A PROJECT REPORT

*On*

***“MAILMATE,EMAIL SENDER APPLICATION”***

*Submitted in partial fulfilment of the requirements to*

KKR & KSR INSTITUTE OF TECHNOLOGY & SCIENCES

*For the award of the degree*

***BACHELOR*** *of* ***TECHNOLOGY***

***In***

***COMPUTER SCIENCE &ENGINEERING***

*Submitted by*

G. NAGENDRA REDDY (20JR1A4445)

THOTA JAYAKRISHNA (20JR1A4464)

BATHULLA VAMSI (20JR1A4438)

KARASALA AMOS (20JR1A4448)

*Under the guidance of*

**A.SANDEEP**

ASSOCIATE PROFESSOR, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING-DATA SCIENCE

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES

Autonomous

(Approved by A.I.C.T.E New Delhi || Permanently Affiliated to JNTUK, Kakinada) || Accredited with ‘A’ Grade by NAAC || NBA Accreditation status for 5 B.Tech Programmes (CSE, ECE, EEE & Mech,Civil))

Vinjanampadu (Vil), Vatticherukuru (Md), Guntur (DT), A.P-522017. [www.kitsguntur.ac.in](http://www.kitsguntur.ac.in/)

APRIL-2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING -

DATA SCIENCE

KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES

(Approved by A.I.C.T.E New Delhi || Permanently Affiliated to JNTUK, Kakinada) || Accredited with ‘A’ Grade by NAAC || NBA Accreditation status for 5 B.Tech Programmes (CSE, ECE, EEE & Mech, Civil))

Vinjanampadu (Vil), Vatticherukuru (Md), Guntur (DT), A.P-522017.

CERTIFICATE

This is to certify that this project work titled “MAILMATE,AN EMAIL APPLICATION” is the bonafide work of G. NAGENDRA REDDY**(**20JR1A4445**)**, THOTA JAYAKRISHNA (20JR1A4464)**,** BATHULLA VAMSI (20JR1A4438) and KARASALA AMOS (20JR1A4448) who carried out the work under supervision and submitted in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in Computer Science & Engineering -Data Science from **JNTU-Kakinada** during the academic year 2022-2023.

HEAD OF THE DEPARTMENT PROJECT GUIDE

EXTERNAL EXAMINER

**DECLARATION**

We hereby inform that this mini project entitled “MAILMATE,AN EMAIL APPLICATION” has been Carried out and submitted in partial fulfillment for the award to the degree of **Bachelor of Technology in Computer Science and Engineering - Data Science** to KKR & KSR INSTITUTE OF TECHNOLOGY & SCIENCES

under the guidance of A.Sandeep, ASSOCIATE PROFESSOR, Dept. of Computer Science and Engineering- Data Science. The work embodied in this project work is original and has not been submitted in part or full for any degree of this or any degree of any other university.

G. NAGENDRA REDDY (20JR1A4445)

THOTA JAYAKRISHNA (20JR1A4464)

BATHULLA VAMSI (20JR1A4438)

KARASALA AMOS (20JR1A4448)

**ACKNOWLEDGEMENT**

We would like to express our profound gratitude towards **A. Sandeep, ASSOCIATE PROFESSOR,** Department of COMPUTER SCIENCE AND ENGINEERING-Data Science**,** whoplayed a supervisory role to utmost perfection, enabled us to seek through our B.Tech main project and for guiding as an internal guide methodically and meticulously.

We are highly indebted to **Dr B Bhanu Prakash** , **Head of the Department**, Computer Science and Engineering-Data Science for providing us all the necessary support.

We are very much thankful to the **college management for** their continuous support and facilities provided.

We render our deep sense of gratitude to **Dr. P. BABU, Principal,** for permitting us to carry out our main project works. We would like to express our sincere thanks to Computer Science and Engineering staff for lending us their time to help us and complete the work successfully.

We would also like to thank our staff, parents and friends for their enduring encouragement and assistance whenever required.

G. NAGENDRA REDDY (20JR1A4445)

THOTA JAYAKRISHNA (20JR1A4464)

BATHULLA VAMSI (20JR1A4438)

KARASALA AMOS (20JR1A4448)

**INDEX**

**UNIT –I INTRODUCTION PageNo**

1.1 Introduction of the Project 15

1.2 Existing System 15

1.3 Problems of the Existing Systems 15

1.4 Proposed System 16

1.5 Benefits of the Proposed System 16

**UNIT II - ANALYSIS**

2.1 Requirements Analysis 17

2.1.1 Functional Requirements Analysis

2.1.2 User Requirements

2.1.3 Non Functional Requirements

2.1.4 System Requirements

2.2 Modules Description 19

2.3 Feasibility Study 19

2.3.1 Technical Feasibility

2.3.2 Operational Feasibility

2.3.3 Behavioral Feasibility

2.4 Process Model used 21

2.5 Hardware and Software Requirements 22

2.6 SRS Specification 22

**UNIT-3 DESIGN PHASE**

3.1 Design concepts 26

3.2 Design Constraints 27

3.3 Conceptual Design 29

3.4 Logical Design (Logical Tools/Logical Diagrams) 30

3.5 Architectural Design 32

3.6 Algorithms Design 33

3.7 Database Design 33

3.8 Module design Specifications 33

**UNIT IV- CODING & OUTPUT SCREENS**

4.1 Sample Coding 34

4.2 Output Screens & screens reports 37

**UNIT- 5 TESTING**

5.1 Introduction to Testing 38

5.2 Types of Testing 38

5.3 Test cases and Test Reports 40

**UNIT- 6 IMPLEMENTAION**

6.1 Implementation Process 42

6.2 Implementation Steps 42

6.3 Implementation procedure 43

6.4 User Manual 43

**UNIT- 7 CONCLUSION AND FUTURE ENHANCEMENTS**

7.1 Conclusion 44

7.2 Future Enhancements 44

**UNIT- 8BIBLIOGRAPHY**

8. Books Referred 46

**KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES**

**AUTONOMOUS**

(Approved by A.I.C.T.E New Delhi|| Permanently Affiliated to JNTUK, Kakinada|| Accredited with ‘A’ Grade by NAAC || NBA Accreditation status for 5 B.Tech Programmes (CSE, ECE, EEE & Mech, Civil))

Vinjanampadu (Vil), Vatticherukuru (Md), Guntur (DT), A.P-522017.

**INSTITUTE VISION AND MISSION**

**INSTITUTION VISION**

To produce eminent and ethical Engineers and Managers for society by imparting quality professional education with emphasis on human values and holistic excellence.

**INSTITUTION MISSION**

* To incorporate benchmarked teaching and learning pedagogies incurriculum.
* To ensure all round development of students through judicious blend of curricular, co- curricular and extra-curricularactivities.
* To support cross-cultural exchange of knowledge between industry andacademy.
* To provide higher/continued education and researched opportunities to the employees of theinstitution.

**DEPARTMENT OF CSE-DATA SCIENCE**

**Vision**

To achieve academic excellence and to build the leaders with escalating multi – skilled professionals in Data Science Engineering with global competence empowered by technical expertise with in-depth knowledge, innovative research and lifelong learning, promoting employability, higher education with socio-ethical and entrepreneurial values for development.

**Mission**

To impart quality technical education and to develop the learners as leaders in Data Science Engineering with fundamental engineering principles and professionally deft, innovative research capabilities to lead and to use technology for the progress of mankind. To changing technological environment by providing the high quality instruction, faculty, modern teaching, socio-ethical and entrepreneurial values as the inner strength for rural development.

Mission 1: Quality Education

Mission 2: Professional career

Mission 3: Entrepreneur and Research.

**KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES**

**AUTONOMOUS**

(Approved by A.I.C.T.E New Delhi|| Permanently Affiliated to JNTUK, Kakinada|| Accredited with ‘A’ Grade by NAAC || NBA Accreditation status for 5 B.Tech Programmes (CSE, ECE, EEE & Mech, Civil))

|  |  |
| --- | --- |
| **PEO:1** | Domain Knowledge: Graduates of Computer Science and Engineering shall have a string foundation in areas like mathematics, science and engineering fundamentals so as to enable them to solve and analyze engineering problems and prepare them to Careers, R&D and studies of higher level. |
| **PEO:2** | Professional Employment: Graduates of Computer Science and Engineering shall have an ability to analyze and understand the requirements of software, technical specifications required and provide novel engineering solutions to the problems associated with hardware and software. |
| **PEO:3** | Higher Degrees: Graduates of Computer Science and Engineering shall have exposure to cutting edge technologies thereby making them to achieve excellence in the areas of their studies. |
| **PEO:4** | Engineering Citizenship: Graduates of Computer Science and Engineering shall work in teams on multidisciplinary projects with effective communication skills and leadership qualities. |
| **PEO:5** | Lifelong Learning:Graduates of Computer Science and Engineering shall have successful career wherein they strike a balance between ethical values and commercial values. |

Vinjanampadu (Vil), Vatticherukuru (Md), Guntur (DT), A.P-522017.

**Program Specific Outcomes (PSOs)**

**PSO1: Application Development**

Able to develop the business solutions through Latest Software Techniques and tools for real time Applications.

**PSO2: Professional and Leadership**

Able to practice the profession with ethical leadership as an entrepreneur through participation in various events like Ideathon, Hackathon, project expos and workshops.

**PSO3: Computing Paradigms**

Ability to identify the evolutionary changes in computing using Data Sciences, Apps, Cloud computing and IoT.

**Program Educational Objectives (PEOs)**

|  |  |
| --- | --- |
| **PEO:1** | Domain Knowledge: Graduates of Computer Science and Engineering shall have a string foundation in areas like mathematics, science and engineering fundamentals so as to enable them to solve and analyze engineering problems and prepare them to Careers, R&D and studies of higher level. |
| **PEO:2** | Professional Employment: Graduates of Computer Science and Engineering shall have an ability to analyze and understand the requirements of software, technical specifications required and provide novel engineering solutions to the problems associated with hardware and software. |
| **PEO:3** | Higher Degrees: Graduates of Computer Science and Engineering shall have exposure to cutting edge technologies thereby making them to achieve excellence in the areas of their studies. |
| **PEO:4** | Engineering Citizenship: Graduates of Computer Science and Engineering shall work in teams on multidisciplinary projects with effective communication skills and leadership qualities. |
| **PEO:5** | Lifelong Learning:Graduates of Computer Science and Engineering shall have successful career wherein they strike a balance between ethical values and commercial values. |

**KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES**

**AUTONOMOUS**

(Approved by A.I.C.T.E New Delhi|| Permanently Affiliated to JNTUK, Kakinada|| Accredited with ‘A’ Grade by NAAC || NBA Accreditation status for 5 B.Tech Programmes (CSE, ECE, EEE & Mech, Civil))

Vinjanampadu (Vil), Vatticherukuru (Md), Guntur (DT), A.P-522017.

**Program Outcomes**

1. 1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2. **2. Problem Analysis:** Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineeringsciences.
3. **3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety and the cultural, societal and environmental considerations.
4. **4. Conduct investigations of complex problems:** Use research based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide validconclusions.
5. **5. Modern tool usage:** Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of thelimitations.
6. **6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of and the need for sustainabledevelopment.
8. **8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineeringpractice.
9. **9. Individual and team work:** Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinaryenvironments.
10. **10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with the society at large, such as, being able to comprehend and write effective reports and design documentations and presentations and also give clear instructions.
11. **11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and as a leader in a team, to manage projects and in multidisciplinaryenvironments.
12. **12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technologicalchange.

**KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES**

**AUTONOMOUS**

(Approved by A.I.C.T.E New Delhi|| Permanently Affiliated to JNTUK, Kakinada|| Accredited with ‘A’ Grade by NAAC || NBA Accreditation status for 5 B.Tech Programmes (CSE, ECE, EEE & Mech, Civil))

Vinjanampadu (Vil), Vatticherukuru (Md), Guntur (DT), A.P-522017.

**Project Course Outcomes**

**CO425.1:** Analyze the System of Examinations and identify the problem.

**CO425.2:** Identify and classify the requirements.

**CO425.3:** Review the related Literature.

**CO425.4:** Design and modularize the project.

**CO2425.5:** Construct, integrate, test and implement the project.

**CO425.6:** Prepare the project Documentation and present the report using appropriate method.

**Course Outcomes - Program Outcomes mapping**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **PSO1** | **PSO2** | **PSO3** |
| **C425.1** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C425.2** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C425.3** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C425.4** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C425.5** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C425.6** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Course Outcomes – Program Outcome correlation**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **PSO1** | **PSO2** | **PSO3** |
| **C425.1** | 3 | 2 |  | 2 |  | 2 | 2 |  |  |  |  |  | 3 |  |  |
| **C425.2** | 3 | 2 |  | 2 |  | 2 |  |  | 1 | 2 | 1 |  | 2 | 2 | 3 |
| **C425.3** |  | 3 |  |  | 2 |  |  |  |  |  |  | 2 | 1 |  | 3 |
| **C425.4** | 2 | 2 | 3 | 2 | 2 |  | 1 | 1 | 2 | 2 |  |  | 3 |  | 2 |
| **C425.5** |  |  | 3 |  | 2 |  | 2 |  | 1 |  | 2 | 2 | 3 | 2 |  |
| **C425.6** |  |  |  |  |  | 3 |  |  |  | 2 |  | 2 |  | 3 |  |

**3:High 2:Medium 1: Low**

**CO-PO Mapping with Reasons:**

* 1. **1. CO425.1** is mapped with PO1, PO2 and PO4, PO6, PO7 as basic knowledge of Engineering and problem Analysis activities are highly essential to conduct examinations on existing systems which have been using in industries as a part of and to define the problem of proposedsystem.
  2. **2. CO425.2** is mapped with PO1, PO2, PO4 and PO6, PO9, PO10, PO11 as for identification, gathering analysis and classification of requirements for the proposed system, basic knowledge of engineering and Analysis steps along with complex problem analysis through the efforts of team work in order to meet the specific needs of thecustomer.
  3. **3. CO425.3** is mapped with PO2, PO5 and PO12 as to conduct the literature review and to examine the relevant systems to understand and identify the merits and demeritsof each too enhance and develop the proposed as per theneed.
  4. **4. CO425.4** is mapped with PO1, PO2, PO3, PO4, PO5 and PO7, PO8, PO9, PO10 because modularization and design of the project is needed after requirements elicitation. For modularization and design of the project, Basic knowledge of Engineering, Analysis capabilities, Design skills and communication is needed between team members as different modules are designed individually beforeintegration.
  5. **5. CO425.5** is mapped with PO3, PO5, PO7, PO9, PO11 and PO12 as to construct the project latest technologies are needed. The development of project is done individually and in groups with well-defined communication by using the engineering and management principles.
  6. **6. CO425.6** is mapped with PO6, PO10 and PO12 because during and after completion of the project, documentation is needed along with proper methods of presentation through understanding and application of engineering and management principles, which in turn needs well defined communication between the team members with all the ethical values. Even the project development team defines the future enhancements as a part of the project development after identifying the scope of theproject.

**CO-PSOs Mapping with Reasons:**

* + **1. CO425.1** is mapped with **PSO1** as examining of existing systems and identification of the problem is a part of Application Development activity and identification of evolutionary changes in latesttechnologies.
  + **2. CO425.2** is mapped with **PSO1, PSO2** and **PSO3**as identifying and classifying the requirements is a part of Application development and evolutionary computing changes and also follows ethicalprinciples.
  + **3. CO425.3** is mapped with **PSO1, PSO3** as review of literature is a part of application development activity by recognizing the computing technologies and their evolutionary changes.
  + **4. CO425.4** is mapped with **PSO1, PSO3** because modularization and logical design is also a part of Application development and follows computing changes using Deep learning technology.
  + **5. CO425.5** is mapped with **PSO1, PSO2** as Testing, Development and Integration of project activities are part of Application development and follows ethicalprinciples.
  + **6. CO425.6** is mapped with **PSO2** as for project documentation and presentation; the project team members apply the professional and leadership qualities

**KKR & KSR INSTITUTE OF TECHNOLOGY AND SCIENCES**

**AUTONOMOUS**

(Approved by A.I.C.T.E New Delhi|| Permanently Affiliated to JNTUK, Kakinada|| Accredited with ‘A’ Grade by NAAC || NBA Accreditation status for 5 B.Tech Programmes (CSE, ECE, EEE & Mech, Civil))

Vinjanampadu (Vil), Vatticherukuru (Md), Guntur (DT), A.P-522017.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name of Course from | Description of the application | Attained PO |  |  |
| which Principles are |  |  |  |  |
| applied in this project |  |  |  |  |
|  |  |  |  |  |
| C324.1 | Gathering the requirements and define the problem, plan to | PO1,PO3 |  |  |
| C325.1 | develop the “Mushroom Agrology” |  |  |  |
|  |  |  |  |
| C425.1 |  |  |  |  |
|  |  |  |  |  |
| C221.1 | Each and every requirement is critically analysed, the | PO2,PO3 |  |  |
| C221.2 | process model is identified and divide into several modules. |  |  |  |
|  |  |  |  |
| C221.3 |  |  |  |  |
|  |  |  |  |  |
| C313.1 | Logical design is done by using the unified modelling | PO3,PO5,PO |  |  |
| C313.3 | language which involves individual team work. | 6,PO9 |  |  |
|  |  |  |  |
| C313.4 |  |  |  |  |
|  |  |  |  |  |
| C324.6 | Each and every module is tested, integrated, and evaluated | PO1,PO5 |  |  |
| C221.4 | in our project. |  |  |  |
|  |  |  |  |
|  |  |  |  |  |
| C221.1 | Documentation is done by all four members in the form of a | PO10 |  |  |
| C221.2 | group. |  |  |  |
|  |  |  |  |
| C222.3 |  |  |  |  |
|  |  |  |  |  |
| C221.1 | Each and every phase of the work in group is presented | P09,PO10,P |  |  |
| C221.2 | periodically. | O11,P012 |  |  |
|  |  |  |  |
| C221.3 |  |  |  |  |
|  |  |  |  |  |
| C221.6 | Implementation is done and the project will be handled by | PO3,PO5,PO |  |  |
| C325.1 | the administrator and in future updates in our project can be | 10,PO11 |  |  |
| done by the administrator. |  |  |  |
|  |  |  |  |
| C325.3 |  |  |  |  |
|  |  |  |  |  |
| C414.5 | The physical design is done by using MINT App inventor. | PO5,PO10,P |  |  |

**UNIT –I INTRODUCTION**

**ABSTRACT :**

simple yet powerful program for sending multiple emails with attachments to recipients listed in a CSV file. It uses the Python programming language and several built-in and third-party libraries to interact with the email server, construct email messages, and process CSV data. The program features a graphical user interface (GUI) that allows users to select the CSV file containing email data, enter their email account credentials, and customize the content, attachments, and subject of each email. The program performs error checking and informs the user when all emails have been sent successfully. The program can be run on any compatible computer or server and is useful for businesses or individuals who need to send a large number of personalized emails with attachments.

**1.2 EXISTING SYSTEM:**

In general , there are no such systems in the existence in comparison to our proposed idea. The idea implements over a service named as SMTP . Where there are multiple SMTP providers and there applications. Where our idea can be extension for the common mailing applications which can send mails in bulk with uncommon data as well , in comparison to the existing services like google mail , Microsoft outlook and many more.

**1.3 PROBLEMS OF THE EXISTING SYSTEMS:**

There are mainly 3 common problems that are identified by our team

There is no definite process of sending mails in bulk other than a uncommon process of sending group mails.

Overcoming the time complexity of sending the mails.

There is no such tech app which use a .csv file extension to transfer mails.

**1.4 PROPOSED SYSTEM:**

Here, we are providing a user-friendly platform to send automated mails using a simple gun based application which is developed on the basis of the python platform , the application interface contains simple user identification with there authentication . After the data which is stored as .csv extension is loaded into the application and the file contains with to address , subject of the mail , body of the mail, another features like image attachments , file attachments and in just a single click the emails will be sent with the notification dialogue box appears informing the status of the mails sent or not.

**1.5BENEFITS OF THE PROPOSED SYSTEM:**

● Time saving.

● User friendly.

● Simple to use.

● Provides accurate results.

**UNIT -II ANALYSIS**

The analysis is defined as a detailed examination of the elements or structure of something.

**2.1 REQUIREMENT ANALYSIS**:

The process to gather the software requirements from clients, analyze and document them is known as requirements engineering or requirements analysis. The goal of requirement engineering is to develop and maintain sophisticated and descriptive ‘System/Software Requirements Specification’ documents. It is a four-step process generally, which includes –

• Feasibility Study four-step

• Requirements Gathering

• Software Requirements Specification

• Software Requirements Validation

The basic requirements of our project are:

• Python installed

• Research Papers

• Datasets

• Accuracy calculation

**2.1.1 FUNCTIONAL REQUIREMENT ANALYSIS :**

Functional requirements explain what has to be done by identifying the necessary task, action, or activity that must be accomplished. Functional requirements analysis will be used as the top-level functions for functional analysis.

**2.1.2 USER REQUIREMENTS ANALYSIS** :

User Requirements Analysis is the process of determining user expectations for a new or modified product. These features must be quantifiable, relevant, and detailed. The main user requirements of our project are as follows:

Operating system: Windows 7 or later, macOS 10.9 or later, or a Linux distribution

Python version: 3.6 or later

Required Python packages: tkinter, smtplib, email

Additional Python packages: csv (included with Python), filedialog (included with tkinter), messagebox (included with tkinter)

Internet connection to send emails using an SMTP server (in this case, Gmail's SMTP server)

A CSV file containing the email addresses, subjects, bodies, image paths, and PDF paths for each email to be sent

Valid email address and password for the email account that will be used to send the emails

Note that the specific requirements may vary depending on the operating system, Python version, and other factors.

**2.1.3 NON-FUNCTIONAL REQUIREMENTS ANALYSIS :**

Non-functional requirements describe the general characteristics of a system. They are also known as quality attributes. Some typical non-functional requirements are Performance, Response Time, Throughput, Utilization, and Scalability.

**2.1.4 SYSTEM REQUIREMENTS ANALYSIS :**

Hardware Requirements:

System: Windows 7 or later, macOS 10.9 or later, or a Linux distribution(intel i3 / arm m1)

Hard Disk: 256 GB.

RAM: 4GB.

**2.2 MODULES DESCRIPTION :**

**User Module :** It is the place where user can access our website to use the services regarding the download of the application and visiting documentation.

**System Module :** The task which we are done in the system is shown by the system module.

**2.3 FEASIBILITY STUDY:**

Feasibility Study is a high-level capsule version of the entire process intended to answer a number of questions like What is the problem? Is there any feasible solution to the given problem? Is the problem even worth solving? A feasibility study is conducted once the problem is clearly understood. A feasibility study is necessary to determine that the proposed system is Feasible by considering the technical, Operational, and Economical factors. By having a detailed feasibility study the management will have a clear-cut view of the proposed system. A well-designed feasibility study should provide a historical background of the business or project, the operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. The following feasibilities are considered for the project in order to ensure that the project is variable and it does not have any major obstructions. Feasibility study encompasses the following things:

● Technical Feasibility

● Operational Feasibility

● Behavioral feasibility

In this phase, we study the feasibility of all proposed systems And pick the best feasible solution for the problem. The feasibility is studied based on three main factors as follows:

**2.3.1 TECHNICAL FEASIBILITY:**

In this step, we verify whether the proposed systems are technically feasible or not. i.e., all the technologies required to develop the system are available readily or not. Technical Feasibility determines whether the organization has the technology and skills necessary to carry out the project and how this should be obtained. The system can be feasible because of the following grounds.

• All necessary technology exists to develop the system

• This system is flexible and can be expanded further

• This system can give a guarantee of accuracy, ease of use, and reliability

• Our project is technically feasible because all the technology needed for our project is readily available.

**2.3.2 OPERATIONAL FEASIBILITY:**

In this step, we verify different operational factors of the proposed systems like manpower, time, etc., whichever solution uses less operational resources, is the best operationally feasible solution. The solution should also be operationally possible to implement. Operational Feasibility determines if the proposed system satisfied user objectives could be fitted into the current system operation. The present system of predicting can be justified as operationally feasible based on the following grounds.

● The methods of processing and presentation are completely accepted by the clients since they can meet all user requirements.

● The clients have been involved in the planning and development of the system.

● The proposed system will not cause any problem under any circumstances.

Our project is operationally feasible because the time requirements and personnel requirements are satisfied. We are a team of four members and we worked on this project for three working months.

**2.3.3 BEHAVIORAL FEASIBILITY:**

This device will help people to save time. As there will be no wastage of time, the user will be satisfied.

**2.4 PROCESS MODEL USED:**

The model that is basically being followed is the WATERFALL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If a system exists and modification and the addition of a new module are needed, analysis of the present system can be used as a basic model. The design starts after the requirement analysis is compl3ete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model, the sequence of activities performed in a software development project is Requirement Analysis, Project Planning, System Design, Detail Design, Coding, Unit testing, System integration & testing. Here the linear ordering of these activities is critical. The end of the phase and the output of one phase is the input of the other phase.

The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of the spiral model are also incorporated after the people concerned with the project review completion of each of the phases of the work done. WATERFALL MODEL was chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

Image

Fig. Waterfall model

**2.5 SOFTWARE AND HARDWARE REQUIREMENTS:**

**SOFTWARE REQUIREMENTS:**

Operating system: Windows 7 or later, macOS 10.9 or later, or a Linux distribution

Coding language: Python.

IDE: Python IDLE, viscode .

**HARDWARE REQUIREMENTS:**

System: Intel Core i3 2.4 GHz/ arm m1.

Hard Disk: 500GB.

RAM: 4GB.

**2.6 SRS SPECIFICATION:**

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

System Requirements Specification It is a collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Projects are subject to three sorts of required elements. Business requirements describe in business terms what must be delivered or accomplished to provide value.

● Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)

● Process requirements describe activities performed by the developing organization. For instance, process requirements could specify methodologies that must be followed, and constraints that the organization must obey.

Product and process requirements are closely linked. Process requirements often specify the activities that will be performed to satisfy a product requirement. For example, a maximum development cost requirement (a process requirement) may be imposed to help achieve a maximum sales price requirement (a product requirement) a requirement that the product is maintainable (a product requirement) often is addressed by imposing requirements to follow particular development styles. A system engineering requirement can be a description of what a system must do, referred to as a Functional Requirement. This type of requirement specifies something that the delivered system must be able to do. Another type of requirement specifies something about the system itself, and how well it performs its functions. Such requirements are often called Nonfunctional requirements, or ‘Performance requirements’ or ‘Quality of service requirements’. Examples of such requirements include usability, availability, reliability, supportability, testability, and maintainability.

A collection of requirements define the characteristics or features of the desired system. A ‘good’ list of requirements as far as possible avoids saying how the system should implement the requirements, leaving such decisions to the system designer. Specifying how the system should be implemented is called “implementation bias” or “solution engineering”. However, implementation constraints on the solution may validly be expressed by the future owner, for example for required interfaces to external systems; for interoperability with other systems; and for commonality with other owned products.

**Functional requirements:**

The Functional Requirements Specification gives the operations and activities that a system must be able to perform. Functional requirements should include functions performed by specific screens, outlines of workflows performed by the system, and other business or compliance requirements the system must meet. It also depends upon the type of software, expected users, and the type of system where the software is used. Some constraints were given as input. This was given as input to code which predicts the disease with accuracy. The output was in the form of a predicted result.

**Non-functional requirements:**

In systems engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The nonfunctional requirements can be considered as quality attributes of a system.

Performance: The time required

Reliability: The system should be 90% reliable. Since it may need some maintenance or preparation for some particular day, the system does not need to be reliable every time. so, 80% reliability is enough.

Efficiency: Based upon the density of given values or input to calculate.

Maintainability: The system should be optimized for supportability, or ease of maintenance as far as possible.

**UNIT- III DESIGN PHASE**

**3.1 DESIGN CONCEPTS:**

The set of fundamental software design concepts are as follows:

**1. Abstraction:**

The lower level of abstraction provides a more detailed description of the solution. A sequence of instructions that contains a specific and limited function refers to a procedural abstraction. A collection of data that describes a data object is data abstraction.

**2. Architecture:**

The complete structure of the software is known as software architecture. The structure provides conceptual integrity for a system in a number of ways. The architecture is the structure of program modules where they interact with each other in a specialized way. The aim of the software design is to obtain an architectural framework of a system.

**3. Patterns:**

A design pattern describes a design structure and that structure solves a particular design problem in a specified content.

**4. Modularity:**

Modularity is the single attribute of software that permits a program to be managed easily.

**5. Information hiding:**

Modules must be specified and designed so that the information like algorithm and data presented in a module is not accessible for other modules not requiring that information.

**6. Functional independence:**

Functional independence is the concept of separation and is related to the concept of modularity, abstraction, and information hiding. The functional independence is accessed using two criteria i.e. Cohesion and coupling. Cohesion is an extension of the information hiding concept. A cohesive module performs a single task and it requires a small interaction with the other components in other parts of the program. Coupling is an indication of interconnection between modules in a structure of software.

**7. Refinement:**

Refinement is a top-down design approach. It is a process of elaboration. A program is established for refining levels of procedural details.

**8. Refactoring:**

Refactoring is the process of changing the software system in a way that it does not change the external behavior of the code and still improves its internal structure.

**9. Design classes:**

The model of software is defined as a set of design classes. Every class describes the elements of the problem domain and that focus on features of the problem which are user-visible.

**3.2DesignConstraints:**Design Constraints are generally the limitations on a design. They include imposed limitations that you don't control and limitations that are self-imposed as a way to improve a design. The following are common types of design constraints. 9 Types of Design Constraints:

**Commercial Constraints:**

Basic commercial constraints such as time and budget come under commercial constraints

**Requirements:**

Requirements specify the basic needs of a project. Ex: Functional requirements.

Non-Functional Requirements:

Non-Functional Requirements are the requirements that specify intangible elements of a design.

**Compliance:**

Compliance refers to applicable laws, regulations, and standards.

**Style:**

A style guide or multiple style guides related to an organization, brand, product, service, environment or project. For example, a product development team may follow a style guide for a brand family that constrains the colors and layout of package designs.

**Sensory Design:**

Beyond visual design, constraints may apply to taste, touch, sound and smell. For example, a brand identity that calls for products to smell fruity.

**Usability:**

Usability principles imply frameworks and standards. Ex: The principle of least astonishment.

**Principles:**

Principles include the design principles of an organization, team, or individual. For example, a designer who uses form follows function to constrain designs.

**Integration:**

A design that needs to work with other things such as products, services, systems, processes, controls, partners, and information.

**3.3 CONCEPTUAL DESIGN:**

Conceptual Design is an early phase of the design process, in which the broad outlines of function and form of something are articulated. It includes the design of interactions, experiences, processes, and strategies. It involves an understanding of people's needs - and how to meet them with products, services, & processes. Common artifacts of conceptual design are concept sketches and models.

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic, and pragmatic rules.

A UML system is represented using five different views that describe the system from a distinctly different perspective. Each view can be defined by a set of diagrams.

UML is specifically constructed through two different domains. They are:

● UML analysis modeling focuses on the user model and structural model views of the system.

● UML design modeling focuses on behavioral modeling, implementation modeling, and environment model views.

**Use case diagram:**

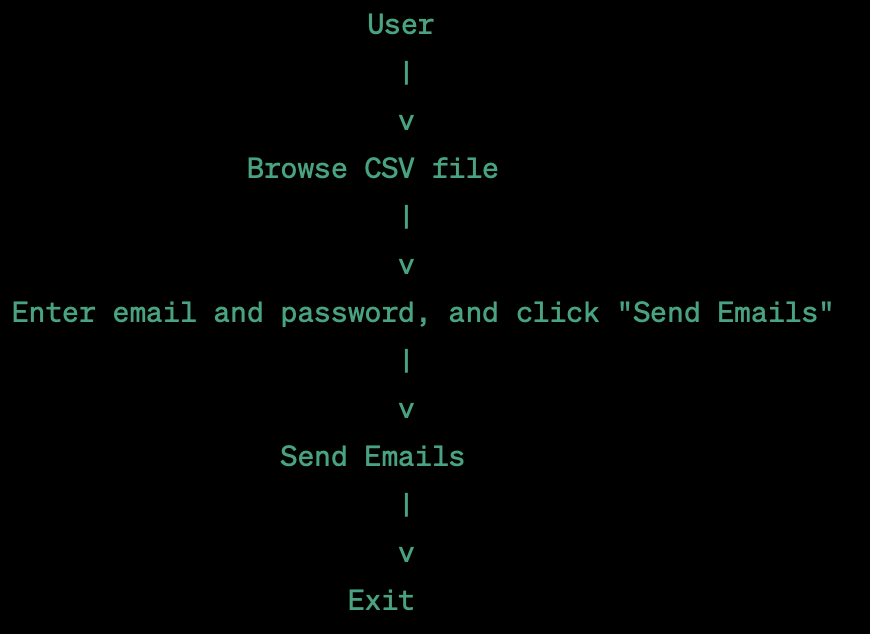
Use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. Actors are the external entities that interact with the system. The use cases are represented by either circles or ellipses.

Fig. Use case diagram

**Class diagram:**

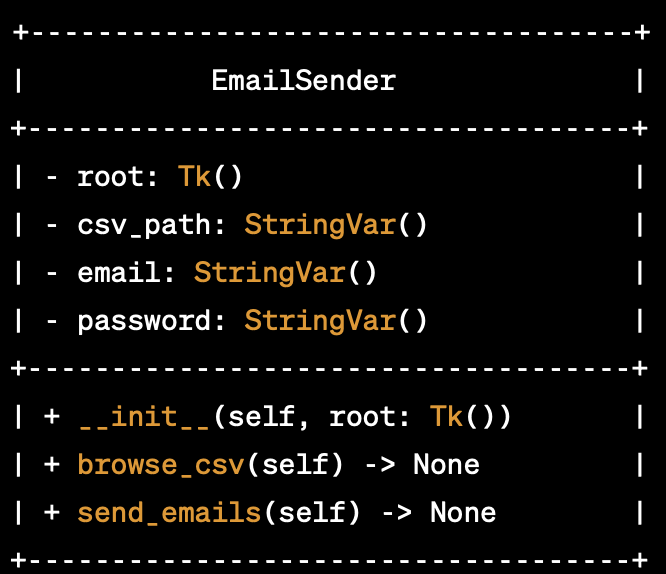
Class diagrams give an overview of a system by showing its classes and the relationships among them. Class diagrams are static – they display what interacts but not what happens when they do interact. In general a class diagram consists of some set of attributes and operations. Operations will be performed on the data values of attributes.

Fig. Class diagram

**3.4 LOGICAL DESIGN:**

The logical design of a system pertains to an abstract representation of the data flows, inputs and outputs of the system. This is often conducted via modeling, using an over-abstract and sometimes graphical model of the actual system.

**Sequence diagram:**

A Sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

Fig. Sequence diagram

**Activity diagram:**

Activity diagram is essentially a fancy flowchart: Activity and state diagrams are related. Statechart diagram focuses on objects undergoing a process. An activity diagram focuses on the flow of activities involved in a single process. The activity diagram shows the activities depend on one another.

An activity represents the performance of the task or duty in a workflow. It may also represent the execution of a statement in a procedure. You can share activities between state machines. However, transitions cannot be shared.

Activity diagrams provide a way to model the workflow of a business process, code specific information such as a class operation. The transitions are implicitly triggered by the completion of the actions in the source activities.

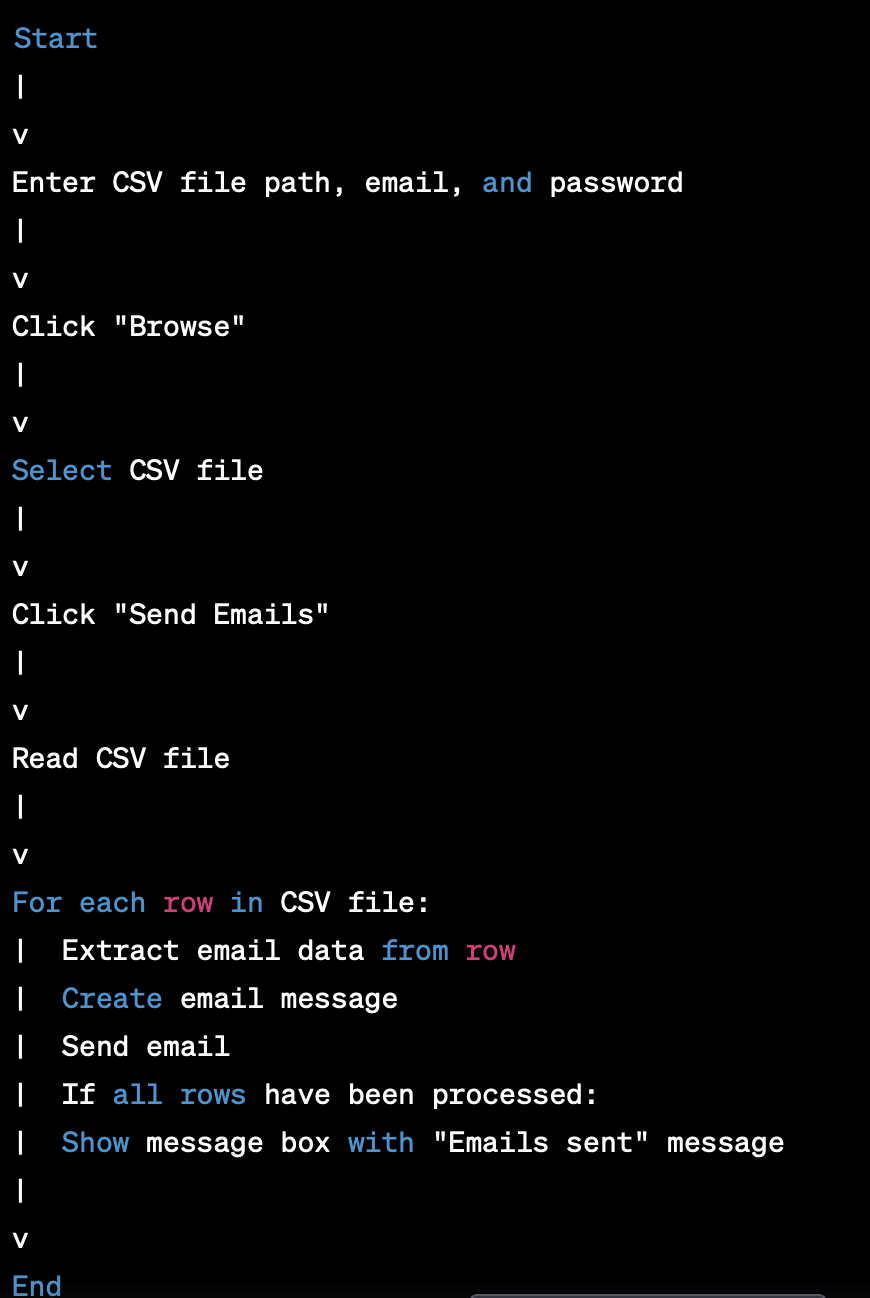
The main difference between activity and state chart diagrams is activities are activity-centric, while state chart diagrams are state-centric.

Fig. Activity diagram

**Note :**  generating UML diagrams for the code provided does not have any classes or objects to model. It is a simple procedural script. Therefore, there is no need for UML diagrams.

UML diagrams are typically used to model object-oriented systems, where classes, objects, and their relationships are important to understand and design the system. In this case, the code is a procedural script that performs a specific task, and UML diagrams are not necessary to understand or design the system.

**3.5 ARCHITECTURAL DESIGN:**

**Open interface:**

The gui asks for the login credentials of your email and the load the data set and click on send mails that’s it task is completed.

**Background:**

Building the data necessary and converting it into .csv file.

**Result:**

Displays a notification pop up showing the email are sent or not.

**3.6 ALGORITHMIC DESIGN:**

EmailSender

├── \_\_init\_\_(self, root)

│ ├── Create GUI widgets

├── browse\_csv(self)

│ ├── Open file dialog to select CSV file

│ ├── Set self.csv\_path variable to file path

├── send\_emails(self)

│ ├── Read data from CSV file

│ ├── For each row in CSV file

│ │ ├── Extract email data from row

│ │ ├── Create email message

│ │ ├── Add body text, image, and PDF attachment to email

│ │ ├── Send email

│ ├── Display message box when all emails have been sent

**3.7 DATABASE DESIGN :**

In this project we are not using the database design**.**

**3.8 MODULE DESIGN SPECIFICATIONS :**

**User Module :** It is the place where user can access our website to use the services regarding the download of the application and visiting documentation.

**System Module :** The task which we are done in the system is shown by the system module.

**UNIT-IV CODING AND OUTPUT SCREENS**

**4.1 . sample coding:**

import tkinter as tk

from tkinter import filedialog

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.image import MIMEImage

from email.mime.application import MIMEApplication

import csv

class EmailSender:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Email Sender")

self.root.geometry("400x300")

self.csv\_path = tk.StringVar()

self.email = tk.StringVar()

self.password = tk.StringVar()

# Create GUI widgets

tk.Label(self.root, text="CSV file:").pack()

tk.Entry(self.root, textvariable=self.csv\_path, width=40).pack()

tk.Button(self.root, text="Browse", command=self.browse\_csv).pack(pady=10)

tk.Label(self.root, text="Email:").pack()

tk.Entry(self.root, textvariable=self.email, width=40).pack()

tk.Label(self.root, text="Password:").pack()

tk.Entry(self.root, textvariable=self.password, width=40, show="\*").pack()

tk.Button(self.root, text="Send Emails", command=self.send\_emails).pack(pady=20)

def browse\_csv(self):

# Open a file dialog to select a CSV file

file\_path = filedialog.askopenfilename(filetypes=[("CSV files", "\*.csv")])

self.csv\_path.set(file\_path)

def send\_emails(self):

# Read data from CSV file

with open(self.csv\_path.get(), 'r') as file:

reader = csv.reader(file)

next(reader) # skip header row

for row in reader:

# Extract data from CSV row

to\_address = row[0]

subject = row[1]

body = row[2]

image\_path = row[3]

pdf\_path = row[4]

# Create email message

msg = MIMEMultipart()

msg['From'] = self.email.get()

msg['To'] = to\_address

msg['Subject'] = subject

# Add body text to email

text = MIMEText(body)

msg.attach(text)

# Add image attachment to email

if image\_path:

with open(image\_path, 'rb') as file:

img = MIMEImage(file.read(), name='image.jpg')

msg.attach(img)

# Add PDF attachment to email

if pdf\_path:

with open(pdf\_path, 'rb') as file:

pdf = MIMEApplication(file.read(), name='document.pdf')

msg.attach(pdf)

# Send email

with smtplib.SMTP('smtp.gmail.com', 587) as smtp:

smtp.starttls()

smtp.login(self.email.get(), self.password.get())

smtp.send\_message(msg)

# Show a message box when all emails have been sent

tk.messagebox.showinfo("Emails sent", "All emails have been sent!")

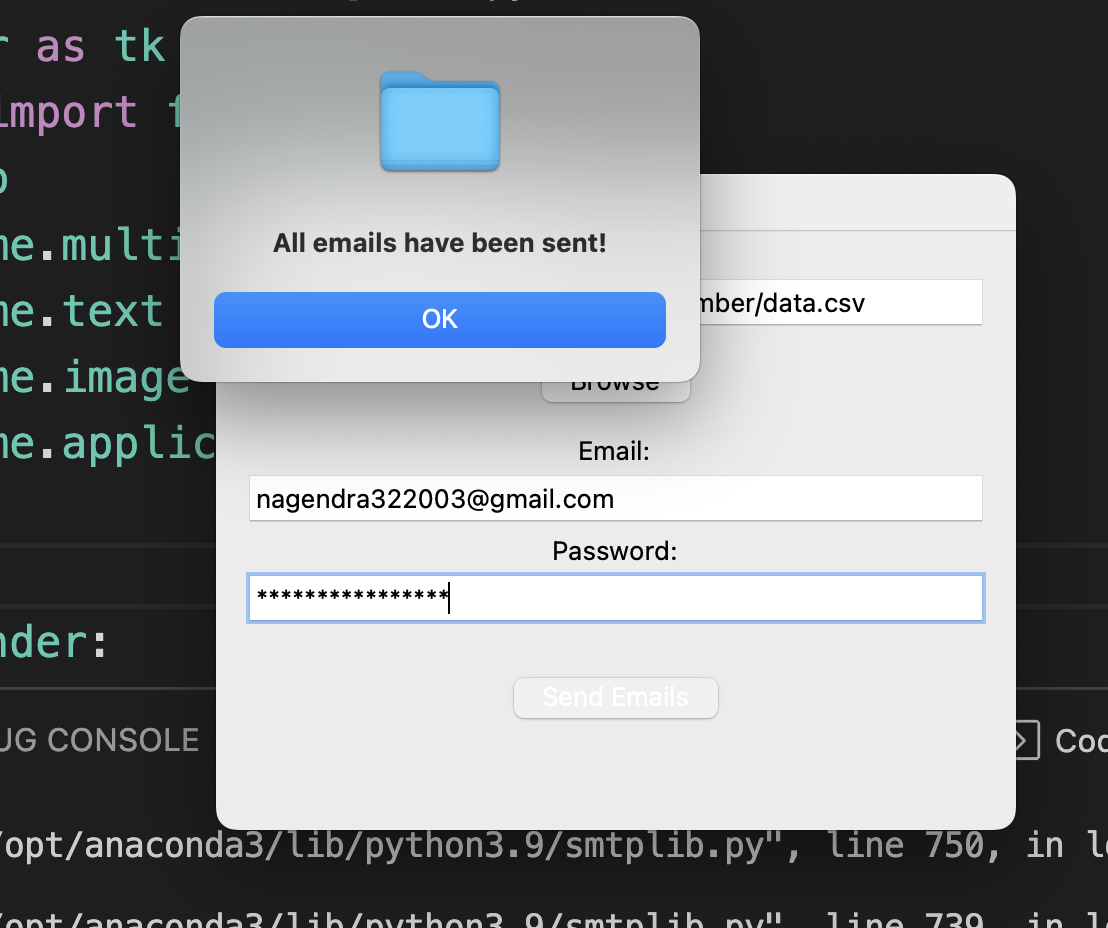
if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = EmailSender(root)

root.mainloop()

**4.2 OUTPUT SCREENS & SCREEN REPORTS:**

****

**UNIT- V TESTING**

**5.1 INTRODUCTION TO TESTING:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies, and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

**5.2 TYPES OF TESTING:**

**Unit Testing**

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

**Integration Testing**

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

**Functional Test:**

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

Functional testing is centered on the following items:

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

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

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked. The organization and preparation of functional tests are focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identifying Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined

**System Test:**

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

**White Box Testing:**

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

**Black Box Testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure, or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated as a black box. You cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**5.3 TEST CASES AND TEST REPORTS:**

**Unit testing:**

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

Test strategy and approach

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

**Integration testing:**

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

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

Test Results:

All the test cases mentioned above passed successfully. No defects were encountered.

**Acceptance testing:**

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

Test Results:

All the test cases mentioned above passed successfully. No defects were encountered.

**UNIT- VI IMPLEMENTATION**

**6.1 IMPLEMENTATION**

The email sender application is implemented in Python using the tkinter and smtplib modules. The tkinter module is used to create the GUI-based interface, while the smtplib module is used to send emails through Gmail's SMTP server.

The main functionality of the email sender application is implemented in the **EmailSender** class. The **\_\_init\_\_** method of this class creates the GUI widgets, including buttons to browse for a CSV file, enter email address and password, and send emails. The **browse\_csv** method is called when the user clicks the "Browse" button, and opens a file dialog to select a CSV file. The **send\_emails** method is called when the user clicks the "Send Emails" button, and reads the recipient information from the CSV file, creates personalized email messages for each recipient with custom subject, body, and attachments, and sends the emails using Gmail's SMTP server.

**6.2 IMPLEMENTATION STEPS:**

Step 1: Open interface

Step 2: Upload data sets

Step 3: Pre-processing

Step 4:click on send mail

Step 5: Get the result

**6.3 IMPLEMENTATION PROCEDURE :**

To use the email sender application, the user should perform the following steps:

Launch the Python app.

Enter the email address and password for the Gmail account that will be used to send emails.

Click the "Browse" button and select a CSV file containing recipient email addresses, subject, body text, image, and PDF attachments.

Click the "Send Emails" button to send personalized emails to all recipients in the CSV file.

**6.4 USER MANUAL:**

The user has to download the application from our web portal . Launch the Python app.

Enter the email address and password for the Gmail account that will be used to send emails.Click the "Browse" button and select a CSV file containing recipient email addresses, subject, body text, image, and PDF attachments.Click the "Send Emails" button to send personalized emails to all recipients in the CSV file.

**UNIT- VII CONCLUSION AND FUTURE ENHANCEMENTS**

**7.1 CONCLUSION:**

The email sender application provides a simple and user-friendly way to send personalized emails to multiple recipients using Gmail's SMTP server. It can be easily customized to add additional features or support other email providers.

**7.2 FUTURE ENHANCEMENTS:**

Here are some potential enhancements for the project that could be implemented in the future:

User authentication: Currently, the program requires the user's email and password to send emails through the Gmail server. In the future, the program could be enhanced to use OAuth 2.0 authentication, which would provide an extra layer of security by not requiring the user's password to be stored in the program.

Customization options: The program currently uses a fixed format for the email body, image, and PDF attachment. In the future, the program could be enhanced to allow users to customize these options and create their own email templates.

Error handling: The program does not currently handle errors that could occur during the email sending process, such as invalid email addresses or failed email deliveries. In the future, the program could be enhanced to handle these errors and provide more informative error messages to the user.

User feedback: The program does not currently provide any feedback to the user during the email sending process, aside from a message box at the end. In the future, the program could be enhanced to provide real-time feedback to the user, such as a progress bar or status updates for each email that is sent.

Support for multiple email providers: Currently, the program only supports sending emails through the Gmail server. In the future, the program could be enhanced to support other email providers, such as Yahoo or Outlook, and allow users to choose their preferred provider.

## UNIT- VIII BIBILOGRAPHY

Here is a bibliography of resources that could be useful for understanding the above code:

Python Software Foundation. (n.d.). Python Documentation. Retrieved from <https://docs.python.org/3/>

TkDocs. (n.d.). Tkinter 8.6 Documentation. Retrieved from <https://tkdocs.com/tutorial/>

Real Python. (n.d.). Python GUI Programming With Tkinter. Retrieved from <https://realpython.com/tkinter-python-gui-tutorial/>

Stack Overflow. (n.d.). Community-driven question-and-answer site. Retrieved from <https://stackoverflow.com/>

GeeksforGeeks. (n.d.). Computer Science portal for geeks. Retrieved from <https://www.geeksforgeeks.org/>

Python.org. (n.d.). csv - CSV File Reading and Writing. Retrieved from <https://docs.python.org/3/library/csv.html>

Python.org. (n.d.). smtplib - SMTP protocol client. Retrieved from <https://docs.python.org/3/library/smtplib.html>

Python.org. (n.d.). email - An email and MIME handling package. Retrieved from <https://docs.python.org/3/library/email.html>

GitHub. (n.d.). Where the world builds software. Retrieved from <https://github.com/>