VISVESVARAYA TECHNOLOGICAL UNIVERSITY

***"Jnana Sangama"*, Belgavi- 590014**

A Mini Project Report on

"Simple Mail Transfer Protocol"

Submitted By:

SANTHOSH	**4RA16CS082**
UZAIR AHMED	**4RA16CS102**

To the Visvesvaraya Technological University during the academic year 2018-2019
in partial fulfillment for the subject

Computer Graphics and Visualization
(VI Semester)
in
Computer Science and Engineering

Under the Guidance of:

Ms. Pruthvi Pournami Y R **B.E, M.Tech**
Assistant Professor,
Department of Computer Science & Engineering

Department of Computer Science & Engineering
Rajeev Institute of Technology
Hassan-573201
2018-2019

RAJEEV INSTITUTE OF TECHNOLOGY, HASSAN

Department of Computer Science & Engineering

C E R T I F I C A T E

This is to certify that the **Mini Project work of CGV Lab (15CSL68)** entitled **"Simple Mail Transfer Protocol"** is carried out by **Mr. Santhosh, [4RA16CS082], Mr. Uzair Ahmed, [4RA16CS102],** a bona fide student of **Rajeev Institute of Technology, Hassan** in partial fulfillment for the award of **Bachelor of Engineering,** in **Computer Science & Engineering** under Visvesvaraya Technological University, Belgaum during the year 2018. It is certified that all corrections or suggestions indicated for internal assessment have been incorporated in the report and deposited in the department library. The Mini Project report has been approved as it satisfies all the academic requirements in respect of Mini Project work prescribed by the University.

Ms.Pruthvi Pournami Y.R B.E, M.Tech	**Dr. PRAKASH H N** B.E., M.Tech, MISTE, Ph.D	**Dr. RAMAKRISHNA A N** B.E, M.Tech, PhD
Assistant Professor	Head of the Department	Principal
Dept. of CS&E	Dept. of CS&E	RIT, HASSAN
RIT, HASSAN	RIT, HASSAN	

.

Name of the examiners **Signature with date**

1.

2.

ABSTRACT

The SMTP protocol, used in the transport and delivery of e-mail messages, includes control headers along with the body of messages which, as opposed to other protocols, are not striped after the message is delivered, leaving a detailed record of e-mail transaction in the recipient mailbox. So, the depiction of this SMTP protocol has been implemented using OpenGL Functions here in this project.

We have tried to depict two cases of SMTP i.e., Successful and unsuccessful transmission of mails between sender & receiver. The objects like trees, house, computer and mail servers generated in this project. Trees are generated with triangular shaped tree tops using rectangles and triangles. Houses are achieved using pentagons and server stacks using quadrilaterals like rectangles. The mail icon is achieved using rectangles and thick lines. User is given the option to start the simulation and resetting the view using keyboard control. The respective captions from the sending of data to receiving acknowledgment and error response when receiver id not found is also displayed according to the simulation schedule.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned own efforts with success.

We would like to express our sincere thanks to our *Principal* **Dr. A N Ramakrishna,** *Rajeev Institute of Technology* for his encouragement.

We wish to express our gratitude to **Dr. H N Prakash**, *Head of the Department of CSE* for providing a good working environment and for his constant support and encouragement.

We would like to express our profound sense of gratitude to our guide **Ms. Pruthvi Pournami Y R** *Assistant professor, Department of CSE* for her guidance, initiative and encouragement that led us to complete this project.

We would also like to thank all our staffs of Computer Science and Engineering department who have directly or indirectly helped us in the successful completion of this project

SANTHOSH

UZAIR AHMED

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Overview of Computer graphics

OpenGL (**O**pen **G**raphics **L**ibrary) is a standard specification defining a cross-language, cross-platform API for writing applications that produce 2D and3D computer graphics.

The interface consists of function calls, which can be used to draw complex two-dimensional and three-dimensional scenes from simple primitives. The header file <GL/glut.h> is included in order to utilize the inbuilt functions needed in the implementation. C programming language is used for the project coding.

OpenGL serves two main purposes:

- To hide the complexities of interfacing with different 3D accelerators, by presenting the programmer with a single, uniform API.
- To hide the differing capabilities of hardware platforms, by requiring that all implementations support the full OpenGL feature set.

OpenGL's basic operation is to accept primitives such as points, lines and polygons, and convert them into pixels. This is done by a graphics pipeline known as **OpenGL state machine**. Most OpenGL commands either issue primitives to the graphics pipeline, or configure how the pipeline processes these primitives.

Basic Functionalities in OpenGL include:

- Rasterized points, lines and polygons as basic primitives
- Transform and lighting pipeline
- Z-buffering
- Texture mapping
- Alpha blending

1.2 Applications of Computer graphics

Nowadays Computer Graphics used in almost all the areas ranges from science, engineering, medicine, business, industry, government, art, entertainment, education and training.

- **CG in the field of CAD**

Computer Aided Design methods are routinely used in the design of buildings, automobiles, aircraft, watercraft, spacecraft computers, textiles and many other applications.

- **CG in presentation Graphics**

Graphics used to produce illustrations for reports or generate slides. Presentation graphics is commonly used to summarize financial, statistical, mathematical, scientific data for research reports and other types of reports.2D and 3D bar chart to illustrate some mathematical or statistical report.

- **CG in computer Art**

CG methods are widely used in both fine art and commercial art applications. Artists use a variety of computer methods including special purpose hardware, artist's paintbrush program, other pain packages, desktop packages, mathematical packages, animation packages that provide facility for designing object motion. Ex: cartoons decision is an example of computer art which uses CG.

- **Entertainment**

Computer graphics methods are now commonly used in making motion pictures, music, videos, games and sounds. Sometimes graphics objects are combined with the actors and live scenes.

- **Education and Training**

Computer generated models of physical financial, economic system is often used as education aids. For some training application special systems are designed. Ex: Specialized system is simulator for practice sessions or training of ship captain, aircraft pilots and traffic control.

- **Image Processing**

Although the methods used in CG image processing overlap, the 2 areas are concerned with fundamentally different operations. In CG a computer is used to create picture. Image processing on the other hand applies techniques to modify existing pictures such as photo scans, TV scans.

- **User Interface**

It is common for software packages to provide a graphical interface. A major component of a graphical interface is a window manager that allows a user to display multiple window area. Interface also displays menus, icons for fast selection and processing

1.3 OpenGL (Open Graphics Library)

- Most of our applications will be designed to access openGL directly through functions in three libraries. Function in the GL library have name that begin with letter 'gl' and stored in the library.
- The second is the OpenGL utility Library (GLU).
- Library uses only GL function but contains codes for creating common object and viewing.
- Rather than using a different library for each system we used available library called openGL utility toolkit (GLUT).It is used as #include<GL/glut.h>.

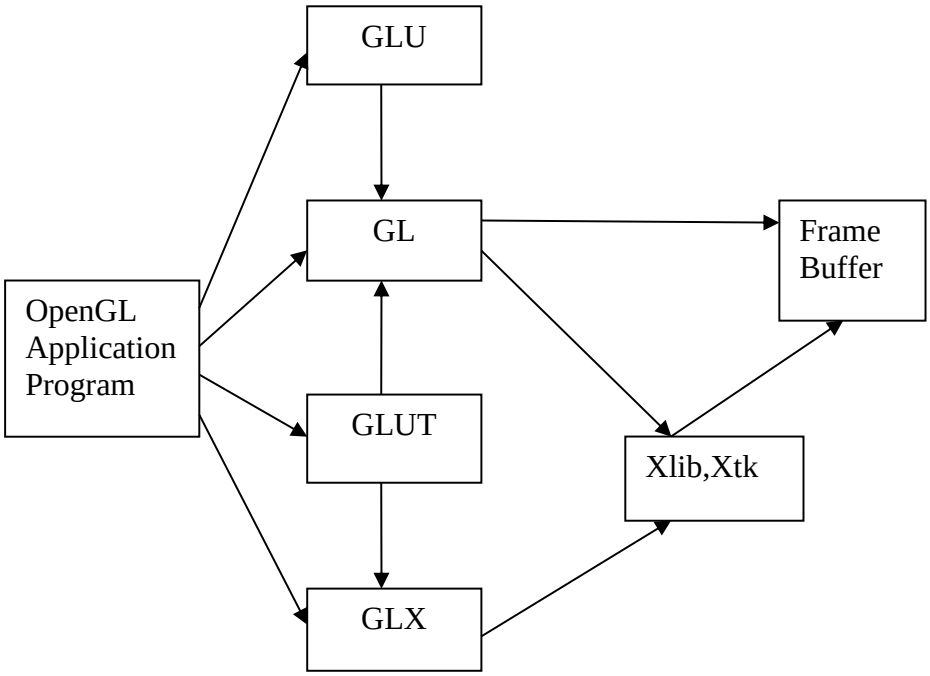


Fig1.1: Library Organization.

1.4 About Project

Simple Mail Transfer Protocol

Email is emerging as one of the most valuable services on the internet today. Most of the internet systems use SMTP as a method to transfer mail from one user to another. SMTP is a push protocol and is used to send the mail whereas POP (post office protocol) or IMAP (internet message access protocol) are used to retrieve those mails at the receiver's side.

SMTP is an application layer protocol. The client who wants to send the mail opens a TCP connection to the SMTP server and then sends the mail across the connection. The SMTP server is always on listening mode. As soon as it listens for a TCP connection from any client, the SMTP process initiates a connection on that port (25). After successfully establishing the TCP connection the client process sends the mail instantly.

Chapter 2

REQUIREMENT SPECIFICATION

2.1 Hardware requirements

- Processor : Intel 486/Pentium processor or better
- Processor Speed : 500 MHz or above
- Hard Disk : 20GB(approx)
- RAM : 64MB or above
- Storage Space : Approx. 50MB

2.2 Software Requirement

- Operating System : Windows
- Language : C/C++
- Development Tool : Code: Blocks
- Graphics Libraries : OpenGL & GLUT packages

2.3 System requirements

- An operating system to write the program and to execute it.
- The operating system should be compatible to project working environment.
- Compilers with GLUT libraries installed to compile the written OpenGL source code execute it.

2.4 User requirements

- Throughout knowledge of the OpenGL programming.
- Knowledge of basic OpenGL keywords and built in function.
- Some knowledge about the importance and requirement of OpenGL.

Chapter 3

DESIGN AND IMPLEMENTATION

This chapter gives an insight as to how the algorithm was designed, the flow of the application standard functions used, and user defined functions used in the implementation

3.1 Algorithm Design

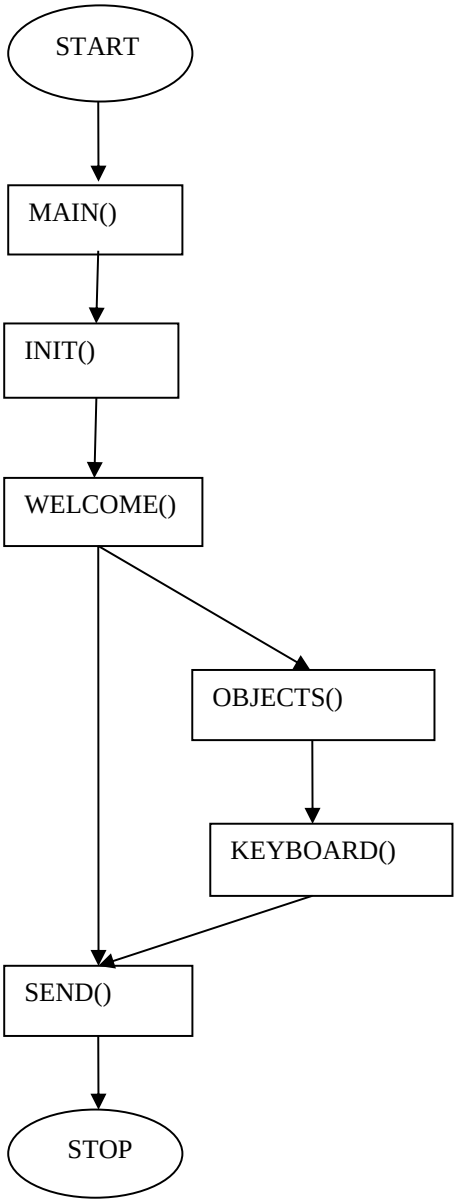
In this project an simulation of active monitoring algorithm to monitor the E-mail system SMTP is illustrated. The objects like trees, house, computer and mail servers generated in this project. Trees are generated with triangular shaped tree tops using rectangles and triangles. Houses are achieved using pentagons and server stacks using quadrilaterals like rectangles. The mail icon is achieved using rectangles and thick lines. The sender and receiver user icons are achieved by pentagons and points.

We achieve the motion of the simulation using translation and delayed projection. User is given the option to start the simulation and resetting the view using keyboard control. The respective captions from the sending of data to receiving acknowledgment and error response when receiver id not found is also displayed according to the simulation schedule.

Illustration describes the SMTP mail monitoring algorithm using the following views.

- IDLE FRONT VIEW
- SENDING OF DATA
- ACKNOWLEDGEMENT RECEIVING
- MESSAGE SENDING FAILED

3.2 Flowchart



3.3 OpenGL Functions

Glut library functions used are:

- **main():** Execution of any program always starts with main function irrespective of where it is written in a program.
- **glutSwapBuffers(void):** Performs a buffer swap on the layer in use for the current window. Specifically, glutSwapBuffers promotes the contents of the back buffer of the layer in use of the current window to become the contents of the front buffer. The contents of the back buffer then become undefined. The update typically takes

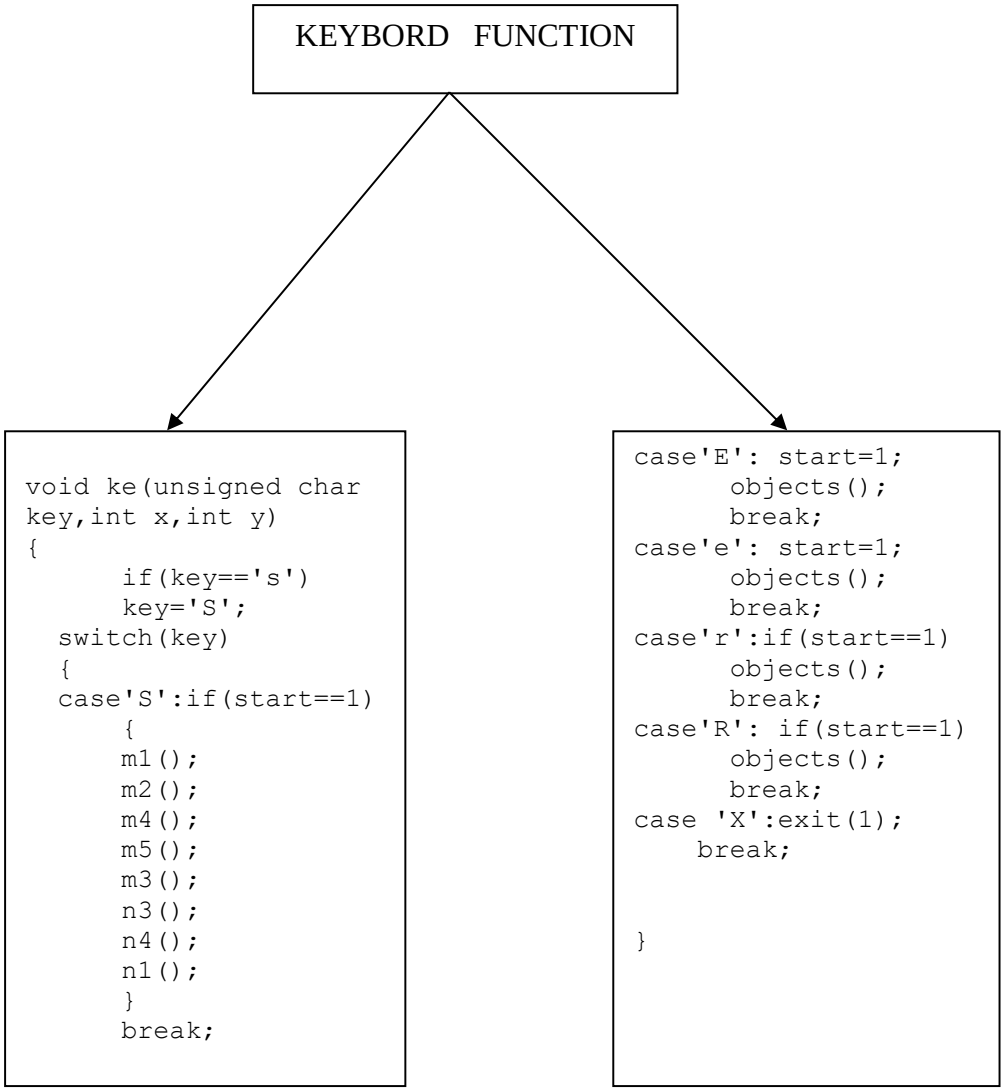
place during the vertical retrace of the monitor, rather than immediately after glutSwapBuffers is called.

- **glutPostRedisplay() :** marks the plane of current window as needing to be redisplayed. The next iteration through glutMainLoop, the window's display call back will be called to redisplay the window's normal plane. Multiple calls to glutPostRedisplay before the next display call back opportunity generates only a single redisplay call back.
- **glutInit(&argc,char ** argv):** glutInit will initialize the GLUT library and negotiate a session with the window system. During this process, glutInit may cause the termination of the GLUT program with an error message to the user if GLUT cannot be properly initialized.
- **glutInitDisplayMode(unsigned int mode):** The initial display mode is used when creating top level windows, sub windows, and overlays to determine the OpenGL display mode for the to be created window or overlay.
- GLUT_RGB specifies the structure of each pixel.
- GLUT_DEPTH is a buffer to hold the depth of each pixel.
- **glutCreateWindow(char *title):** The glutCreateWindow creates a top-level window. The name will be provided to the window system as the window's name. The intent is that the window system will label the window with the name.
- **glutMainLoop():**glutMainLoop enters the GLUT event processing loop. This routine should be called at most once in a GLUT program. Once called, this routine will never return. It will call as necessary any callbacks that have been registered.
- **glutBitmapCharacter(font,c):** Without using any display lists, glutBitmapCharacter renders the character 'c' in the named bitmap font.
- **glPointSize(size): glPointSize** specifies the rasterized diameter of points.
- **glFlush():**Different GL implementations buffer commands in several different locations, including network buffers and the graphics accelerator itself. glFlush empties all of these buffers, causing all issued commands to be executed as quickly as they are accepted by the actual rendering engine. Though this execution may not be completed in any particular time period, it does complete in finite time.
- **glRasterPos3f(x , y, z):**This function is used to set the cursor at the x, y, z location.
- **glColor3f(R,G,B):** This function is used to pick a color of specified R,G,B value.

3.3 User Defined Functions

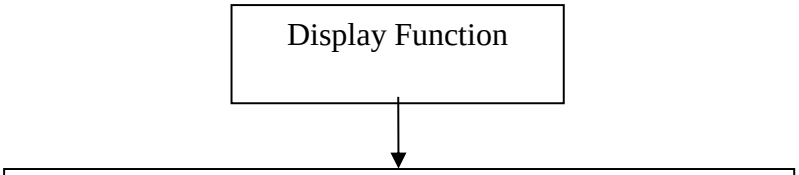
3.3.1 Keyboard Function

- E key use to enter the front view of the window
- S to start the sending simulation
- R key use to reset the Screen
- X key is use to Exit the Window



3.3.1 Display Function

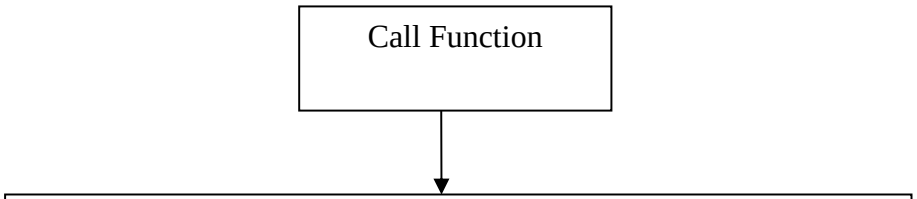
- The OpenGL graphics system calls back display () in response to a window-paint request to re-paint the window




```
void objects()
{
glClearColor(0.0,0.0,0.0,0.0);
glClear(GL_COLOR_BUFFER_BIT);

glColor3f(0.5,0.5,0.9); //serve
glBegin(GL_POLYGON);
glVertex2i(100,650);
glVertex2i(900,650);
glVertex2i(900,450);
glVertex2i(100,450);
glEnd();

glColor3f(0.5,0.0,1.0);
glBegin(GL_POLYGON); //sender home
glVertex2i(10,50);
glVertex2i(200,50);
glVertex2i(200,220);
glVertex2i(100,300);
glVertex2i(10,220);
glEnd();

glBegin(GL_POLYGON); //reciver home
glVertex2i(990,50);
glVertex2i(800,50);
glVertex2i(800,220);
glVertex2i(900,300);
glVertex2i(990,220);
glEnd();

glBegin(GL_POLYGON); //sender stack
glVertex2i(120,635);
glVertex2i(210,635);
glVertex2i(210,465);
glVertex2i(120,465);
glEnd();
glBegin(GL_POLYGON);  //reciver stack
glVertex2i(880,635);
glVertex2i(790,635);
glVertex2i(790,465);
glVertex2i(880,465);
glEnd();
```

3.3.1 Call Function

- GLUT call is use to call that particular **function** to whenever a keyboard event occurs.




```
void m2()
{
float i, j, temp = 0.3;
      temp = 0.3;
      for (i=0;i<45;i = i + temp) //data packet
storage in left side server
      {
            glColor3f(0.8, 0.8, 0.0);
            glBegin(GL_POLYGON);
            glVertex2i(140 +i, 525);
            glEnd();
            glFlush();
      }
}
void m4()
{
float i, j, temp = 0.3;
      temp = 0.3;
      for (i = 0;i < 515;i = i + temp) //left server
to right server
      {
            glColor3f(1, 0.5, 1);
            glBegin(GL_QUADS);
            glVertex2i(210 + i, 500);

            glEnd();
            glFlush();
                  glColor3f(0.0,0.0, 0.0);
      drawString(380,470, 0, "Sending Data");
      glEnd();
      glFlush();
      }
}
```

Chapter 5

SNAPSHOTS

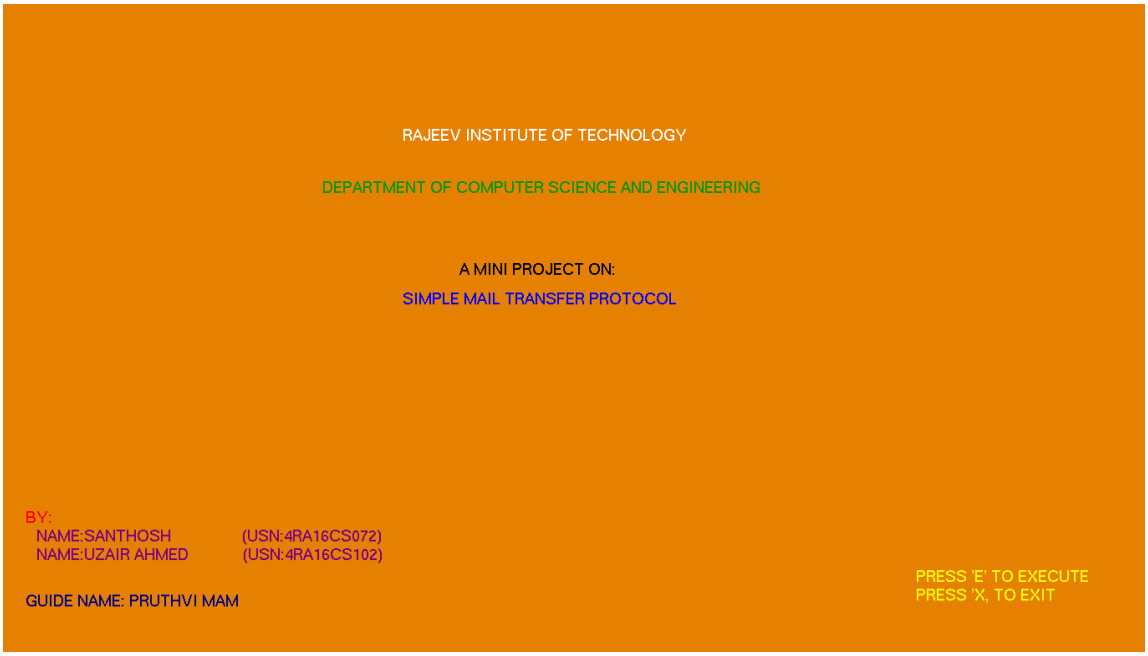


Fig 4. 1: Instruction Page

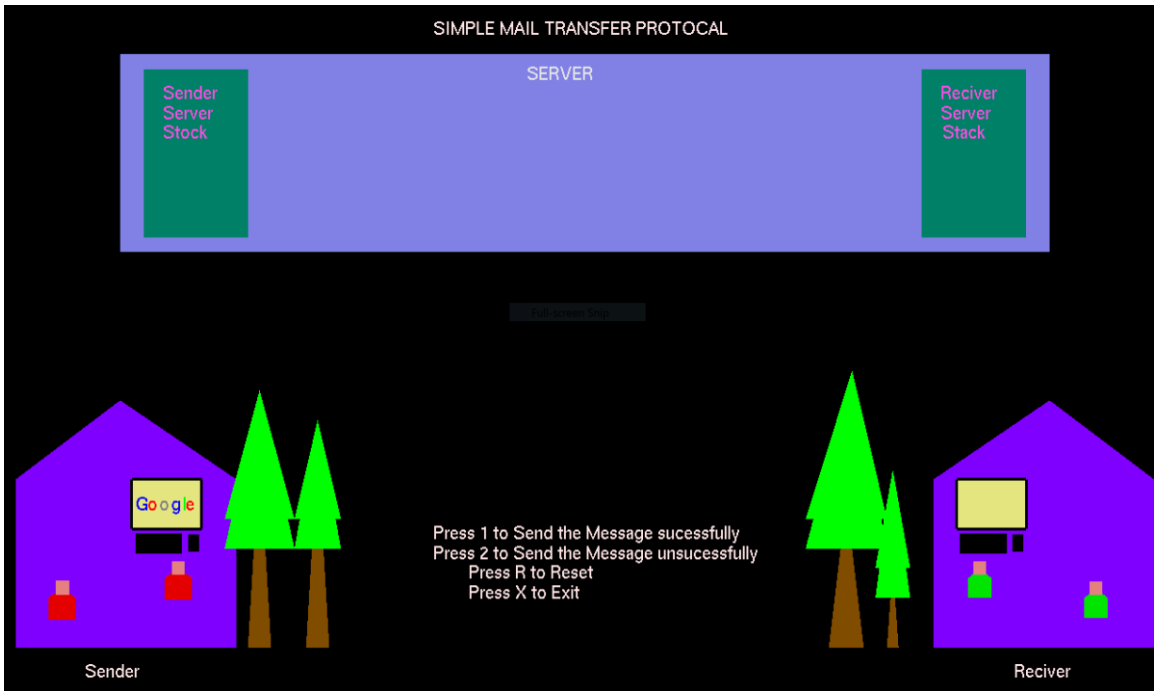


Fig 4. 2: Front View

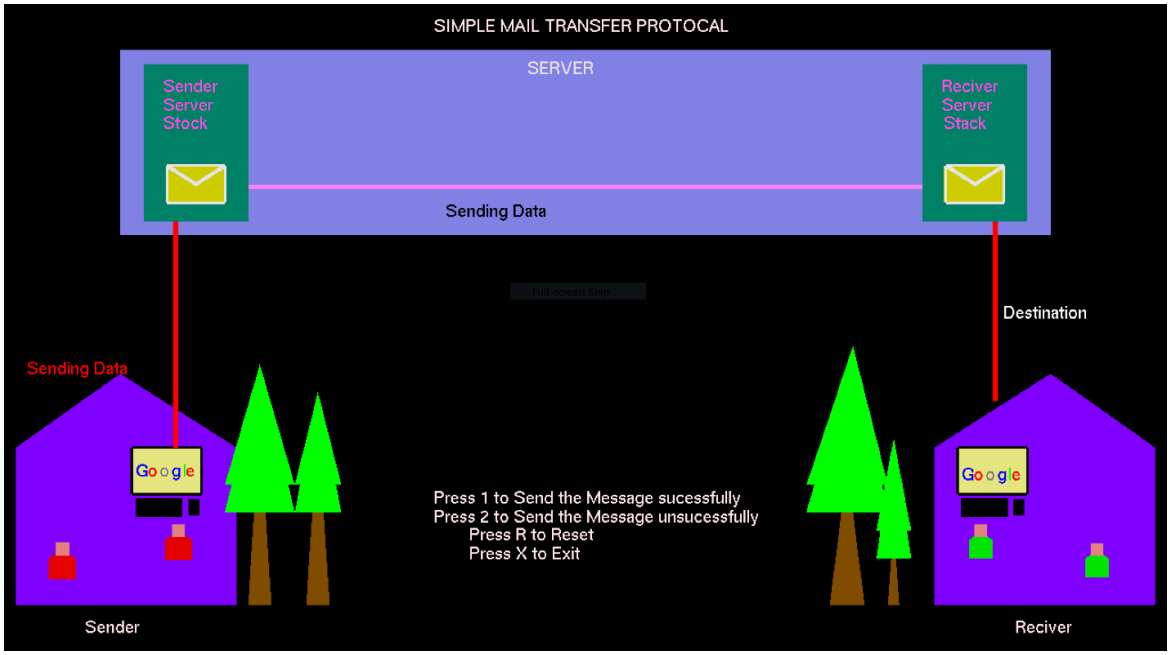


Fig 4. 3: Sending of Message

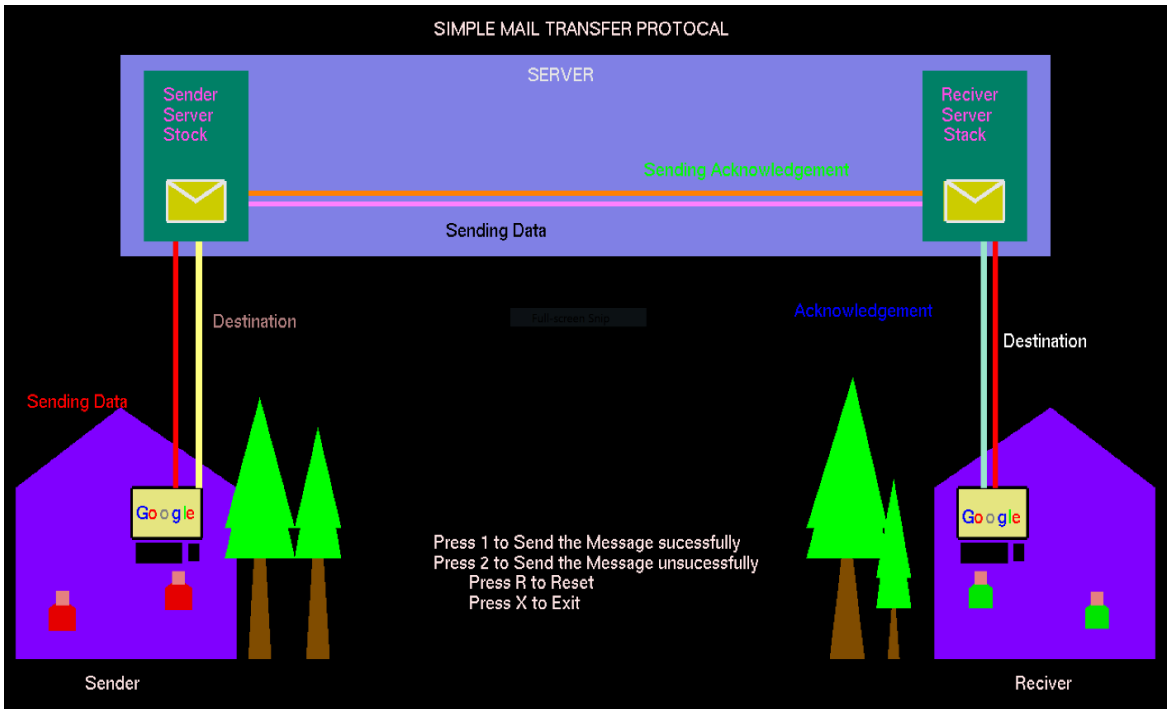


Fig 4. 4: Receiving of Acknowledgement

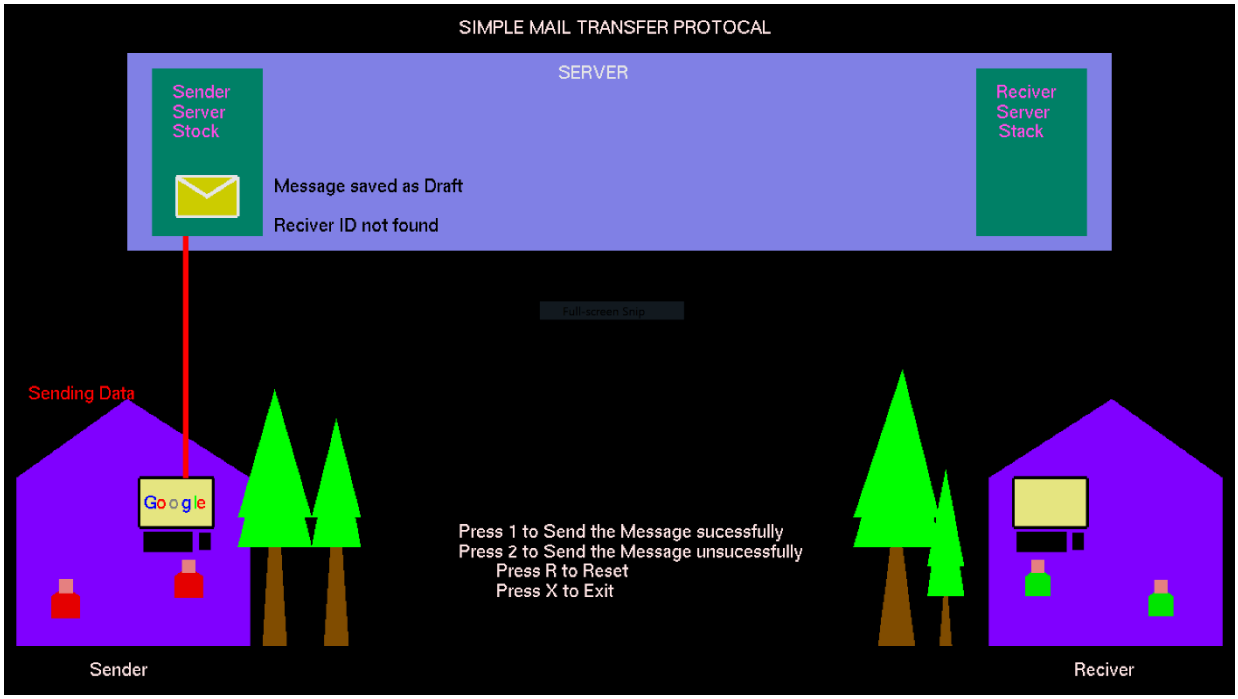


Fig 4. 5: Sending Message Unsuccessful

CONCLUSION

E-mail system uses the transaction protocols to send and retrieve E-mail messages over the local network and Internet networks. This paper proposed an active E-mail system SMTP protocol monitoring algorithm using the active monitoring technique, to provide more efficient tools which are able to monitor the E-mail system protocol in real-working environment and detect any problems related to the SMTP protocol. The paper also tested out the proposed algorithms and presented the output of the experiments on that algorithm. It is one of the most widely used and implemented application. With the explosively growing reliance on electronic mail for commercial and personal services, there grows the demand of authentication and confidentiality. To complement the weak security feature of SMTP industry use PGP-SMIME-PEM. Still there is need of implementing the measures to eliminate spam and other security breaches.

REFERENCES

[1]. Interactive computer Graphics: A Simple Mail Transfer Protocol with OpenGL- Edward Angel, 5 edition, Addison-Wellesley, 2008.

[2]. Online tutorials for OpenGL preparation on project Worms.

[3]. OpenGL Redbook and Bluebook for reference.

[4]. www.opengl.org for OpenGL tutorials.

[5]. www.stackoverflow.com