software requirement specification

1. What is this program for?

* This program is an implementation of *chat application* and *mail system* using ***C*** to engage in private and public conversations.
* It is an ***intranet***-based *chat application* with no internet connection requirement.
* It uses secure ***end-to-end encryption***, encrypting *mail* and *messages* before sending out to the server.
* Interactive *mail system* and *chat application* with ***FTP*** support.

1. Why this program as a project?

* It is simple, beginner friendly, and provides complete autonomy for the users.
* Implements end-to-end encryption for the security of the user.

1. Why only this program?

* This program uses existing software within an operating system with no extra requirements.
* Ease of use and seamless communication within an organisation.
* Requires no internet connection.

1. Why should I use it?

|  |  |  |  |
| --- | --- | --- | --- |
| FEATURES | CRYPT-CONVO | LMS | SQIGGLE |

|  |  |  |  |
| --- | --- | --- | --- |
| Works without internet |  |  |  |
| Group Messaging |  |  |  |
| Encryption of Messages |  |  |  |
| Digital Signature Authentication |  |  |  |
| Crash  Recovery |  |  |  |
| Message Backup & Restore |  |  |  |
| Cross  Platform |  |  |  |
| Offline Message Delivery |  |  |  |

1. What should I have to have to use it?

* **Hardware**
* RAM : 256 MB
* HDD Space : 100 MB
* Connection Type : LAN (Intranet)
* Connection Speed : 64 Kbps
* Processor Speed : 300 MHz
* Others : Keyboard
* **Software**
* Operating System : Windows XP or higher, Linux 12.04 or higher
* File System (Hierarchal)
* C Compiler

1. What rules are there if I use this?

* **UI**
* Very simple, easy to be use, beginner-friendly, and intuitive UI.
* Simple, Non-disturbing ***popup messages*** notifies user of any important notifications.
* Data is encrypted and backed up onto the server preventing accidental data loss.
* Data is automatically encrypted before sending and automatically decrypted after receiving mail/message.
* **Verification**
* The source user is notified when the destination user receives the message/mail.
* Only registered members of the organisation can use the application.
* Upon the arrival of message/mail, the application checks for the digital signature and notifies in case of tampering.
* **Validation**
* During *sign up* of a new member, the application checks for appropriateness for that particular field (mail id, name, DOB, passwords, etc.).
* At the time of *sign in*, application checks for username and password match against database before allowing the member access to his account.
* *Spell Checker* feature checks for any typos in mail/messages before sending.

1. How often does this program crash? How do I fix it?

* The program is run around 400 times on various systems and failed for 10 times.
* Probability of failure : *0.025.*
* In case of failure, a simple restart would fix it.

1. List Do’s and Don’ts

* **Do’s**
* User has to sign up before they can start mailing/chatting.
* Tick mark next to a field during sign up indicates that the field fulfils its appropriateness criteria.
* Do check for the digital signature of the mail/message after receiving.
* **Don’ts**
* Don’t use short and guessable passwords.
* Don’t spam other users.
* Don’t tick the ***remember me*** option on a system which you don’t use regularly.