

---

A

**PRACTICAL TRAINING REPORT**

ON

**SENTIMENT ANALYSIS**

By

**Aadil Azad Khan(CSE-17-06)**

**Sheikh Zubair(CSE-17-45)**

**Uzair Bashir(CSE-17-13)**

**Danish Bashir(CSE-17-47)**

**Submitted to:**

**Department of Computer Science and Engineering,**

**ISLAMIC UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**AWANTIPORA, JAMMU AND KASHMIR**

**CONTENTS**

<b>S.N.</b>	<b>TITLE</b>
1.	Introduction About The Project
2.	Organization Overview
3.	System Analysis 3.1 Proposed System 3.1.1. Feasibility Study 3.1.2. Technical Feasibility 3.1.3. Economic Feasibility 3.1.4. Behavioral Feasibility
4.	Introduction 4.1. Module Design 4.2. Data Flow Diagram
5.	Software Environment 5.1. Software and Hardware Requirement 5.2. Hardware Specifications 5.3. Software Specifications
6.	Testing Introduction 6.1. Test Plans 6.1.1. Unit Testing 6.1.2. Integration Testing 6.1.3. Validation Testing or System Testing 6.1.4. Output Testing or User Acceptance Testing
7.	Implementation 7.1. Implementation Procedures

- 
- 7.1.1. User Training
  - 7.1.2. Training On the Application Software
  - 7.1.3. Operational Document
  - 7.1.4. System Maintenance

8. Screen Shots

9. Conclusion and scope for future enhancement

10. Bibliography

## **INTRODUCTION ABOUT THE PROJECT**

Sentiment analysis is the task of identifying whether the opinion expressed in a text is positive or negative in general, or about a given topic. For example: "I am so happy today, good morning to everyone", is a general positive text, and the text: "titanic is such a good movie, highly recommends 10/10", expresses positive sentiment toward the movie, named titanic, which is considered as the topic of this text. Sometimes, the task of identifying the exact sentiment is not so clear even for humans, for example in the text: "I'm surprised so many people put titanic in their favorite films ever list, I felt it was a good watch but definitely not that good", the sentiment expressed by the author toward the movie is probably positive, but surely not as good as in the message that was mentioned above. In many other cases, identifying the sentiment of a given text is very difficult for an algorithm, even when it looks easy from a human perspective, for example: "if you haven't seen titanic, you're not worth my time. if you plan to see it, there's hope for you yet."

Nowadays, when the micro-blogging platforms, such as Twitter, are commonly used, the of sentiment analysis becomes even more interesting. Micro-blogging introduces a new way of communication, where people are forced to use short texts to deliver their messages, hence containing new acronyms, abbreviations, and grammatical mistakes that were generated intentionally.

Although there are several known tasks related to sentiment analysis, in this project we will focus on the common binary problem of identifying the positive/negative sentiment that is expressed by a given text toward a specific topic. In other

words, the texts that we deal with in this project must express either positive or negative sentiment, and they cannot be neutral about the topic. There are other tasks that allow a text to be neutral about a specific topic, or even totally objective, i.e. expressing no interest in the topic at all. By narrowing down the problem like this, we get a classic binary classification case.

In this work, we will use two sets of manually annotated texts: 1) movie reviews from IMDB ([www.imdb.com](http://www.imdb.com)) – a world-known cinema portal, and 2) Twitter statuses.

The second set is a collection of Twitter short messages (bounded by 140 characters), an annotated topic, and the sentiment that is expressed toward it. The annotation of the topic and the sentiment was done by human annotators. The first set contains reviews about various movies, unlimited in their text length, annotated by humans with their sentiment toward the movie being described. The topic in this case is not specifically mentioned, but it can be implied from the text. The main reason for using both sets in this project, was to show the difficulties of guessing the sentiment in short and often ungrammatical English texts, as opposed to a relatively long and well established English texts, as used in the second set. We used three classifiers: Naïve Bayes, KNN and SVM over a set of features that were extracted from the texts using techniques taken from the field of natural language processing (NLP).

## **Modules:**

### **Login:**

In this module, user will enter the User Id and Password. It is then checked, and only valid user id and password will get entry into search zone. This is a security feature to avoid entry of unauthorized users.

### **Enter and Predict:**

Through this, user can input a review message and the review analysis system gives the predicted result to the user.

---

## **2. ORGANIZATION OVERVIEW**

### **About DUCAT Creative**

Getting a job is as difficult as beating the crowd because being in the corporate world demands a lot from the applicant because of which the applicants are putting their best, which results in the increment of difficulty level. You can see each and every thing is connected but the solution of this problem is either spending years to reach to a desired position or come to Ducat. At Ducat we provide the entire necessary computer training which helps the newbies and also the experienced workers so that they can achieve better recognition in this competitive world.

### **How Ducat is Helpful?**

Like other educational and training industry at Ducat you will be offered varieties programs but the instructors make the difference and make Ducat stand out from others. We have a variety of skilled and trained trainers whose approach is different which you can see anywhere. Ducat contributes a lot to the knowledge of its trainees and we try our level hard to contribute the best to increase our trainee's ability so that they stand out from others and whatever they contribute to the corporate world automatically becomes productive. Not only the fresher but also the corporates who are not able to deal with the rising technology and software are also helped here. We try our level best to deliver our services to every corner of the world by the help of customized education. Our motto is to deliver the best services to you and that is why we have taken the customized approach because we do not want you to compromise with your education.

It is not necessary that you have to leave your job in order to make-up with us. You can contact our experts and can get the best result. To serve you we are always at your service, you can contact us as and when you get time and clear your queries.

### **What types of services are offered by Ducat?**

Ducat provides the best available programs which helps in enhancing the technical skills which seems to be beneficial for all the applicants.

- **Software Development:**

We provide the best and latest IT software training which helps all the fresher and the corporates to understand well and give them the knowledge to go hand in hand with the latest technologies. This does not only help the companies but also increases the self-level to deal with all the necessary software.

- **Instructor led campus:**

Ducat helps all the new instructors to get the best exposure to show their talent in right way.

- **Workshops and Placement Service:**

At Ducat, workshops are held to increase the understanding level because theoretical values are always not enough and workshops helps in getting the practical knowledge which results in better understanding. As everything leads to the placement because if the institute does not provide placement services then it is ultimately bad for the applicants but we provide the best placement services and for that we give our best to give you the best.



---

### **3. SYSTEM ANALYSIS**

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It

---

does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

Here in the Email to Fax server project, a detailed study of existing system is carried along with all the steps in system analysis. An idea for creating a better project was carried and the next steps were followed.

### **3.4.1 FEASIBILITY STUDY**

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

### **3.4.2 TECHNICAL FEASIBILITY**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology

---

may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

### **3.4.3 ECONOMIC FEASIBILITY**

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it gives an indication of the system is economically possible for development.

### **3.4.4 BEHAVIORAL FEASIBILITY**

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

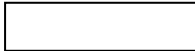

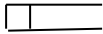

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are

---

considered carefully and conclude that the project is behaviorally feasible.

#### 4.6 Data Flow Diagram

The Data flow Diagram shows the flow of data. It is generally made of symbols given below:

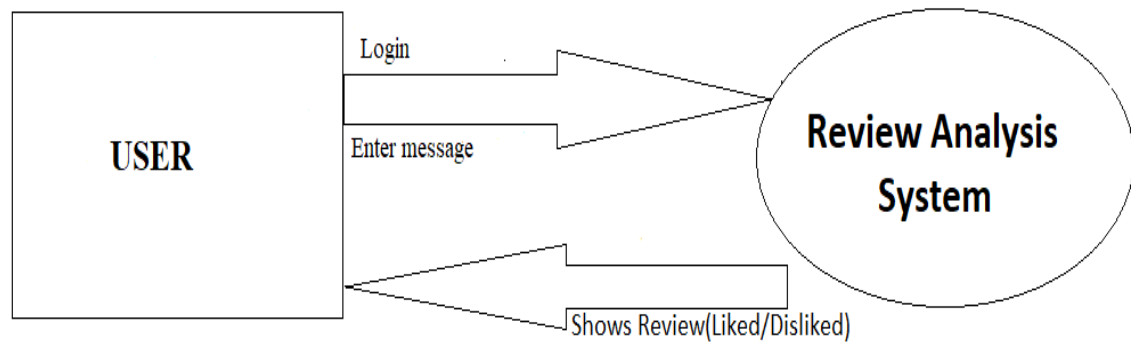
- (1) A **square** shows the Entity: - 
- (2) A **Circle** shows the Process: - 
- (3) An **open Ended Rectangle** shows the data store: -- 
- (4) An **arrow** shows the data flow: - 

The DFD can be up to several levels.

The 0 level DFD states the flow of data in the system as seen from the outward in each module.

The first level DFD show more detail, about the single process of the 0 level DFD

The second level DFD can show even more details and so on.

**Context Level DFD**

## **5. SOFTWARE ENVIRONMENT**

### **Python:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**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 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, is "sunsetting" in less than a month on January 1, 2020 (after extension; first planned for 2015), and the Python team of volunteers will not fix security issues, or improve it in other ways after that date. With the end-of-life, only Python 3.5.x and later will be supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

### **Python Interpreter:**

An interpreter is a program that reads and executes code. This includes source code, pre-compiled code, and scripts. Common interpreters include Perl, Python, and Ruby interpreters, which execute Perl, Python, and Ruby code respectively.

Interpreters and compilers are similar, since they both recognize and process source code. However, a compiler does not execute the code like an interpreter does. Instead, a compiler simply converts the source code into machine code, which can be run directly by the operating system as an executable program. Interpreters bypass the compilation process and execute the code directly.

Since interpreters read and execute code in a single step, they are useful for running scripts and other small programs. Therefore, interpreters are commonly installed on Web servers, which allows developers to run executable scripts within their webpages. These scripts can be easily edited and saved without the need to recompile the code.

While interpreters offer several advantages for running small programs, interpreted languages also have some limitations. The most notable is the fact that interpreted code requires an interpreter to run. Therefore, without an interpreter, the source code serves as a plain text file rather than an executable program. Additionally, programs written for an interpreter may not be able to use built-in system functions or access hardware resources like compiled programs can. Therefore, most software applications are compiled rather than interpreted.

## **Why Python**

### **7 Reasons Why You Must Consider Writing Software Applications in Python**

#### **1) Readable and Maintainable Code**

While writing a software application, you must focus on the quality of its source code to simplify maintenance and updates. The syntax rules of Python allow you to express concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords instead of punctuations. Hence, you can use Python to build custom applications without writing additional code. The readable and clean code base will help you to maintain and update the software without putting extra time and effort.

#### **2) Multiple Programming Paradigms**

Like other modern programming languages, Python also supports several programming paradigm. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features help you to use Python for developing large and complex software applications.

#### **3) Compatible with Major Platforms and Systems**

At present, Python is supports many operating systems. You can even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted programming language. It allows you to you to run the same code on multiple platforms without recompilation. Hence, you are not required to recompile the code after making any alteration. You can run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to make changes to the code without increasing development time.

#### **4) Robust Standard Library**

Its large and robust standard library makes Python score over other programming languages. The standard library allows you to choose from a wide range of modules according to your precise needs. Each module further enables you to add functionality to the Python application without writing additional code. For instance, while writing

---

a web application in Python, you can use specific modules to implement web services, perform string operations, manage operating system interface or work with internet protocols. You can even gather information about various modules by browsing through the Python Standard Library documentation.

## **5) Many Open Source Frameworks and Tools**

As an open source programming language, Python helps you to curtail software development cost significantly. You can even use several open source Python frameworks, libraries and development tools to curtail development time without increasing development cost. You even have option to choose from a wide range of open source Python frameworks and development tools according to your precise needs. For instance, you can simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and CherryPy. Likewise, you can accelerate desktop GUI application development using Python Gui frameworks and toolkits like PyQt, PyJs, PyGUI, Kivy, PyGTK and WxPython.

## **6) Simplify Complex Software Development**

Python is a general purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization. You can take advantage of the data analysis features of Python to create custom big data solutions without putting extra time and effort. At the same time, the data visualization libraries and APIs provided by Python help you to visualize and present data in a more appealing and effective way. Many python developers even use Python to accomplish artificial intelligence (AI) and natural language processing tasks.

## **7) Adopt Test Driven Development**

You can use Python to create prototype of the software application rapidly. Also, you can build the software application directly from the prototype simply by refactoring the Python code. Python even makes it easier for you to perform coding and testing simultaneously by adopting test driven development (TDD) approach. You can easily write the required tests before writing code and use the tests to assess the application code continuously. The tests can also be used for checking if the application meets predefined requirements based on its source code.

However, Python, like other programming languages, has its own shortcomings. It lacks some of the built-in features provided by other

---

modern programming language. Hence, you have to use Python libraries, modules, and frameworks to accelerate custom software development. Also, several studies have shown that Python is slower than several widely used programming languages including Java and C++. You have to speed up the Python application by making changes to the application code or using custom runtime. But you can always use Python to speed up software development and simplify software maintenance.

## 5.1 SOFTWARE AND HARDWARE SPECIFICATION

### **Hardware Specification:**

Microprocessor	: Intel Atom(minimum)
RAM	: 2 GB(min)
Hard Disk	: 200 MB

### **Software Specification:**

Language	: Python
Concept	: Data Science and Machine Learning
Libraries	: <ul style="list-style-type: none"><li>▪ tkinter</li><li>▪ pandas</li><li>▪ string</li><li>▪ PIL</li><li>▪ re</li><li>▪ sklearn</li><li>▪ pyinstaller</li></ul>
Other Tools	: Python 3.x IDLE Anaconda(Jupyter Lab)

## 5.2 Project Architecture and Platform

**Front End** : Python (tkinter)

**Back End** : Python and Machine Learning

## 6. TESTING INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

---

Validation : Are we doing the right job?

Verification : Are we doing the job right?

Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives.

They are

- Testing is a process of executing a program with the intend of finding an error.
- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

---

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## **6.1 TEST PLAN**

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing

- 
- ❖ Data validation Testing
  - ❖ Output Testing

### **6.1.1 UNIT TESTING**

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum



---

of test inputs. Some flaws in the internal logic of the modules were found and were rectified.

### **6.1.2 INTEGRATION TESTING**

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After unit testing in Sell-Soft System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

### **6.1.3 VALIDATION TESTING OR SYSTEM TESTING**

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in

---

data structures or external data access, performance errors and initialization errors and termination errors.

#### **6.1.4 OUTPUT TESTING OR USER ACCEPTANCE TESTING**

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points

- Input Screen Designs,
- Output Screen Designs,
- Online message to guide the user and the like.

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

#### **Validation Checking:**

At the culmination of integration testing, software is completely assembled as a package; interfacing errors have been uncovered and corrected, and a final series of software test-validation checks may begin. Validation can be defined in many ways, but a simple definition (Albeit Harsh) is that validation succeeds when software functions in a manner that can be reasonably expected by a customer. Software validation is achieved through a series of black-box tests to be

---

conducted and a test procedure defines specific test cases that will be used in attempt to uncover errors in conformity with requirements. Both the plan and procedure are designed to ensure that all functional requirements are satisfied; all performance requirements are achieved; documentation is correct and human –Engineered and other requirements are met. Once the application was made free of all logical and interface errors , inputting dummy data to ensure that the software developed satisfied all the requirements of the user did validation checks .However , the data are created with the intent of determining whether the system will process them correctly .

In the proposed system, if the clients click the send button after selecting a file from his file list, then the system will show the confirmation message for sending files. Similarly if a client makes an attempt to download a file from the server file list, then also the system will show the confirmation message for downloading. This is how the data validations were made in the proposed system.

---

## **7. IMPLEMENTATION**

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

Implementation is the stage of the project where the theoretical design is tuned into a working system. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after thorough testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system

---

testing and changeover. The implementation state involves the following tasks:

- ◆ Careful planning.
- ◆ Investigation of system and constraints.
- ◆ Design of methods to achieve the changeover.
- ◆ Training of the staff in the changeover phase.

## **7.1 Implementation Procedures**

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that

- ◆ The active user must be aware of the benefits of using the new system.
- ◆ Their confidence in the software is built up.
- ◆ Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

### **7.1.1 User Training**

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected

---

from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

### **7.1.2 Training on the Application Software**

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered.

It should then cover information needed by the specific user/group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

### **7.1.3 Operational Document**

Once the implementation plan is decided, it is essential that the user of the system is made familiar and comfortable with the environment. Education involves right atmosphere and motivating the user. A documentation providing the whole operations of the system is being developed in such a way that the user can work with it in well consistent way. The system is developed user friendly so that the user can work the system from the tips given in the application itself. Useful tip and guidance is given inside the application itself to help the user. Users have to be made aware that what can be achieved with the new

---

system and how it increases the performance of the system. The user of the system should be given a general idea of the system before he uses the system.

#### **7.1.4 System Maintenance**

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes". Maintenance may be defined by describing four activities that are undertaken after a program is released for use

---

## 8. SCREENSHOTS

### Home page



My Project

**RESUME FINDER**

username: admin

password: \*\*\*\*\*

Log In Reset

 **ISLAMIC UNIVERSITY OF SCIENCE & TECHNOLOGY**  
Approved Under 12(b) & 2(f) of UGC Act 1956



```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (i
l)] on win32
> print("copyright",
>
> : RESTART: E:\D
code\GUI.py ==
```



## 9. CONCLUSION AND FUTURE SCOPE

- >sentiment or emotion classification using NRC emotion lexicon is proposed.
- >plan to include some topic modelling based sentiment analysis features
- >to predict the best product as per the customers' needs.

## 10. BIBILOGRAPHY

INTERNET:

<https://www.researchgate.net>

<https://www.cs.tau.ac.il>

<https://www.ducatinidia.com>

<https://www.python.org>