
Mail Server (Emulator)

S Satyanarayana

August 21, 2023

Table of Contents

Abstract	3
Project Specifications.....	4
Definitions, Acronyms and Abbreviations.....	5
File Structures.....	6
System Requirements.....	7
Usecase model.....	8
Function Prototypes.....	9
Project Screenshots.....	10
Code Repository.....	13
Future Enhancements.....	14
Bibilography.....	15

Abstract

Mail Server(Emulator) works like browser based(GUI) email service but this emulator works on Command Line Interface(CLI). All basic functionalities are provided. In this project we are emulating an emailing service. You will create and delete users, present them with their inbox, and send emails back and forth. This project was done at end of 4th semester for the better understanding of arrays and file structures. Most of the project is designed using modularity principles and implemented user defined functions.

Project Specifications

I did this project to recollect my skillsets in Programming and Data Structures(Semester-I). As I am using email mail services on the Internet and very enthusiastic to understand inside functionality of email service. After understanding the internal functionality and I planned to build the following features:

- Creating and deleting user by the server admin
- User credentials are stored and referred for later use
- Sending initial service mail or welcome mail
- Each mail stored as separate entity
- Use well formatted outputs

I have some time constraints during planning and development of this project. In the break semester summer holidays, I have various aspects to focus like competitive programming and others. And in this short period of time like one month it was planned and developed.

The deliverable components of this project is very simple. It does not need any additional configuration. The executable file component is enough to launch the application.

Definitions, Acronyms and Abbreviations

File: File is storage clause to store the information. All files in this project are text formatted. To identify each entity in the used various delimiting character like #.

User: Initially mail server starts with new user creation then user can start sending/receiving mail

Credentials: Entire user session flow goes with user name and password. These are stored in a file. Each user has separate file for his/her credentials.

Inbox: All incoming mails are available in inbox storage. Each user is associated with separate inbox file entity.

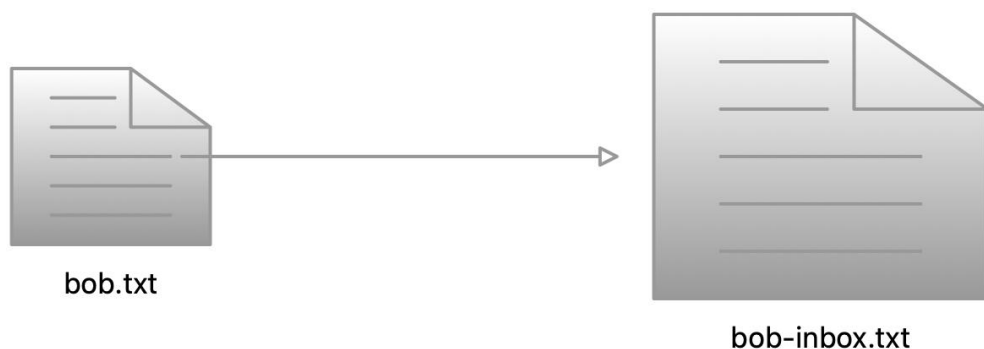
Signup/Signin: Signup is registration of new user and afterwards user can login and start sending or reading of mails.

File Structures

I used simple record classification model using sequential file system in this project. Registered user credentials are stored in master file called <user_name>.txt and each login attempt this user compares the existing credential data.

All mails are stored in another file <user_name>-inbox.txt after registration of the user first welcome service mail is stored in this file. Each of the inbox mail is separated a delimiting field to traverse the records.

Example: a user-name “bob” file structure



System Requirements

Functional requirements:

- Place all the application files in a specific folder
- Run the application and choose the options in Command Line Interface
- Required any text editor to read the text files content

Non-functional requirements:

(a) Hardware requirements

- Basic computer system with 4gb RAM
- CPU with 2ghz or higher

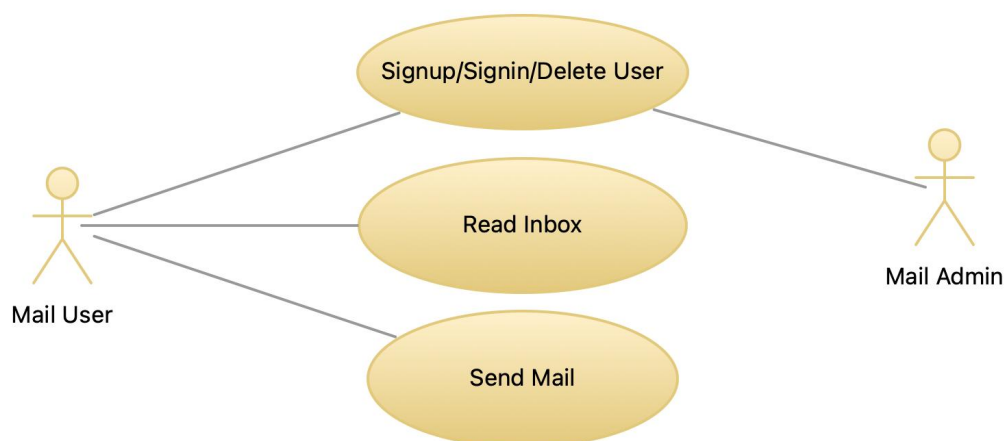
(b) Software requirements

- Any operating system of 64bit is recommended
- gcc/g++ or turbo c++ compiler is required

Usecase Model

The actor mail-user can create new mail account. Then login with credentials provided. And any time he/she can send or receive the mails.

The actor mail-admin who is super user and having authority to delete other users. And super admin also can use the mail service like other mail users.



Function Prototypes

void MainMenu(): It popups first list of menu choices for sign-in/signup and etc.,.

void RegisterNewUser(): Creates new user with credentials given.

void Login(): Authenticates the user credentials.

void LoggedInMenu(string userName): Shows the inbox of the current user and also allows other operations.

void DeleteUser(): Deletes the specific user account and inbox also.

string GetUserFile(string userName): Gets user filename.

string GetPassword(string userName): Gets the password from user credentials.

void ShowInbox(string userName): Shows all the inbox contents of the user.

string GetInboxFile(string userName): Gets the inbox filename.

int GetInboxCount(string userName): Returns the number of mails in the inbox.

string FormatEmailString(string from, string to, string subject, string msg):
Returns output formatting for the mail.

void SendEmail(string userName): Sends the mails to the specific user, if exists.

bool DoesUserExist(string userName): Checks the existence of the user.

Project Screenshots

User Signup and Signin

```
1) Login
2) Register new user
3) Delete user
99) Exit
Selection: 1
Enter username: sampreeth
User does not exist!
1) Login
2) Register new user
3) Delete user
99) Exit
Selection: 2
Enter new username: sampreeth
Enter new password: nanu2004
User created successfully!!
1) Login
2) Register new user
3) Delete user
99) Exit
Selection: 1
Enter username: sampreeth
Enter password: nanu2004
1) Inbox[1]
2) Send Email
99) Logout
Selection: 1

From: mailservice
To: sampreeth
Subject: Introduction
Message: Hello, sampreeth! Welcome to our service

1) Inbox[1]
2) Send Email
99) Logout
Selection: 99
```

Sending Mail

```
1) Login
2) Register new user
3) Delete user
99) Exit
Selection: 1
Enter username: sampreeth
Enter password: nanu2004
1) Inbox[1]
2) Send Email
99) Logout
Selection: 2
Send email to: ravi
Subject: testing mail
Message: Hi, Ravi how are you. How is your studies going on
Sending:
#email
sampreeth
ravi
testing mail
Hi, Ravi how are you. How is your studies going on
Email sent!
1) Inbox[1]
2) Send Email
99) Logout
Selection: 99
Logging out!!
```

View the Inbox

```
1) Login
2) Register new user
3) Delete user
99) Exit
Selection: 1
Enter username: ravi
Enter password: ravi123
1) Inbox[2]
2) Send Email
99) Logout
Selection: 1

From: mailservice
To: ravi
Subject: Introduction
Message: Hello, ravi! Welcome to our service

From: sampreeth
To: ravi
Subject: testing mail
Message: Hi, Ravi how are you. How is your studies going on

1) Inbox[2]
2) Send Email
99) Logout
Selection: █
```

Delete the user

```
1) Login
2) Register new user
3) Delete user
99) Exit
Selection: 3
Enter the username of the account to delete: satyacom
User removed successfully!
1) Login
2) Register new user
3) Delete user
99) Exit
Selection: 1
Enter username: satyacom
User does not exist!
1) Login
2) Register new user
3) Delete user
99) Exit
Selection: █
```

Code Repository

[1] https://github.com/satyacomp/Mail-Server_Emulator/

Future Enhancements

- 1) No echo in password inputs
- 2) Encrypt/decrypt the credentials
- 3) Password reset
- 4) Show the unread mails count as separate

Bibliography

- [1] <https://www.geeksforgeeks.org/file-handling-c-classes/>