can provide insights into user credibility.

DISADVANTAGES OF EXISTING SYSTEM

- Users can manipulate their activity or review content to appear trustworthy, leading to inaccurate assessments.
- New users or those with limited activity may be unfairly judged based on insufficient data.

3.2 PROPOSED SYSTEM

- The proposed system for Trustworthiness Assessment of Users in Social Reviewing Systems involves using machine learning and natural language processing techniques to evaluate the credibility and trustworthiness of user-generated reviews.
- The system can be integrated seamlessly into existing social reviewing platforms, enhancing their functionality without significant disruptions.
- Incorporating a feedback mechanism allows users to report suspicious reviews, which can further improve the accuracy and effectiveness of the trustworthiness assessment system over time.

ADVANTAGES OF PROPOSED SYSTEM

- Utilizing advanced machine learning algorithms enhances the accuracy of trustworthiness assessment, reducing false positives and negatives in identifying trustworthy and untrustworthy reviews.
- By accurately identifying untrustworthy reviews, the system helps reduce fraudulent activities such as fake reviews, spam, and biased content, improving the overall quality of reviews.

FEASIBILITY STUDY

TECHNICAL FEASIBILITY

The technical feasibility of "Trustworthiness assessment of users in social reviewing system" involves evaluating the ability to design and implement a system that accurately measures and assesses the trustworthiness of users based on their activities and behaviors within a social reviewing platform.

ECONOMIC FEASIBILITY

The economic feasibility of implementing a trustworthiness assessment in social reviewing systems can be effectively evaluated. This helps in making informed decisions regarding investments and expected returns.

OPERATIONAL FEASIBILITY

This system is being automated on the request of the technical department of our company. This new system meets their requirement and covers all aspects required much better than the old manual system. Most of the people involved in this branch are computer literate and do not need much training if this system is implemented. Hence it is operationally feasible.

MODULES

- Service Provider
- Remote User
- Train and Test
- Classification

SOFTWARE REQUIREMENT SPECIFICATION

FUNCTIONAL REQUIREMENTS

The main purpose of functional requirements within the requirement specification document is to define all the activities or operations that take place in the system. But the general functional requirements arrived at the end of the interaction with the users are listed below.

This project provides:

- Authentication of user whenever he/she logs in to the system.
- System shut down in case of a cyber-attack.
- A verification email is sent to user whenever he/she register for the first time on some software system.

NON-FUNCTIONAL REQUIREMENTS

The major non-functional Requirements of the system are as follows.

USABILITY

The system is designed with completely automated process hence there is no or less user intervention.

RELIABILITY

The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

PERFORMANCE

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

SUPPORTABILITY

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having JVM, built into the system.

IMPLEMENTATION

The system is implemented in web environment using struts framework. The apache tomcat is used as the web server and windows xp professional is used as the platform.

Interface the user interface is based on Struts provides HTML Tag.

3.3 SOFTWARE AND HARDWARE REQUIREMENTS

SOFTWARE REQUIREMENTS

- Operating system : Windows 10 64 bit
- Programming language : Python
- Front End : HTML, CSS
- Back End : MySQL 5.0
- Framework : Django
- Server : Xampp

HARDWARE REQUIREMENTS

- Processor : I3
- RAM : 4 GB
- Hard Disk : 500 GB

4. SYSTEM DESIGN

INTRODUCTION

The most creative and challenging phase of the life cycle is system and design. The term design describes a final system and the process by which it is developed. It refers to the technical specifications that will be applied in implementation of the candidate system. The design may be defined as “the process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient details to permit its physical realization”.

The design's goal is how the output is to be produced and in what format. Samples of the output and input are also presented. Second input data and database files have to be designed to meet the requirements of the proposed output. The processing phase is handled through the program construction and testing. Finally, details related to justification of the system and an estimate of the impact of the candidate system on the users and the organization are documented and evaluated by management as a step toward implementation.

The importance of software design can be stated in a single word “**Quality**”. Design provides us with representation of software that can be assessed for quality. Design is the only way that we can accurately translate a customer's requirements into a finished software product or system. Without design, we risk building an unstable system, that might fail if small changes are made or may be difficult to test, or one whose quality can't be tested. So it is an essential phase in the development of a software product.

SDLC METHODOLOGY

The document plays a vital role in the development of life cycle SDLC as it describes the complete requirement of the system. It means for use by developers and will be the basis during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, a spiral model of Software Development and Enhancement.

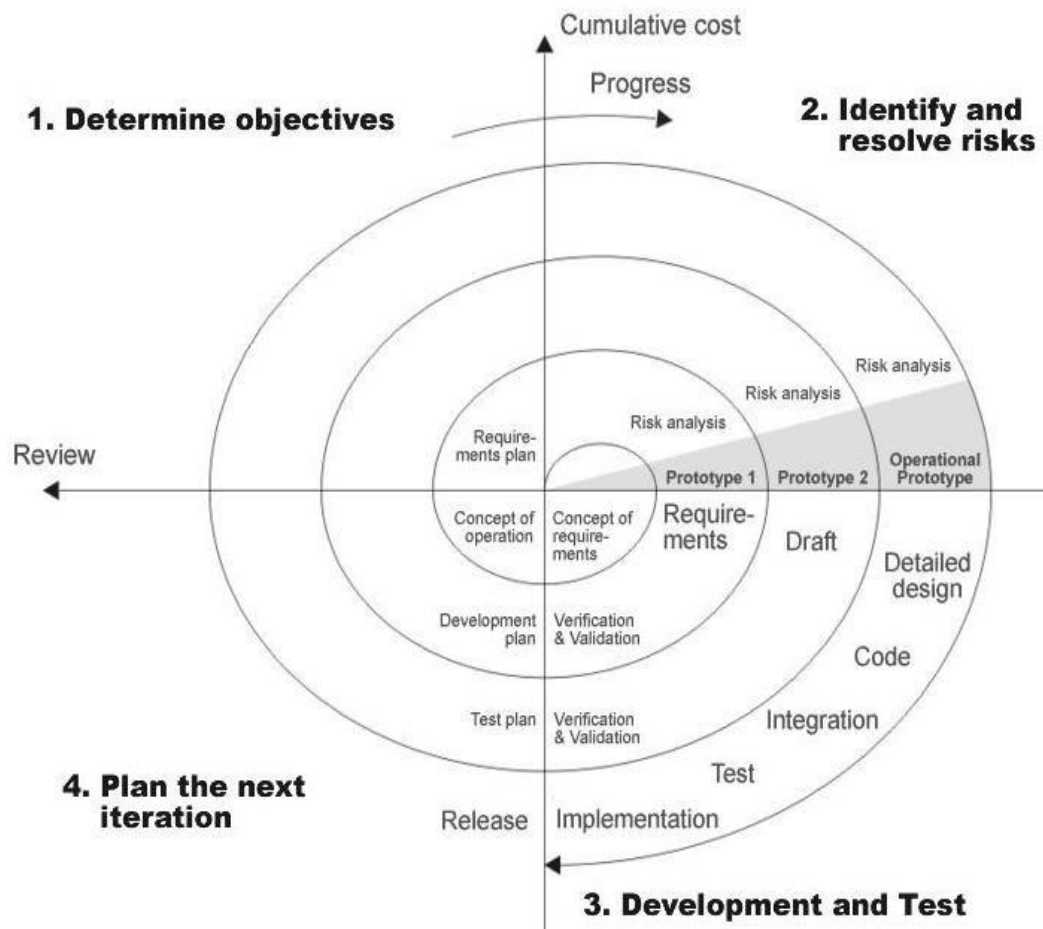


Fig: SDLC Methodology

STAGES IN SDLC

- Requirement Gathering
- Analysis
- Designing
- Coding
- Testing
- Maintenance

REQUIREMENTS GATHERINGSTAGE

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.

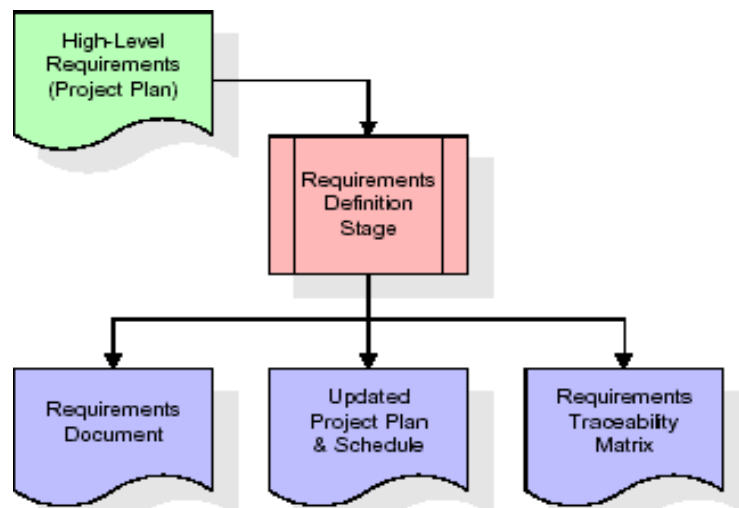


Fig: Requirements Gathering Stage

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.

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

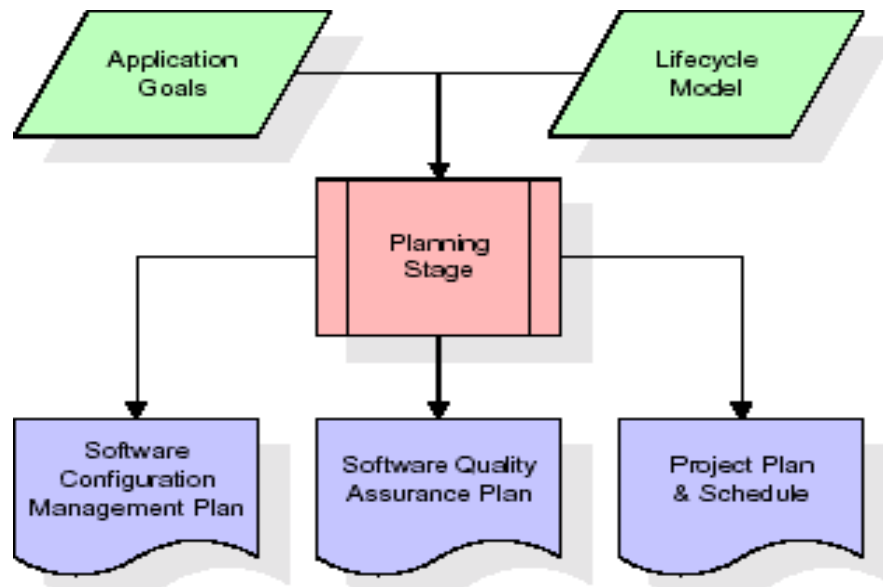


Fig: Analysis Stage

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.

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.

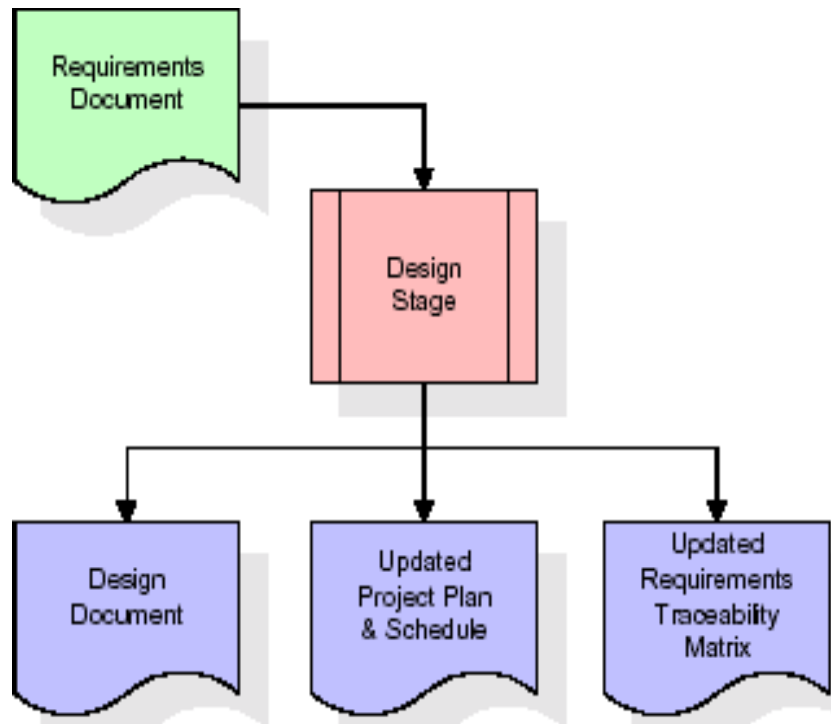


Fig: Analysis Stage2

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.

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, 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.

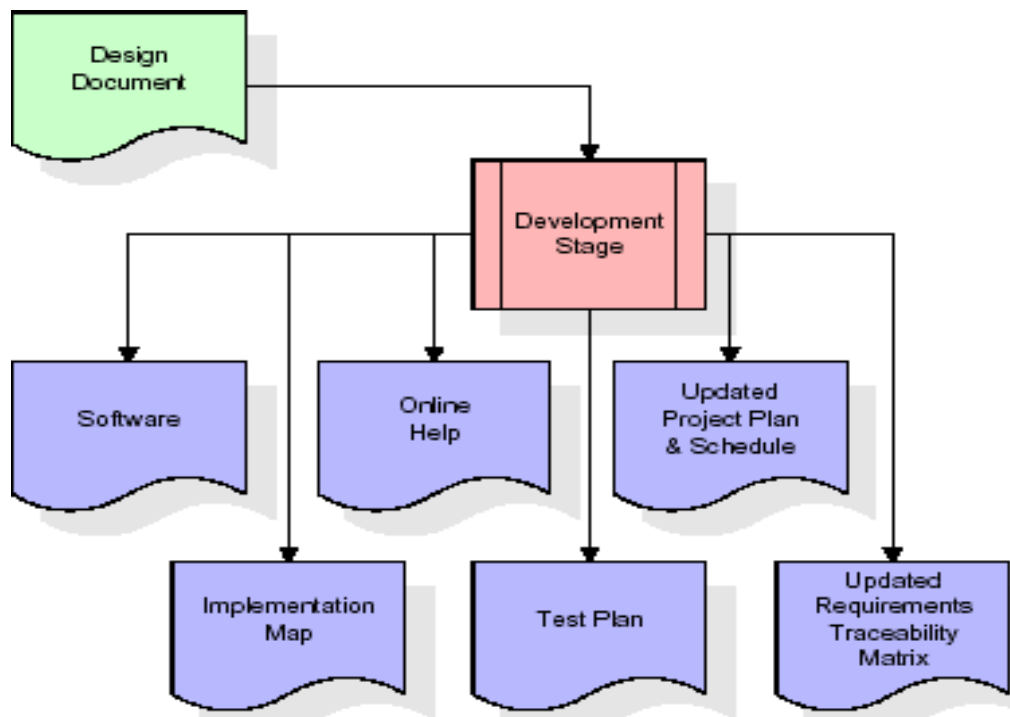


Fig: Analysis Stage3

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.

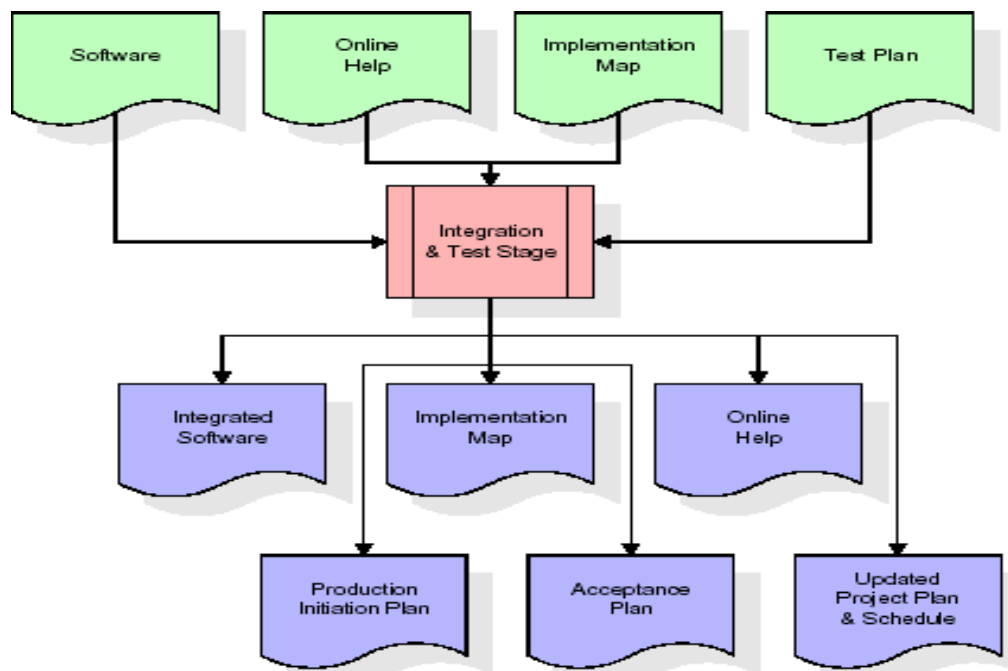


Fig: Integration Test Stage

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 artifacts, 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.

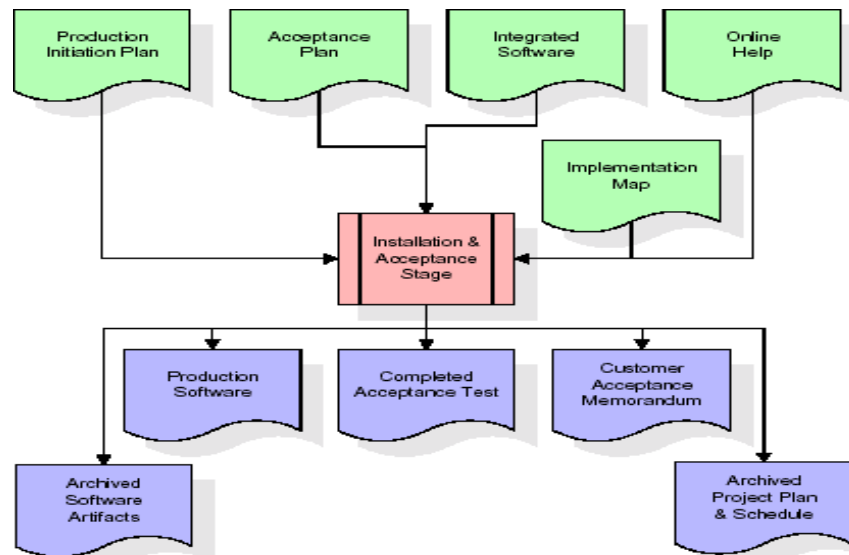


Fig: Installation Acceptance Test

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.

4.1 SYSTEM ARCHITECTURE

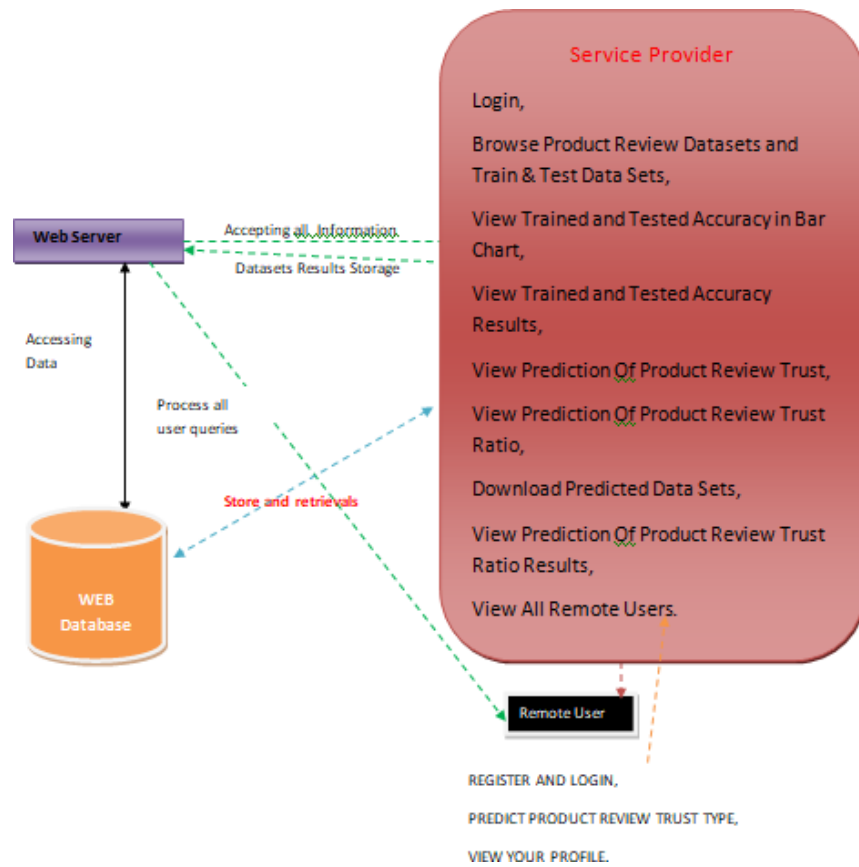


Fig: System Overview

4.2 INPUT & OUTPUT REPRESENTATION

INPUT DESIGN

Input design includes data mediums used for inputting data and validations that are to be done during data entry. Different messages regarding data are given to guide users during data entry. Validation checks are done for each input.

Data entry screens are designed so that the system interacts with the user in providing an effective dialogue. Fields in the screen are logically arranged to help the user.

The design is the process of converting the user-originated inputs into a compute-based format. The goal of the input design is to make the data entry easier, logical and free from error. Errors in the input data are controlled by input design.

The application has been developed in a user-friendly manner. The windows have been designed in such a way that during the processing the cursor is placed in the position where the data must be entered. If any of the data going into the system is wrong then the process and output will magnify these errors.

The decisions made during design of input are:

- 1) To achieve the highest possible level of accuracy.
- 2) To provide a list of possible choices and help while accepting the input for an important field wherever possible outputs from computer system are required primarily to communicate the results of processing to the users. They are also used to provide a permanent copy of these results for later consultation/verification.

OUTPUT DESIGN

Output refers to the results and information that are generated by the system. Output is the main reason for developing the system and based on this, the usefulness and applicability of system are evaluated.

Outputs from computer systems are required primarily to communicate the results of processing to users. Efficiently designed outputs enhance the understandability of the information.

According to the requirements of the system, various types of outputs are considered and designed as follows. Internal outputs, whose destination is within the organization and which require careful design because they are the user's main interface with the computer. Interactive outputs, in which the user communication with the Computer is essential.

4.3 DIAGRAMS

The overall logical structure of a database can be expressed graphically by an E-R diagram. The relative simplicity and pictorial clarity of this diagramming technique may well account in large part for the widespread use of the E-R model. Such a diagram consists of the following major components.

Rectangles: Represent Entity Sets.

Ellipses: Represent attributes.

Diamonds: Represent relationship sets.

Lines: Link attributes to entity sets and entity sets.

DATA FLOW DIAGRAM

A graphical tool used to describe and analyse the movement of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed.

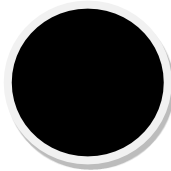
The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart. DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built.

Later during design activity this is taken as the basis for drawing the system's structure charts. The Basic Notation used to create a DFD's are as follows:

1. Dataflow: Data move in a specific direction from an origin to a destination.



2. Process: People, procedures, or devices that use or produce Transform Data.
The physical component is not identified.



3. Source: External sources or destination of data, which may be People, programs, organizations or other entities.



4. Data Store: Here data are stored or referenced by a process in the System.



UML DIAGRAMS

Unified Modeling Language is a language available to perform modeling of software. A model is simplification of reality. A model provides the blue print of the system, model encompasses detailed plans.

BUILDING BLOCKS OF THE UML

The vocabulary of the UML encompasses three kinds of building blocks.

1. Things.
2. Relationships.
3. Diagrams.

THINGS IN THE UML

Things are the abstractions that are first-class citizen in a model. There are four kinds of things in the UML.

- Structure things.
- Behavioral things.
- Grouping things.
- Annotational things.

These things are the basic object-oriented building blocks of the UML. You use them to write well-formed models.

RELATIONSHIPS IN THE UML

Things can be connected to logically or physically with the help of relationship in object oriented modeling. These are four kinds of relationships in the UML.

- Dependency.
- Association.
- Generalization.
- Realization.

DIAGRAMS IN THE UML

A diagram is a graphical representation of a set of elements. There are nine kinds of diagrams in the UML.

- Class diagram.
- Object diagram.
- Use case diagram.
- Sequence diagram.
- Collaboration diagram.
- Activity diagram.
- Component diagram.
- State chart diagram.
- Deployment diagram.

USECASE DIAGRAM

Use case diagram shows a set of usecases and actors a special kind of class and their relationship. Usecase diagrams address the static usecase view of a system. These diagrams are especially important in organizing and modeling the behavioural of a system both sequence and collaboration diagrams are kind of interaction diagram.

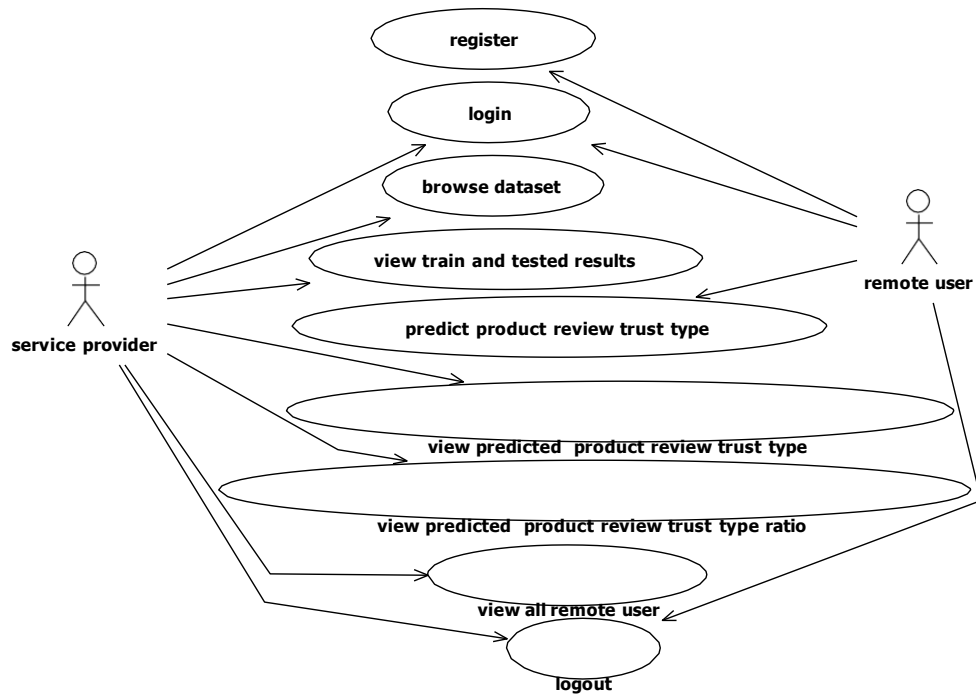


Fig: Use-case Diagram

CLASS DIAGRAM

Class diagrams are one of the foremost common diagrams employed in UML. A class diagram consists of classes, interfaces, associations and collaborations. Class diagrams primarily represent the static structure of a system that is static in nature. An active class diagram is employed in a very class diagram to represent the concurrency of the system.

A class diagram represents the static structure of a system. Therefore it's usually used for development purposes. This can be the foremost widely used diagram at the time of system construction.

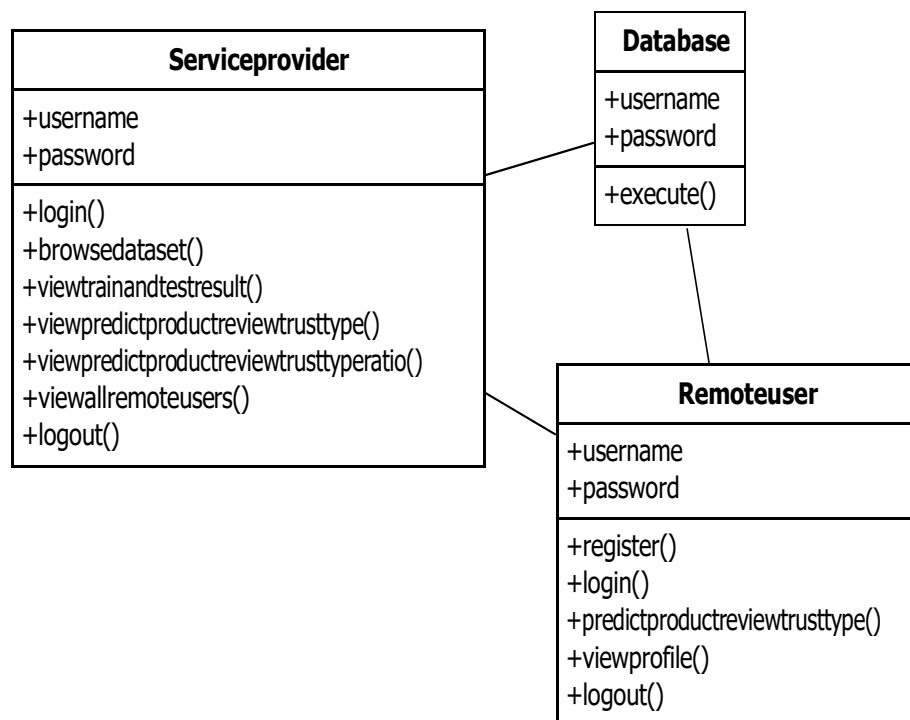


Fig: Class Diagram

SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling 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.

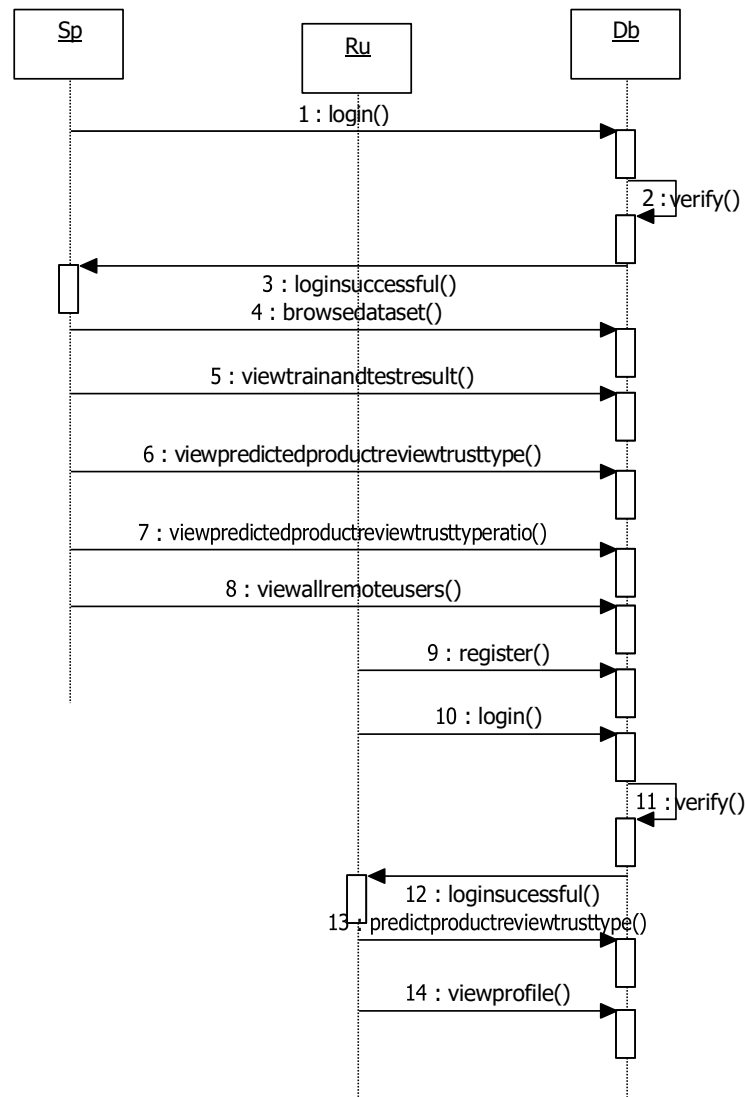


Fig: Sequence Diagram

ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

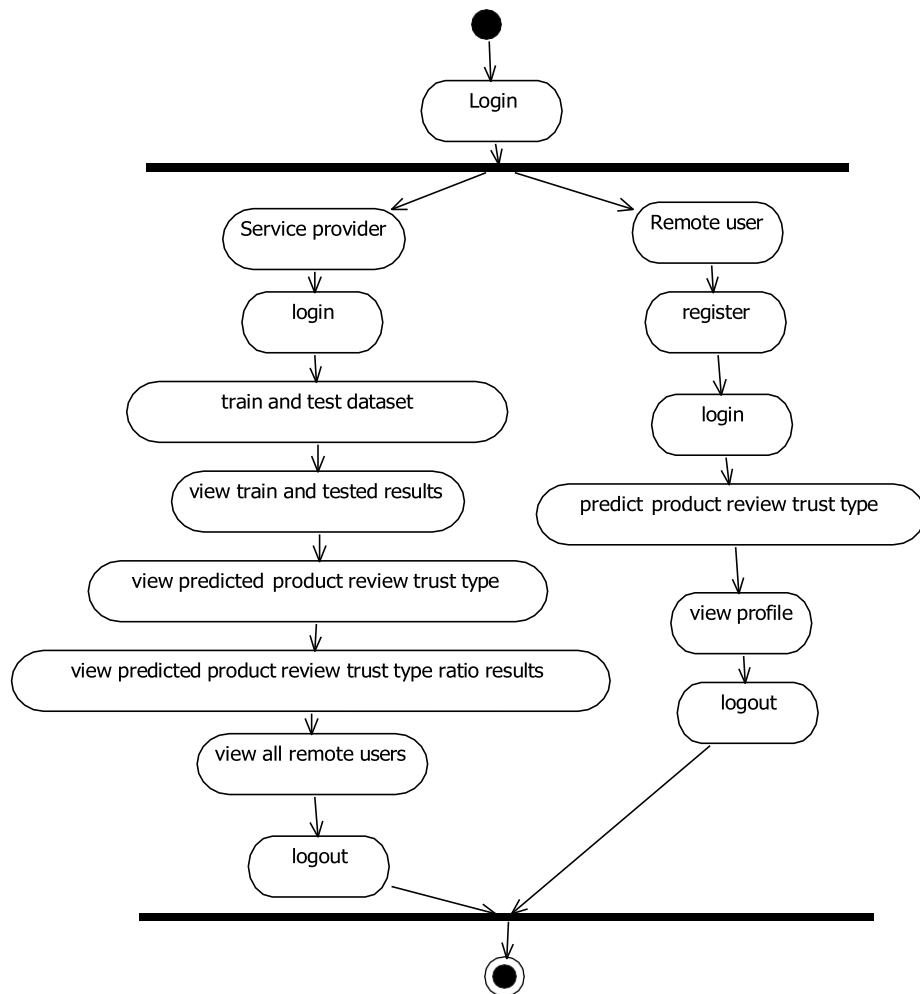


Fig: Activity Diagram

MODULE DESCRIPTION

SERVICE PROVIDER

- In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Browse the dataset, view trained and tested results, view predicted product review trust type, view predicted product review trust type ratio and view all remote users.

TRAIN AND TEST MODEL

- In this module, the service provider split the Used dataset into train and test data of ratio 70 % and 30 % respectively. The 70% of the data is consider as train data which is used to train the model and 30% of the data is consider as test which is used to test the model.

REMOTE USER

- In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like predict product review trust type, View Your Profile.

CLASSIFICATION

- In this module, user enter data and classify them using tested machine learning models.

ALGORITHM

SUPPORT VECTOR MACHINE

- In machine learning, support-vector machines SVMs, also support-vector networks are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. An SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

GRADIENT BOOSTING CLASSIFIER

- Gradient Boosting is a powerful boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error or cross-entropy of the previous model using gradient descent. In each iteration, the algorithm computes the gradient of the loss function with respect to the predictions of the current ensemble and then trains a new weak model to minimize this gradient. The predictions of the new model are then added to the ensemble, and the process is repeated until a stopping criterion is met.

DECISION TREE

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.
- It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

LOGISTIC REGRESSION

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

KNN

- KNN is one of the most basic yet essential classification algorithms in machine learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining, and intrusion detection.
- It is widely disposable in real-life scenarios since it is non-parametric, meaning it does not make any underlying assumptions about the distribution of data as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data. We are given some prior data also called training data, which classifies coordinates into groups identified by an attribute.

5. SYSTEM IMPLEMENTATION

INTRODUCTION

This mapping permits forward engineering: The generation of code from a UML model into a programming language. The reverse is also possible: You can reconstruct a model from an implementation back into the UML. Reverse engineering is not magic. Unless you encode that information in the implementation, information is lost when moving forward from models to code. Reverse engineering thus requires tool support with human intervention.

TECHNOLOGY DESCRIPTION

INTRODUCTION TO PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured particularly, procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3. The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 first planned for 2015 after which security patches and other improvements will not be released for it. 32 33 With Python 2's end-of-life, only Python 3.5.x and later are supported. Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source 35 reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

SYNTAX AND SEMANTICS

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation.

Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

INDENTATION

Main article: [Python syntax and semantics and Indentation](#).

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is sometimes termed the off-side rule, which some other languages share, but in most languages indentation doesn't have any semantic meaning.

STATEMENTS AND CONTROL FLOW

Python's statements include among others:

The assignment statement token '=', the equals sign. This operates differently than in traditional imperative programming languages, and this fundamental mechanism including the nature of Python's version of variables illuminates many other features of the language. Assignment in C, e.g., `x = 2`, translates to "typed variable name x receives a copy of numeric value 2". The right-hand value is copied into an allocated storage location for which the left-hand variable name is the symbolic address. The memory allocated to the variable is large enough potentially quite large for the declared type. In the simplest case of Python assignment, using the same example, `x = 2`, translates to "generic name x receives a reference to a separate, dynamically allocated object of numeric int type of value 2." This is termed binding the name to the object.

Since the name's storage location doesn't contain the indicated value, it is improper to call it a variable.

Names may be subsequently rebound at any time to objects of greatly varying types, including strings, procedures, complex objects with data and methods, etc. Successive assignments of a common value to multiple names, e.g., `x = 2; y = 2; z = 2` result in allocating storage to at most three names and one numeric object, to which all three names are bound.

Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However at a given time a name will be bound to some object, which will have a type; thus there is dynamic typing.

- The if statement, which conditionally executes a block of code, along with else and elif a contraction of else-if.
- The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
- The while statement, which executes a block of code as long as its condition is true.
- The try statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.
- The raise statement, used to raise a specified exception or re-raise a caught exception.
- The class statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.
- The def statement, which defines a function or method.
- The with statement, from Python 2.5 released in September 2006, which encloses a code block within a context manager for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it, allowing Resource Acquisition Is Initialization RAII-like behavior and replaces a common try/finally idiom.
- The break statement, exits from the loop.
- The continue statement, skips this iteration and continues with the next item.

- The pass statement, which serves as a NOP. It is syntactically needed to create an empty code block.
- The assert statement, used during debugging to check for conditions that ought to apply.
- The yield statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement coroutines.

The import statement, which is used to import modules whose functions or variables can be used in the current program. There are three ways of using import: `import <module name> [as <alias>]` or `from <module name> import *` or `from <module name> import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>]`, The print statement was changed to the `print()` function in Python 3.

Python does not support tail call optimization or first-class continuations, and, according to Guido van Rossum, it never will. However, better support for coroutine-like functionality is provided in 2.5, by extending Python's generators. Before 2.5, generators were lazy iterators; information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through multiple stack levels.

EXPRESSIONS

Some Python expressions are similar to languages such as C and Java, while some are not:

Addition, subtraction, and multiplication are the same, but the behavior of division differs. There are two types of divisions in Python. They are floor division or integer division `//` and floating point/division. Python also added the `**` operator for exponentiation.

From Python 3.5, the new `@` infix operator was introduced. It is intended to be used by libraries such as NumPy for matrix multiplication.

From Python 3.8, the syntax `:=`, called the 'walrus operator' was introduced. It assigns values to variables as part of a larger expression.

In Python, `==` compares by value, versus Java, which compares numerics by value and objects by reference. Value comparisons in Java on objects can be performed with the `equals()` method. Python's `is` operator may be used to compare object identities comparison by reference. In Python, comparisons may be chained, for example `a <= b <= c`.

Python uses the words `and`, `or`, `not` for its boolean operators rather than the symbolic `&&`, `||`, `!` used in Java and C.

Python has a type of expression termed a list comprehension. Python 2.4 extended list comprehensions into a more general expression termed a generator expression. Anonymous functions are implemented using lambda expressions; however, these are limited in that the body can only be one expression.

Conditional expressions in Python are written as `x if c else y` different in order of operands from the `c ? x : y` operator common to many other languages.

Python makes a distinction between lists and tuples. Lists are written as `[1, 2, 3]`, are mutable, and cannot be used as the keys of dictionaries dictionary keys must be immutable in Python. Tuples are written as `(1, 2, 3)` are immutable and thus can be used as the keys of dictionaries, provided all elements of the tuple are immutable. The `+` operator can be used to concatenate two tuples, which does not directly modify their contents, but rather produces a new tuple containing the elements of both provided tuples. Thus, given the variable `t` initially equal to `(1, 2, 3)` executing `t = t + (4, 5)` first evaluates `t + (4, 5)` which yields `(1, 2, 3, 4, 5)` which is then assigned back to `t`, thereby effectively "modifying the contents" of `t`, while conforming to the immutable nature of tuple objects. Parentheses are optional for tuples in unambiguous contexts.

Python features sequence unpacking wherein multiple expressions, each evaluating to anything that can be assigned to a variable, a writable property, etc, are associated in the identical manner to that forming tuple literals and, as a whole, are put on the left hand side of the equal sign in an assignment statement.

The statement expects an iterable object on the right hand side of the equal sign that produces the same number of values as the provided writable expressions when iterated through, and will iterate through it, assigning each of the produced values to the corresponding expression on the left.

Python has a "string format" operator `%`. This functions analogous to `printf` format strings in C, e.g. `"spam=%s eggs=%d" % "blah", 2` evaluates to `"spam=blah eggs=2"`.

In Python 3 and 2.6+, this was supplemented by the `format()` method of the `str` class, e.g. `"spam={0} eggs={1}".format("blah", 2)`. Python 3.6 added "f-strings": `blah = "blah"; eggs = 2; f'spam={blah} eggs={eggs}'`.

PYTHON HAS VARIOUS KINDS OF STRING LITERALS:

Strings delimited by single or double quote marks. Unlike in Unix shells, Perl and Perl-influenced languages, single quote marks and double quote marks function identically. Both kinds of string use the backslash (`\`) as an escape character. String interpolation became available in Python 3.6 as "formatted string literals".

Triple-quoted strings, which begin and end with a series of three single or double quote marks. They may span multiple lines and function like here documents in shells, Perl and Ruby.

Raw string varieties, denoted by prefixing the string literal with an `r`. Escape sequences are not interpreted; hence raw strings are useful where literal backslashes are common, such as regular expressions and Windows-style paths. Compare "@-quoting" in C#.

Python has array index and array slicing expressions on lists, denoted as `a[key]`, `a[start:stop]` or `a[start:stop:step]`. Indexes are zero-based, and negative indexes are relative to the end. Slices take elements from the start index up to, but not including, the stop index. The third slice parameter, called step or stride, allows elements to be skipped and reversed. Slice indexes may be omitted, for example `a[:]` returns a copy of the entire list. Each element of a slice is a shallow copy.

In Python, a distinction between expressions and statements is rigidly enforced, in contrast to languages such as Common Lisp, Scheme, or Ruby. This leads to duplicating some functionality.

For example:

- List comprehensions vs. for-loops.
- Conditional expressions vs. if blocks.

The `eval()` vs. `exec()` built-in functions in Python 2, `exec` is a statement; the former is for expressions, the latter is for statements.

Statements cannot be a part of an expression, so list and other comprehensions or lambda expressions, all being expressions, cannot contain statements. A particular case of this is that an assignment statement such as `a = 1` cannot form part of the conditional expression of a conditional statement. This has the advantage of avoiding a classic C error of mistaking an assignment operator `=` for an equality operator `==` in conditions: `if c = 1 { ... }` is syntactically valid but probably unintended C code but `if c = 1: ...` causes a syntax error in Python.

METHODS

Methods on objects are functions attached to the object's class; the syntax instance. Method argument is, for normal methods and functions, syntactic sugar for `Class. Method instance, argument`. Python methods have an explicit `self` parameter to access instance data, in contrast to the implicit `self` or `this` in some other object- oriented programming languages e.g., C++, Java, Objective-C, or Ruby.

APPLICATIONS OF PYTHON

As mentioned before, Python is one of the most widely used language over the web. I'm going to list few of them here:

Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read: Python code is more clearly defined and visible to the eyes.

Easy-to-maintain: Python's source code is fairly easy-to-maintain.

A broad standard library: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

Databases: Python provides interfaces to all major commercial databases.

GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable: Python provides a better structure and support for large programs than shell scripting.

INSTALLATION STEPS OF PYTHON

Installing and using Python on Windows 10 is very simple. The installation procedure involves just three steps:

- Download the binaries.
- Run the Executable installer.
- Add Python to PATH environmental variables.

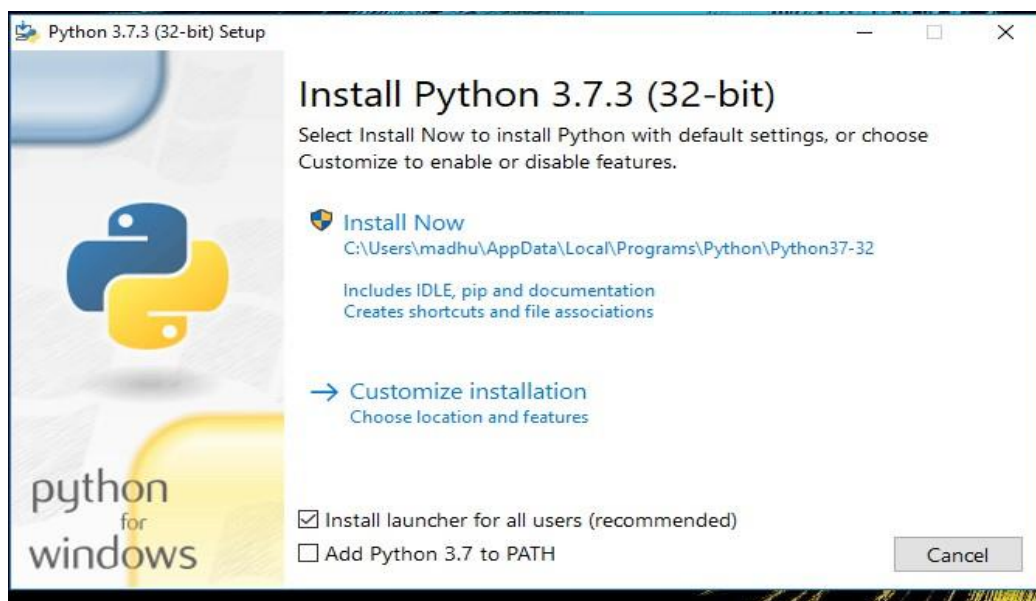
To install Python, you need to download the official Python executable installer. Next, you need to run this installer and complete the installation steps. Finally, you can configure the PATH variable to use python from the command line.

Step 1: Download the Python Installer binaries.

- Open the official Python website in your web browser. Navigate to the Downloads tab for Windows.
- Choose the latest Python 3 release. In our example, we choose the latest Python 3.7.3 version. Click on the link to download Windows x86 executable installer if you are using a 32-bit installer.
- In case your Windows installation is a 64-bit system, then download Windows x86-64 executable installer.

Step 2: Run the Executable Installer.

1. Once the installer is downloaded, run the Python installer.
2. Check the Install launcher for all users check box. Further, you may check the Add Python 3.7 to path check box to include the interpreter in the execution path.

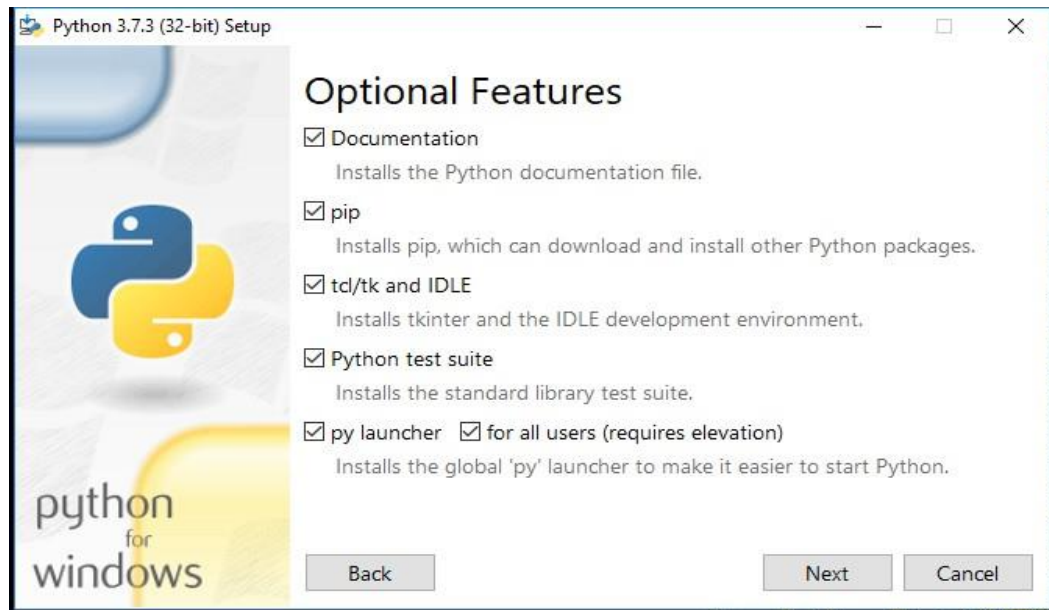


3. Select Customize installation.

Choose the optional features by checking the following check boxes:

1. Documentation
2. pip
3. tcl/tk and IDLE to install tkinter and IDLE
4. Python test suite to install the standard library test suite of Python

5. Install the global launcher for `.py` files. This makes it easier to start Python
6. Install for all users.

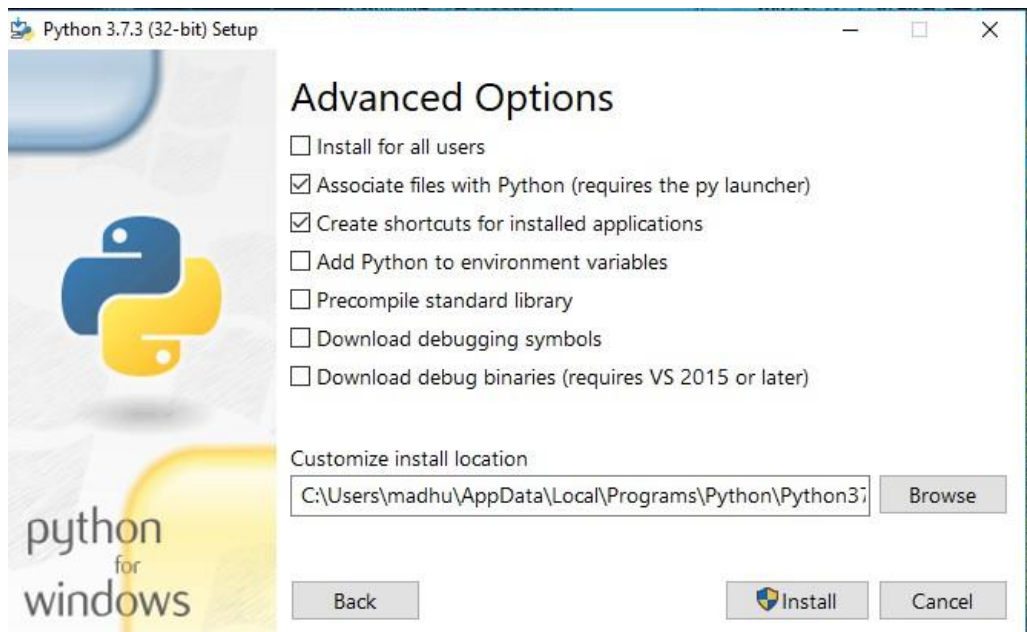


Click Next.

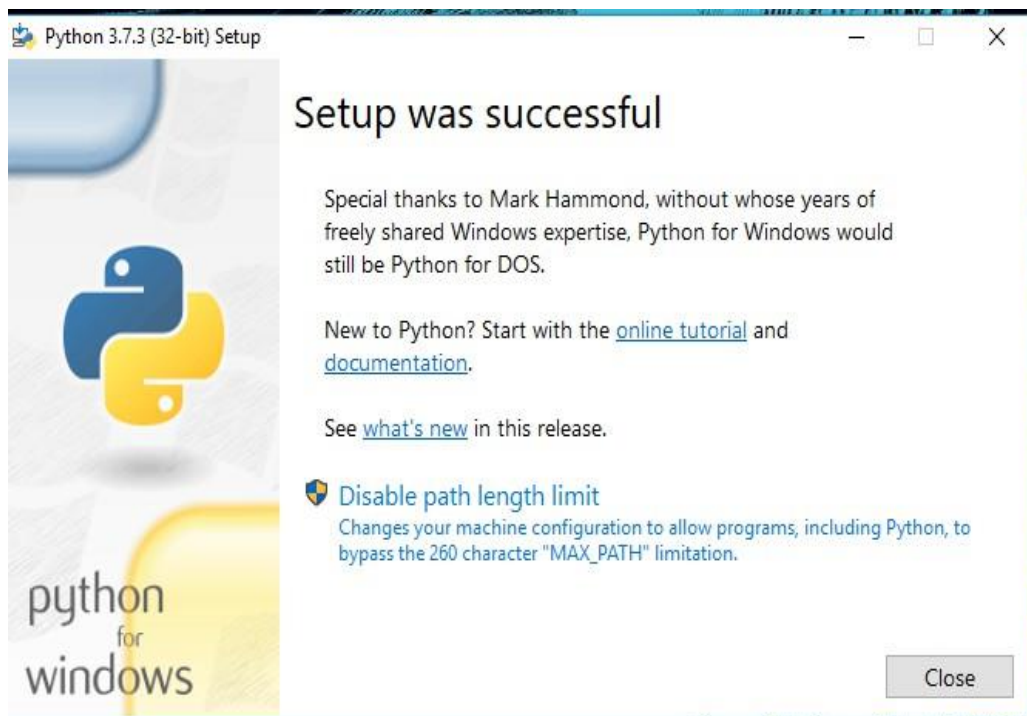
4. This takes you to Advanced Options available while installing Python. Here, select the Install for all users and Add Python to environment variables check boxes.

Optionally, you can select the Associate files with Python, Create shortcuts for installed applications and other advanced options. Make note of the python installation directory displayed in this step. You would need it for the next step.

After selecting the Advanced options, click Install to start installation.



5. Once the installation is over, you will see a Python Setup Successful window.



Step 3: Add Python to environmental variables.

The last optional step in the installation process is to add Python Path to the System Environment variables. This step is done to access Python through the command line.

In case you have added Python to environment variables while setting the Advanced options during the installation procedure, you can avoid this step. Else, this step is done manually as follows.

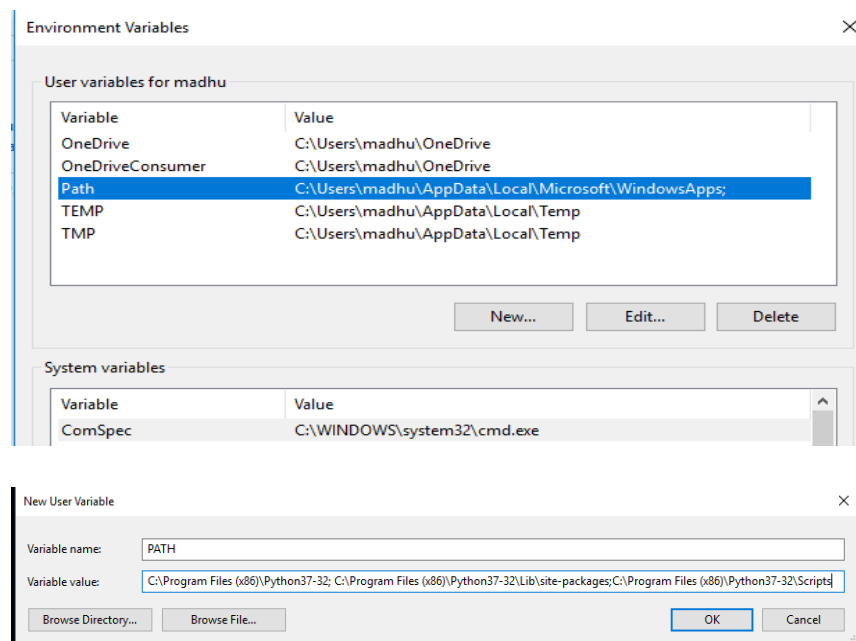
In the Start menu, search for “advanced system settings”. Select “View advanced system settings”. In the “System Properties” window, click on the “Advanced” tab and then click on the “Environment Variables” button.

Locate the Python installation directory on your system. If you followed the steps exactly as above, python will be installed in below locations:

- C:\Program Files (x86)\Python37-32: for 32-bit installation.
- C:\Program Files\Python37-32: for 64-bit installation.

The folder name may be different from “Python37-32” if you installed a different version. Look for a folder whose name starts with Python.

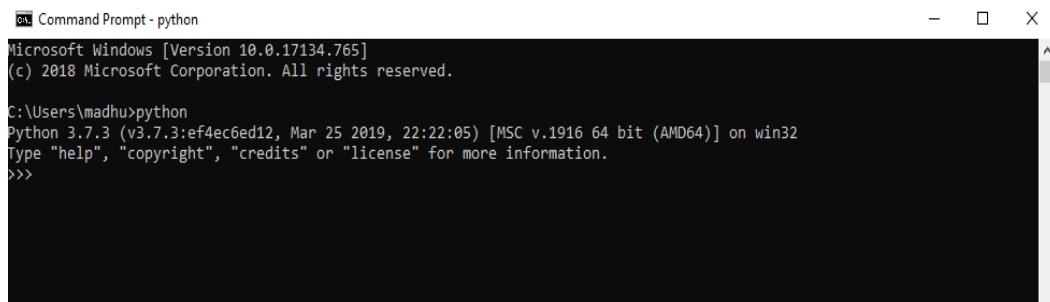
Append the following entries to PATH variable as shown below:



Step 4: Verify the Python Installation.

You have now successfully installed Python 3.7.3 on Windows 10. You can verify if the Python installation is successful either through the command line or through the IDLE app that gets installed along with the installation.

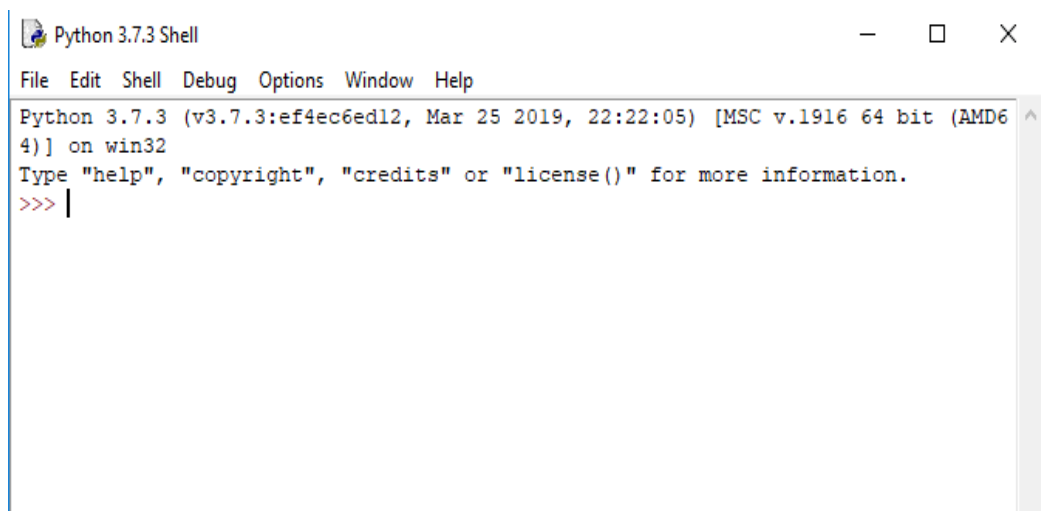
Search for the command prompt and type “python”. You can see that Python 3.7.3 is successfully installed.



```
Command Prompt - python
Microsoft Windows [Version 10.0.17134.765]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\madhu>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

An alternate way to reach python is to search for “Python” in the start menu and clicking on IDLE Python 3.7 64-bit. You can start coding in Python using the Integrated Development Environment IDLE.



```
Python 3.7.3 Shell
File Edit Shell Debug Options Window Help
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

USES

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of February 2020, it is the third most popular language behind Java, and C. It was selected Programming Language of the Year in 2007, 2010, and 2018.

- An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming problems involving string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".

- Large organizations that use Python include Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify and some smaller entities like ILM and ITA. The social news networking site Reddit is written entirely in Python.
- Python can serve as a scripting language for web applications, e.g., via mod wsgi for the Apache web server. With Web Server Gateway Interface, a standard API has evolved to facilitate these applications. Web frameworks like Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and IronPython can be used to develop the client-side of Ajax-based applications.
- SQLAlchemy can be used as data mapper to a relational database. Twisted is a framework to program communications between computers, and is used for example by Dropbox.
- Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing, with specialized libraries such as Biopython and Astropy providing domain-specific functionality. SageMath is a mathematical software with a notebook interface programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus.
- Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like Free CAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, MotionBuilder, Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella. GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go.

- Python is commonly used in artificial intelligence projects with the help of libraries like TensorFlow, Keras, Pytorch and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.
- Many operating systems include Python as a standard component. It ships with most Linux distributions, AmigaOS 4, FreeBSD as a package, NetBSD, OpenBSD as a package and macOS and can be used from the command line terminal. Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.
- Python is used extensively in the information security industry, including in exploit development.
- Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. The Raspberry Pi single-board computer project has adopted Python as its main user-programming language.
- Due to Python's user-friendly conventions and easy-to-understand language, it is commonly used as an intro language into computing sciences with students. This allows students to easily learn computing theories and concepts and then apply them to other programming languages.
- Libre Office includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature 169 since Version 4.0 from 7 February 2013.
- Python is a general-purpose, popular programming language, and it is used in almost every technical field. The various areas of Python use are given below.
- Data Science: Data Science is a vast field, and Python is an important language for this field because of its simplicity, ease of use, and availability of powerful data analysis and visualization libraries like NumPy, Pandas, and Matplotlib.
- Desktop Applications: PyQt and Tkinter are useful libraries that can be used in GUI - Graphical User Interface-based Desktop Applications. There are better languages for this field, but it can be used with other languages for making Applications.

- **Console-based Applications:** Python is also commonly used to create command-line or console-based applications because of its ease of use and support for advanced features such as input/output redirection and piping.
- **Mobile Applications:** While Python is not commonly used for creating mobile applications, it can still be combined with frameworks like Kivy or BeeWare to create cross-platform mobile applications.
- **Software Development:** Python is considered one of the best software-making languages. Python is easily compatible with both from Small Scale to Large Scale software.
- **Artificial Intelligence:** AI is an emerging Technology, and Python is a perfect language for artificial intelligence and machine learning because of the availability of powerful libraries such as TensorFlow, Keras, and PyTorch.
- **Web Applications:** Python is commonly used in web development on the backend with frameworks like Django and Flask and on the front end with tools like JavaScript HTML and CSS.
- **Enterprise Applications:** Python can be used to develop large-scale enterprise applications with features such as distributed computing, networking, and parallel processing.
- **3D CAD Applications:** Python can be used for 3D computer-aided design CAD applications through libraries such as Blender.
- **Machine Learning:** Python is widely used for machine learning due to its simplicity, ease of use, and availability of powerful machine learning libraries.
- **Computer Vision or Image Processing Applications:** Python can be used for computer vision and image processing applications through powerful libraries such as OpenCV and Scikit-image.
- **Speech Recognition:** Python can be used for speech recognition applications through libraries such as Speech Recognition and PyAudio.
- **Scientific computing:** Libraries like NumPy, SciPy, and Pandas provide advanced numerical computing capabilities for tasks like data analysis, machine learning, and more.
- **Education:** Python's easy-to-learn syntax and availability of many resources make it an ideal language for teaching programming to beginners.

- Testing: Python is used for writing automated tests, providing frameworks like unit tests and pytest that help write test cases and generate reports.
- Gaming: Python has libraries like Pygame, which provide a platform for developing games using Python.
- IoT: Python is used in IoT for developing scripts and applications for devices like Raspberry Pi, Arduino, and others.
- Networking: Python is used in networking for developing scripts and applications for network automation, monitoring, and management.
- DevOps: Python is widely used in DevOps for automation and scripting of infrastructure management, configuration management, and deployment processes.
- Finance: Python has libraries like Pandas, Scikit-learn, and Statsmodels for financial modeling and analysis.
- Audio and Music: Python has libraries like Pyaudio, which is used for audio processing, synthesis, and analysis, and Music21, which is used for music analysis and generation.
- Writing scripts: Python is used for writing utility scripts to automate tasks like file operations, web scraping, and data processing. Python has wide range of libraries and frameworks widely used in various fields such as machine learning, artificial intelligence, web applications, etc. We define some popular frameworks and libraries of Python as follows.
- Web development Server-side: Django Flask, Pyramid, CherryPy.
- GUIs based applications: Tkinter, PyGTK, PyQt, PyJs, etc.
- Machine Learning: TensorFlow, PyTorch, Scikit-learn, Matplotlib, Scipy, etc.
- Mathematics: NumPy, Pandas, etc.
- BeautifulSoup: a library for web scraping and parsing HTML and XML.
- Requests: a library for making HTTP requests.
- SQLAlchemy: a library for working with SQL databases.
- Kivy: a framework for building multi-touch applications.
- Pygame: a library for game development.
- Pytest: a testing framework for Python Django.
- REST framework: a toolkit for building restful APIs.
- FastAPI: a modern, fast web framework for building APIs.

- Streamlit: a library for building interactive web apps for machine learning and data science.
- NLTK: a library for natural language processing.

PYTHON FUNCTIONS

- **Lambda Function:** A lambda function is a small, anonymous function that can take any number of arguments but can only have one expression. Lambda functions are often used in functional programming to create functions "on the fly" without defining a named function.
- **Recursive Function:** A recursive function is a function that calls itself to solve a problem. Recursive functions are often used in functional programming to perform complex computations or to traverse complex data structures.
- **Map Function:** The `map()` function applies a given function to each item of an iterable and returns a new iterable with the results. The input iterable can be a list, tuple, or other.
- **Filter Function:** The `filter()` function returns an iterator from an iterable for which the function passed as the first argument returns True. It filters out the items from an iterable that do not meet the given condition.
- **Reduce Function:** The `reduce()` function applies a function of two arguments cumulatively to the items of an iterable from left to right to reduce it to a single value.
- **functools Module:** The `functools` module in Python provides higher-order functions that operate on other functions, such as `partial()` and `reduce()`.
- **Currying Function:** A currying function is a function that takes multiple arguments and returns a sequence of functions that each take a single argument.
- **Memoization Function:** Memoization is a technique used in functional programming to cache the results of expensive function calls and return the cached Result when the same inputs occur again.
- **Threading Function:** Threading is a technique used in functional programming to run multiple tasks simultaneously to make the code more efficient and faster.

PYTHON MODULES

- **Math:** Gives users access to mathematical constants and pi and trigonometric functions.
- **Datetime:** Provides classes for a simpler way of manipulating dates, times, and periods.
- **OS:** Enables interaction with the base operating system, including administration of processes and file system activities.
- **Random:** The random function offers tools for generating random integers and picking random items from a list.
- **JSON:** JSON is a data structure that can be encoded and decoded and is frequently used in online APIs and data exchange. This module allows dealing with JSON.
- **Re:** Supports regular expressions, a potent text-search and text-manipulation tool.
- **Collections:** Provides alternative data structures such as sorted dictionaries, default dictionaries, and named tuples.
- **NumPy:** NumPy is a core toolkit for scientific computing that supports numerical operations on arrays and matrices.
- **Pandas:** It provides high-level data structures and operations for dealing with time series and other structured data types.
- **Requests:** Offers a simple user interface for web APIs and performs HTTP requests.

PYTHON FILE I/O

- Files are used to store data in a computer disk. In this tutorial, we explain the built-in file object of Python. We can open a file using Python script and perform various operations such as writing, reading, and appending. There are various ways of opening a file. We are explained with the relevant example. We will also learn to perform read/write operations on binary files.
- Python's file input/output I/O system offers programs to communicate with files stored on a disc. Python's built-in methods for the file object let us carry out actions like reading, writing, and adding data to files.

- The open() method in Python makes a file object when working with files. The name of the file to be opened and the mode in which the file is to be opened are the two parameters required by this function. The mode can be used according to work that needs to be done with the file, such as "r" for reading, "w" for writing, or "a" for attaching.
- After successfully creating an object, different methods can be used according to our work. If we want to write in the file, we can use the write() functions, and if you want to read and write both, then we can use the append() function and, in cases where we only want to read the content of the file we can use read() function. Binary files containing data in a binary rather than a text format may also be worked with using Python. Binary files are written in a manner that humans cannot directly understand. The rb and wb modes can read and write binary data in binary files.

DJANGO FRAMEWORK

- Django is a web application framework written in Python programming language. It is based on MVT Model View Template design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement. This framework uses a famous tag line: The web framework for perfectionists with deadlines.
- By using Django, we can build web applications in very less time.
- Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.
- Python Django is a web framework that allows to quickly create efficient web pages. Django is also called batteries included framework because it provides built-in features such as Django Admin Interface, default database SQLite3, etc. When you're building a website, you always need a similar set of components: a way to handle user authentication signing up, signing in, signing out, a management panel for your website, forms, a way to upload files, etc. Django gives you ready-made components to use. Django Architecture.
- Django is based on MVT Model-View-Template architecture which has the following three parts .

- **Model:** The model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database generally relational databases such as MySQL, Postgres.
- **View:** The View is the user interface that you see in your browser when you render a website. It is represented by HTML/CSS/Javascript and Jinja files.
- **Template:** A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted. To check more, visit – Django Templates.

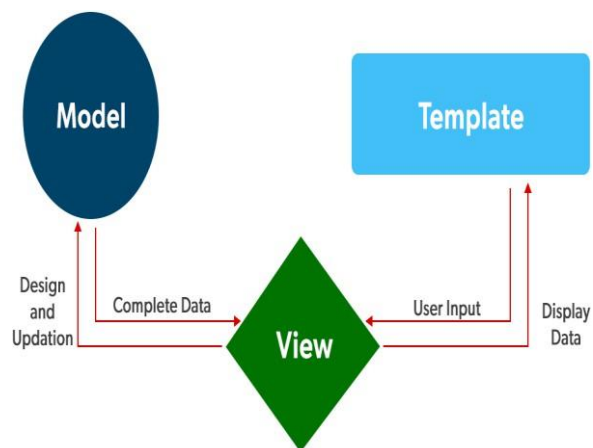


Fig: Django Framework

6. SYSTEM TESTING

6.1 INTRODUCTION

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing presents an interesting anomaly for the software engineer.

TESTING OBJECTIVES INCLUDE

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an undiscovered error.

TESTING PRINCIPLES

- All tests should be traceable to end user requirements.
- Tests should be planned long before testing begins.
- Testing should begin on a small scale and progress towards testing in large.
- Exhaustive testing is not possible.
- To be most effective testing should be conducted by an independent third party.

6.2 TESTING STRATEGIES

A Strategy for software testing integrates software test cases into a series of well planned steps that result in the successful construction of software. Software testing is a broader topic for what is referred to as Verification and Validation. Verification refers to the set of activities that ensure that the software correctly implements a specific function Validation refers he set of activities that ensure that the software that has been built is traceable to customer's requirements.

UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design that is the module. Using procedural design description as a guide, important control paths are tested to uncover errors within the boundaries of the module. The unit test is normally white box testing oriented and the step can be conducted in parallel for multiple modules.

INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associated with the interfacing. The objective is to take unit tested methods and build a program structure that has been dictated by design.

TOP-DOWN INTEGRATION

Top-down integrations an incremental approach to construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control program. Modules subordinate to the main program are incorporated in the structure either in the breath-first or depth-first manner.

BOTTOM-UP INTEGRATION

This method as the name suggests, begins construction and testing with atomic modules i.e., modules at the lowest leveling the program structure. Because the modules are integrated in the bottom-up manner the processing required for the modules subordinate to a given level is always available and the need for stubs is eliminated.

VALIDATION TESTING

At the end of integration testing software ids completely assembled as a package. Validation testing is the next stage which can be defined as successful when the software functions in the manner reasonably expected by the customer.

Reasonable expectations are those defined in the software requirements specifications. Information contained in those sections form a basis for validation testing approach.

SYSTEM TESTING

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all system elements have been properly integrated to perform allocated function.

6.3 TEST CASES

TC. No	Test Case	Input	Expected Output	Observed Output	Result
1	Login	Enter Wrong User Name and Password	Invalid Login Details	User name and Password are invalid	Pass
2	Login	Enter User Name and Password	Login Successful	Login Successful	Pass
3	Mobile Number	Enter Alphanumeric characters	Mobile number must be digits only	Mobile number in 10 digits only	Fail
4	Upload file	Browse file	file uploaded successfully	file Uploaded successful	Pass

7. SCREEN SHOTS

LOGIN PAGE

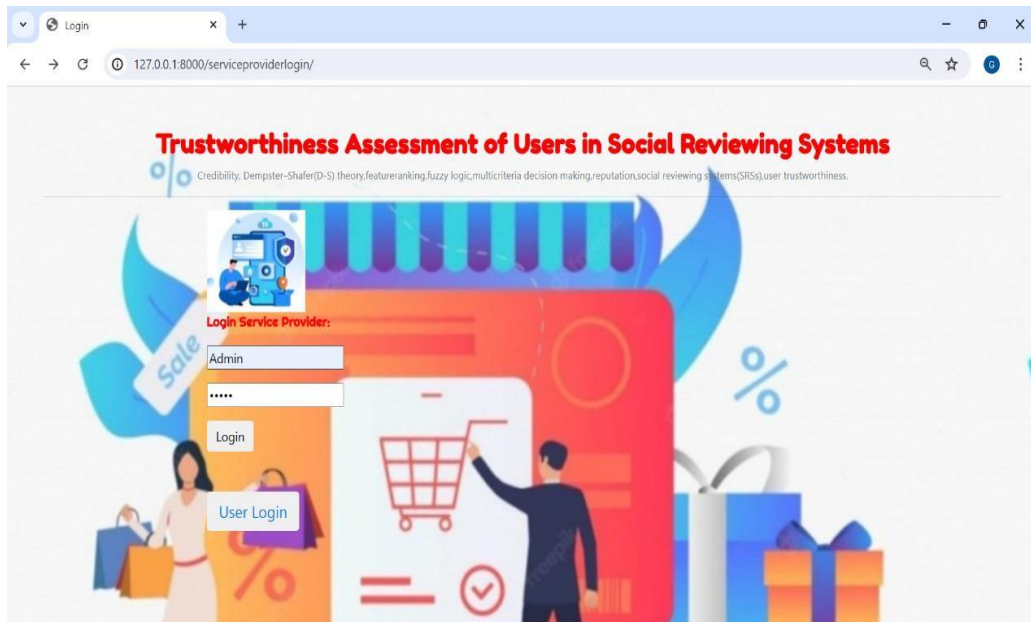


Fig: Login Page

TRUSTWORTHINESS ASSESSMENT OF USERS HOME PAHE

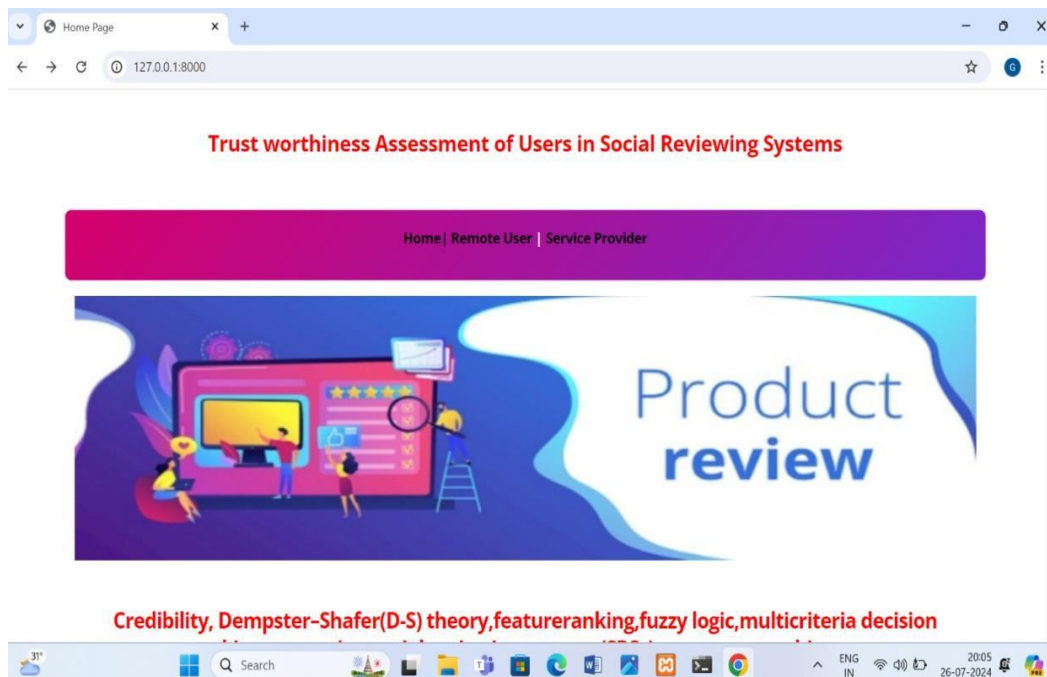


Fig: Trustworthiness Assessment Of Users Home Page

VIEW PREDICTION OF PRODUCT TRUST

PREDICTION OF PRODUCT TRUST!!!

ENTER ALL PRODUCT DATASET DETAILS HERE !!!

Enter reviewerID	A18K1ODH1I2MVB	Enter Product_Id	B007WTAJTO
Enter reviewerName	Omie	Enter helpful	[0, 0]
Enter reviewText	advertised. You can never have too much phone memory, since I download a lot of stuff this was a no brainer for me.	Enter overall_rating	5
Enter summary	MOAR SPACE!!!	Enter unixReviewTime	2182659200
Enter reviewDate	25-10-2021	Enter day_diff	409
Enter helpful_yes	0	Enter total_vote	0

Predict

Fig: View Prediction Of Product Trust

VIEW YOUR PROFILE

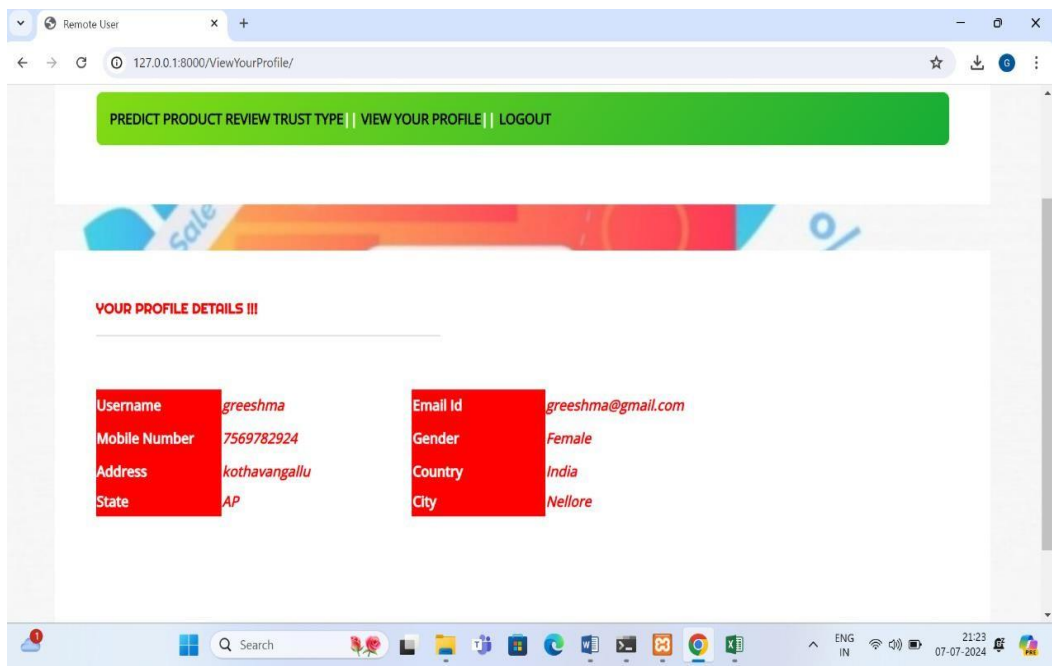


Fig: View Your Profile

VIEW PRODUCT REVIEW DATASETS TRAINED AND TEST RESULTS

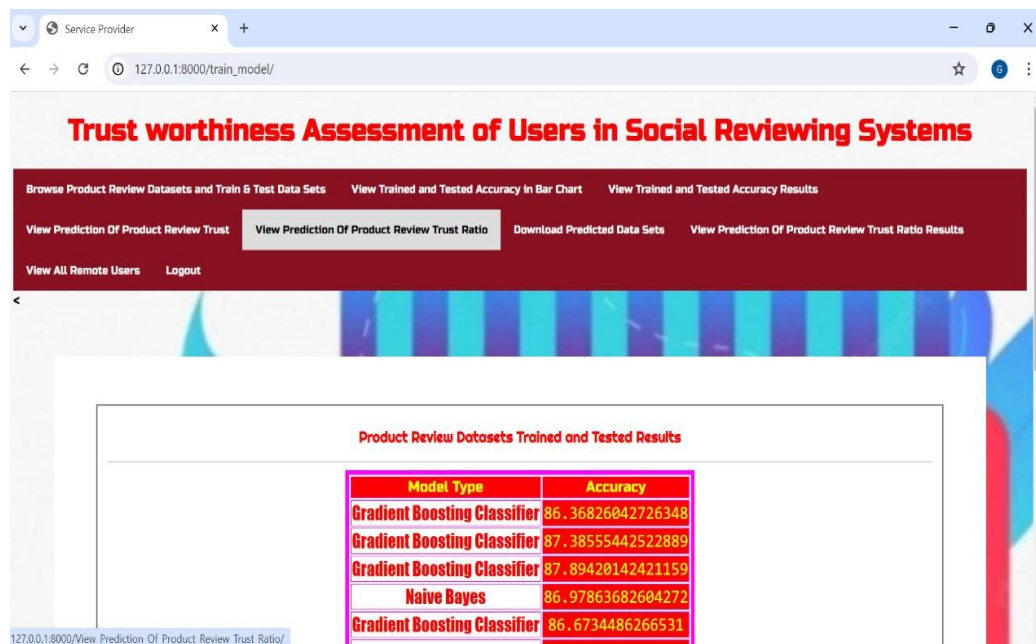


Fig: View Product Review Datasets Trained And Test Results

VIEW TRAINED AND TESTED ACCURACY IN GRAPHS

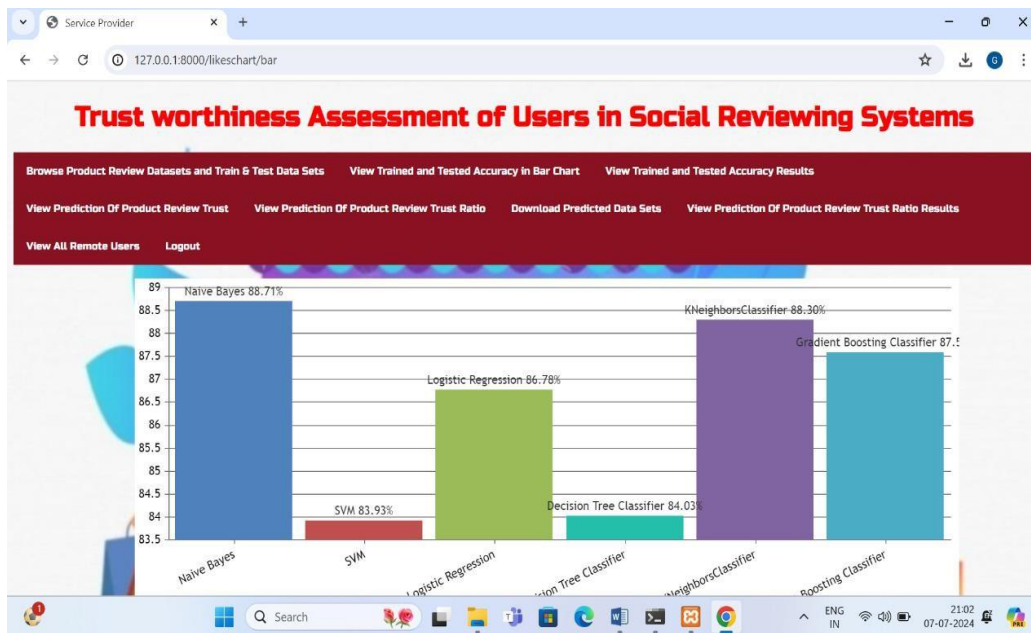


Fig: View Trained And Tested Accuracy In Graphs

VIEW TRAINED AND TESTED ACCURACY IN PIE CHARTS



Fig: View Trained and Tested Accuracy in Pie Chart

VIEW PRODUCT TRUST PREDICTON TYPE DETAILS

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/View_Prediction_Of_Product_Review_Trust/'. The page title is 'Trust worthiness Assessment of Users in Social Reviewing Systems'. Below the title, there is a navigation bar with several links: 'Browse Product Review Datasets and Train & Test Data Sets', 'View Trained and Tested Accuracy in Bar Chart', 'View Trained and Tested Accuracy Results', 'View Prediction Of Product Review Trust', 'View Prediction Of Product Review Trust Ratio', 'Download Predicted Data Sets', 'View Prediction Of Product Review Trust Ratio Results', 'View All Remote Users', and 'Logout'. The main content area displays a table titled 'View Product Trust Prediction Type Details !!!'. The table has columns: reviewerID, Product_id, reviewerName, helpful, reviewText, overall_rating, summary, untdReviewTime, and rev. The table contains three rows of data.

reviewerID	Product_id	reviewerName	helpful	reviewText	overall_rating	summary	unttdReviewTime	rev
AZIDCSC6NVONIZ	B007WTAJTO	2Cents!	(0, 0)	It's mini storage. It doesn't do	5	Not a lot to really be said	1367193600	29
A837QPVOZ9YAD	B007WTAJTO	Airedad	(15, 21)	I'm amazed. I ordered this from	5	Very fast class 10 card - and excellent	2203174400	25
ATE41UMICAP2M	B007WTAJTO	Al Campbell	(0, 0)	This product arrived	5	Quick delivery,	2461022200	27

Fig: View Product Trust Prediction Type Details

VIEW PRODUCT TRUST TYPE FOUND RATIO DETAILS

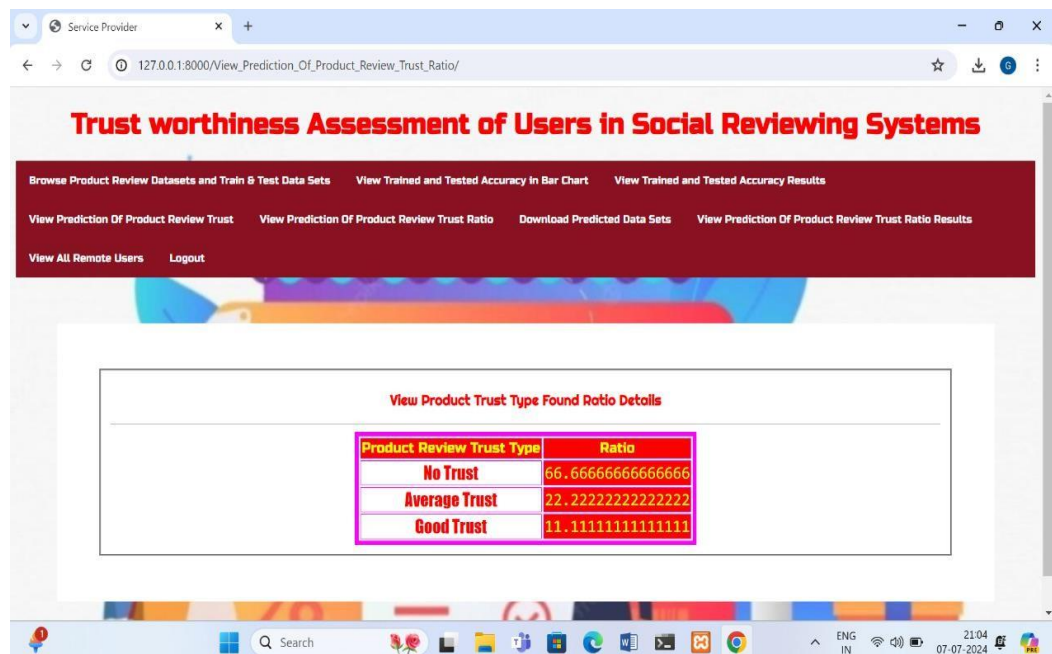
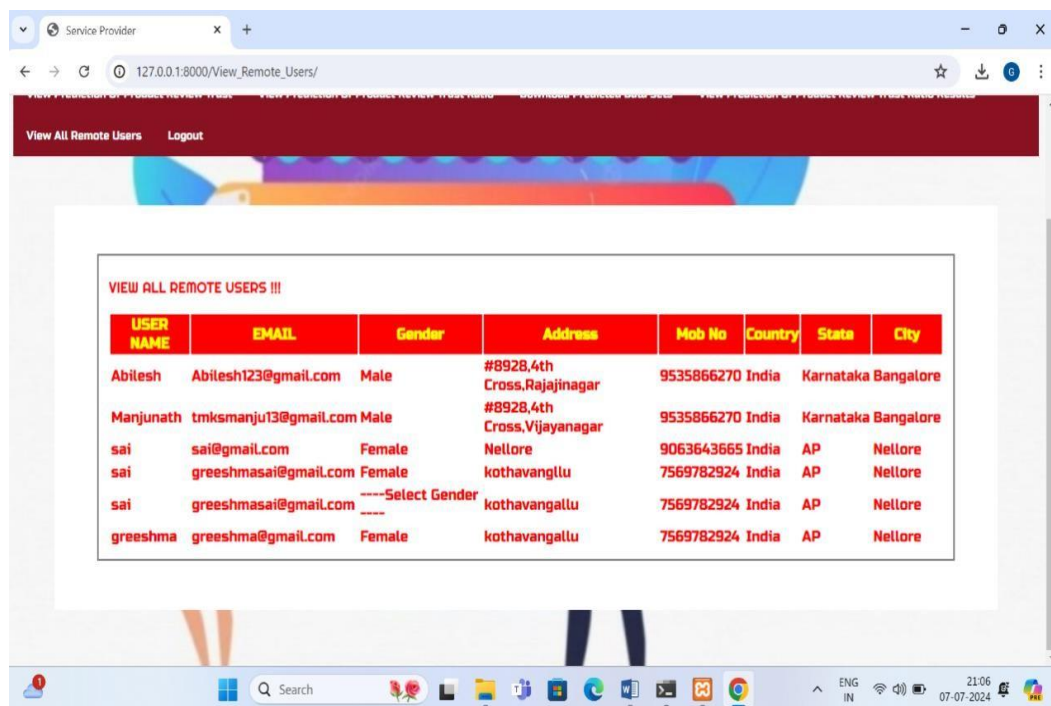


Fig: View Product Trust Type Found Ratio Details

VIEW ALL REMOTE USERS



VIEW ALL REMOTE USERS !!!

USER NAME	EMAIL	Gender	Address	Mob No	Country	State	City
Abilesh	Abilesh123@gmail.com	Male	#8928,4th Cross,Rajajinagar	9535866270	India	Karnataka	Bangalore
Manjunath	tmksmanju13@gmail.com	Male	#8928,4th Cross,Vijayanagar	9535866270	India	Karnataka	Bangalore
sai	sai@gmail.com	Female	Nellore	9063643665	India	AP	Nellore
sai	greeshmasai@gmail.com	Female	kothavangllu	7569782924	India	AP	Nellore
sai	greeshmasai@gmail.com	----Select Gender	kothavangallu	7569782924	India	AP	Nellore
greeshma	greeshma@gmail.com	Female	kothavangallu	7569782924	India	AP	Nellore

Fig: View All Remote Users

8. CONCLUSION

CONCLUSION

This study proposed an answer for the issue of trust the board inside the setting of the informal communities, where it means quite a bit to manage the subjectivity of the location of malevolent ways of behaving and the need of objectivity to plan a programmed cycle to relegate trust degrees to clients in view of their action in the interpersonal organization. To this point, we have moved toward the ambiguity and subjectivity in the survey examination from the interpersonal organization through the fluffy hypothesis. We have utilized on the hypothesis of proof in order to gadget a MCME-DM cycle to total the decisions according to numerous viewpoints and upgrade the trust assessment. We have played out a sensible trial crusade considering the Howl and Amazon dataset and showed that totaling the result of different models permits accomplishing higher exactness in identifying vindictive surveys. We have additionally thought about our methodology against the vitally related works in the current writing and showed that our methodology got improved viability by utilizing 80% and 100 percent of the considered dataset.

FUTURE ENHANCEMENT

In future works, we intend to explore more exhaustively the impact of normal assaults toward a suggestion framework to upgrade the security of such an answer, notwithstanding the investigation of the protection worries of such frameworks, by taking into account the critical lawful structures, for example, The EU General Information Insurance Guideline GDPR.

9. REFERENCES

1. M. Faloutsos, T. Karagiannis, and S. Moon, Online social networks, *IEEE Netw.*, vol. 24, no. 5, pp. 4–5, Sep/Oct. 2010.
2. J. Castro, J. Lu, G. Zhang, Y. Dong, and L. Martinez, —Opinion Dynamics based group recommender systems, *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 12, pp. 2394–2406, Dec. 2018.
3. F. Xiong, X. Wang, S. Pan, H. Yang, H. Wang, and C. Zhang, Social Recommendation with evolutionary opinion dynamics, *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 10, pp. 3804–3816, Oct. 2020.
4. R. Urena, G. Kou, Y. Dong, F. Chiclana, and E. Herrera-Viedma, A Review on trust propagation and opinion dynamics in social networks and Group decision making frameworks, *Inf. Sci.*, vol. 478, pp. 461–475, Apr. 2019.
5. Y. Xiang, E. Bertino, and M. Kutylowski, Security and privacy in social Networks, *Concurrency Comput. Practice Exp.*, vol. 29, no. 7, 2017, Art. No. E4093.
6. D. Irani, S. Webb, K. Li, and C. Pu, Modeling unintended personal information Leakage from multiple online social networks, *IEEE Internet Comput.*, vol. 15, no. 3, pp. 13–19, May/Jun. 2011.
7. A. Nosko, E. Wood, and S. Molema, All about me: Disclosure in online Social networking profiles: The case of Facebook, *Comput. Human Behav.*, vol. 26, no. 3, pp. 406–418, 2010.
8. K. Krombholz, D. Merkl, and E. Weippl, —Fake identities in social Media: A case study on the sustainability of the Facebook business Model, *J. Service Sci. Res.*, vol. 4, no. 2, pp. 175–212, 2012.
9. E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, —The rise of social bots, *Commun. ACM*, vol. 59, no. 7, pp. 96–104, 2016.
10. X. Wang et al., —Game theoretic suppression of forged messages in Online social networks, *IEEE Trans. Syst., Man, Cybern., Syst.*, early Access, Mar. 5, 2019, doi: 10.1109/TSMC.2019.2899626.
11. M. A. Ferrag, L. Maglaras, and A. Ahmim, Privacy-preserving schemes For ad hoc social networks: A survey, *IEEE Commun. Surveys Tuts.*, Vol. 19, no. 4, pp. 3015–3045, 4th Quart., 2017.

12. I. Kayes and A. Iamnitchi, Privacy and security in online social Networks: A survey, *Online Soc. Netw. Media*, vols. 3–4, pp. 1–21, Oct. 2017.
13. S. R. Sahoo and B. B. Gupta, Classification of various attacks and their Defence mechanism in online social networks: A survey, *Enterprise Inf. Syst.*, vol. 13, no. 6, pp. 832–864, 2019.
14. C. Zhang, J. Sun, X. Zhu, and Y. Fang, Privacy and security for online Social networks: Challenges and opportunities, *IEEE Netw.*, vol. 24, No. 4, pp. 13–18, Jul. 2010.
15. H. Gao, J. Hu, T. Huang, J. Wang, and Y. Chen, Security issues in online social networks, *IEEE Internet Comput.*, vol. 15, no. 4, Pp. 56–63, Jul./Aug. 2011.
16. F. Buccafurri, G. Lax, D. Migdal, S. Nicolazzo, A. Nocera, and C. Rosenberger, Contrasting false identities in social networks by trust Chains and biometric reinforcement, in *Proc. Int. Conf. Cyberworlds*, 2017, pp. 17–24.
17. W. Sherchan, S. Nepal, and C. Paris, A survey of trust in social Networks, *ACM Comput. Surveys*, vol. 45, no. 4, p. 47, 2013.
18. G. Liu et al., TOSI: A trust-oriented social influence evaluation method in contextual social networks, *Neurocomputing*, vol. 210, pp. 130–140, Oct. 2016.
19. H. Xia, F. Xiao, S.-S. Zhang, X.-G. Cheng, and Z.-K. Pan, —A reputationbased Model for trust evaluation in social cyber-physical systems, *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 792–804, Apr.–Jun. 2020.
20. X. Niu, G. Liu, and Q. Yang, Trustworthy website detection based on Social hyperlink network analysis, *IEEE Trans. Netw. Sci. Eng.*, vol. 7, No. 1, pp. 54–65, Jan.–Mar. 2020.
21. R. E. Bellman and L. A. Zadeh, Decision-making in a fuzzy environment, *Manag. Sci.*, vol. 17, no. 4, pp. B141–B164, 1970.
22. C. Esposito, A. Castiglione, and F. Palmieri, Information theoretic based Detection and removal of slander and/or false-praise attacks for Robust trust management with Dempster Shafer combination of linguistic Fuzzy terms, *Concurrency Comput. Practice Exp.*, vol. 30, no. 3, 2018, Art. No. E4302.
23. S. K. T. Lam, D. Frankowski, and J. Riedl, do you trust your

Recommendations? An exploration of security and privacy issues in recommender systems, in *Emerging Trends in Information and Communication Security*. Berlin, Germany: Springer, 2006, pp. 14–29.

24. R. Katarya, —A systematic review of group recommender systems techniques, *Proc. Int. Conf. Intell. Sustain. Syst. (ICISS)*, Dec. 2017, Pp.425–428.
25. M. Casanovas and J. Merigo, —Fuzzy aggregation operators in decision Making with Dempster–Shafer belief structure, *Expert Syst. Appl.*, Vol. 39, no. 8, pp. 7138–7149, 2012.