# COMP3334 Computer Systems Security

# Group Project

**Group Number:**  9

|  |
| --- |
| Briann DJOJOMITRO  17086405d |
| Vincent DJOJOMITRO  16094577d |
| LAW Yin Chun  17048449d |
| TUET Ka Ho  17052432d |
| YAU Yuet Man  17056093d |

## Index:

a. Background

b. Security requirement analysis with justifications

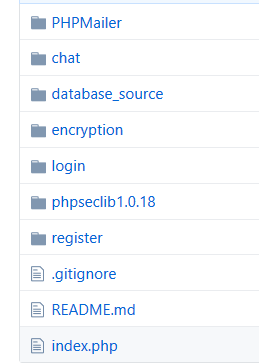
c. Design specification

d. Bibliography

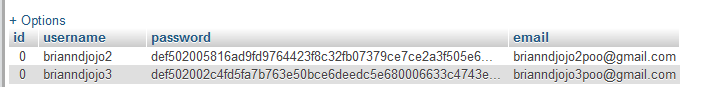
## a. Background

We implemented web-based instant messaging on local hosted Apache and Mysql servers by XAMPP.

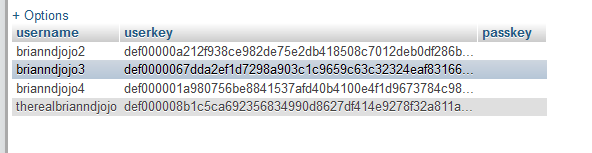
### File Specification:



* The folder, "database\_source" contains the file for importing the database (not the database itself).
  + There are two databases: ‘comp3334’ and ‘comp3334\_userkeys’.
  + 'comp3334’ stores the user’s encrypted information, while ‘comp3334\_userkeys’ stores the key that encrypted the user’s password.
  + No data of actual users will be present in this folder. Data that is present are used as dummy variables to give you an example of what the overall database structure looks like.



comp3334 database holds user information, most importantly encrypted password.



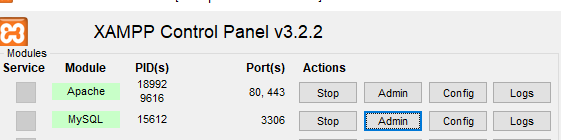
comp3334\_userkeys database holds ENCRYPTED key for decrypting his/her password.

(The encryption was done using AES.) The open-source AES algorithms are in the ‘encryption folder’. Library used for AES: https://github.com/defuse/php-encryption

* The folder, "chat" contains the chat application itself.
  + Used Socket.io & Node.js
  + Video Reference & Example of installing and implementing Socket.io & Node.js: <https://www.youtube.com/watch?v=rxzOqP9YwmM>
* The folder, “encryption” is the library of AES encryption algorithm.
* The folder, “login” is for user login.
* The folder, “register” is for user registration
* The folder, “PHPMailer” is for confirmation of user registration. The mailer support PHP5 version or above.
* The folder, “verify” is for email verification
* The folder, “conf.rar” is for config on apache.

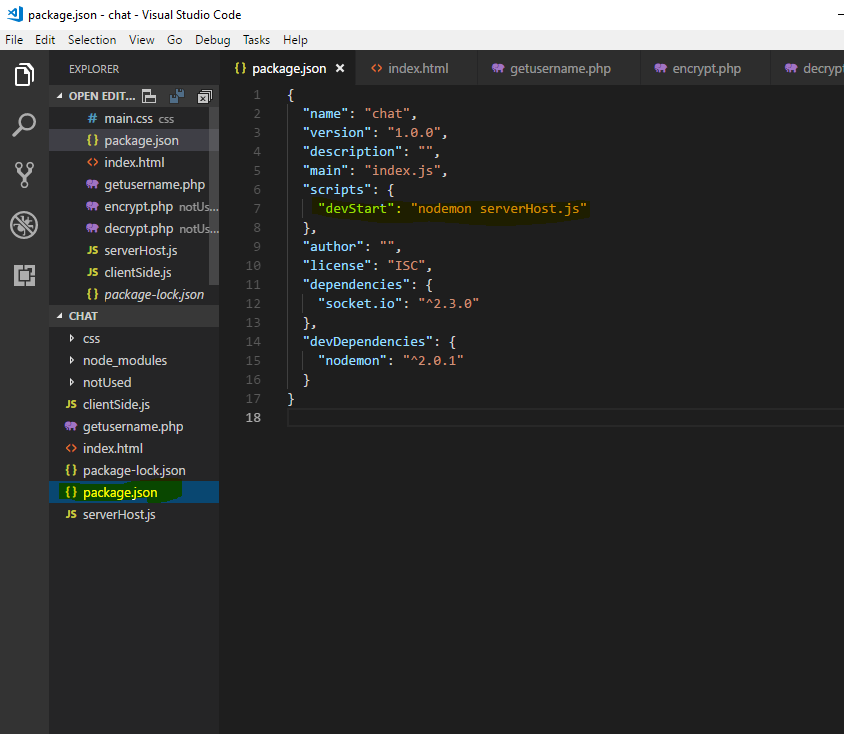
### Environment setup:

1. Set up Localhost. The project was developed on XAMPP version 3.2.2, which can be download though searching “xampp-win32-5.6.38-0-VC11-installer". Other versions might not be working properly. Download xampp 5.6.38 here: <https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/5.6.38/> .
   1. Copy all files to your directory. E.g. ‘C:\Xampp3\htdocs’
   2. After install xampp, start Apache server.
   3. Note: the xampp folder should be named, ‘Xampp3’ and we use the following ports for this project.

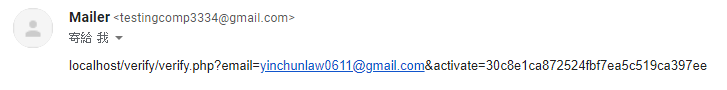


1. Set up SSL.
   1. Unzip “conf.rar”, copy to replace the file “C:\Xampp3\apache\conf”.
2. Set up the database.
   1. Visit the MySQL server directory e.g. “C:\Xampp3\mysql\data”.
   2. Create 2 folders: ‘comp3334’ and ‘comp3334\_userkeys’
   3. Start MySQL server and visit phpmyadmin by pushing “Admin” button in XAMPP
   4. Insert the databases from ‘database\_source’ folder. Insert ‘comp3334.sql’ to ‘comp3334’ folder and ‘comp3334\_userkeys.sql’ to ‘comp3334\_userkeys’ folder.
3. For instant chatting,
   1. It requires Node.js version 12.13.1x64. Install “[node-v12.13.1-x64.msi](https://nodejs.org/dist/v12.13.1/node-v12.13.1-x64.msi)".
   2. The chatting also requires Socket.io. Install it by the following commands:
      1. Set Visual Studio Code’s integrated terminal to Git Bash
      2. Set the directory of the intergrated terminal to ‘Current Project’
      3. Initialize npm by typing “npm init” in the terminal
      4. Install socket.io by typing “npm i socket.io” in the terminal
      5. Use nodemon to keep the server refresh by typing “npm i --save-dev nodemon” in the terminal
      6. Start Socket.io by typing in “npm run devStart” in the same terminal.

These following commands must be run **in the chat directory!**

* + - 
    - 
    - 
    - Then include, “devStart”: “nodemon serverHost.js” in the package.json file.
    - 

For **exact explanations for installing Node.js & Socket.io** follow this video from **0:00 to 3:35**. <https://www.youtube.com/watch?v=rxzOqP9YwmM>

1. Start chatting.
   1. Open two tabs in Chrome. Type ‘localhost’ in search engine, press enter.
   2. Neglect the security warning if you receive one. \*
   3. After loading into login pages, register two accounts and login to each tab.
   4. Receiving confirmation email
   5. Activate the account by accessing the link given in email.
   6. Login to the activated account.
   7. Start chatting.

\*SSL certificate should be obtained by a real company. Right here, we generated a self-signed SSL certificate instead. The browser detects the generated cert and warn users the website is not secure. Therefore, we can neglect the warning here. The website is safe to enter.

**OVERALL REPRESNTATION OF HOW THE CHAT IS SUPPOSED TO WORK**

**Video-Explanation of how to use chat**: <https://www.youtube.com/watch?v=YfW5oMQsQsk>

**Video Explanation of how to use REGISTER**: <https://www.youtube.com/watch?v=YQ0aoHXGyDs&feature=youtu.be>

## b. Security requirement analysis with justifications

|  |  |
| --- | --- |
| **Item** | **Requirements** |
| Confidentiality | Chat message should not be revealed to unauthorized users (users who do not get involved in the chat). |
| User information (password, email, etc.) should be hidden from other users. |
| Integrity | Messages sent should keep its originality. |
| Messages sent should keep its correctness. |
| Availability | User should always able to access the server. |
| Authentication | An account should only be accessed by the user himself/herself. Verification should be done before user can use the chat |

## c. Design specification

**Security design**

|  |  |
| --- | --- |
| **Possible Problem/Security measure** | **Solution/ Security design taken** |
| Wire tapping | No chat message history will be stored so attackers would not able to retrieve chat messages from any databases. |
| Secure Socket Layer (SSL) is used to prevent Man in the Middle (MITM) attacks. SSL provides privacy, security and data integrity on websites and user information thus it is very important that it is implemented. SSL encrypts sensitive information when sensitive information is sent over the internet thus providing confidentiality and privacy. SSL also provides authentication meaning that data will always be of integrity. Because SSL provides authentication, MITM attacks become obsolete when SSL is deployed as impersonating people would become hard due to authentication. |
| Password theft | Instead of storing passwords, databases store the encrypted password. |
| Man in the middle attack | Secure Socket Layer (SSL) is used to prevent Man in the Middle (MITM) attacks. SSL provides privacy, security and data integrity on websites and user information thus it is very important that it is implemented. SSL encrypts sensitive information when sensitive information is sent over the internet thus providing confidentiality and privacy. SSL also provides authentication meaning that data will always be of integrity. Because SSL provides authentication, MITM attacks become obsolete when SSL is deployed as impersonating people would become hard due to authentication. (HTTPS) |
| Correctness of chat messages | SSL is used to prevent Man in the middle thus ensuring correctness of chat messages. |
| Backdoor | There is no backdoor, all users should login using their own set of username and password. Such that exposure of one’s account information would not affect the others. |
| SQL injection | PHP function is used to escape parameters before sending SQL query. |
| Cryptography | AES encryption is used to encrypt the stored password |
| Access Control | For registration, users require an account, password and email address. The database will check whether the information is repeated with the others to confirm the identity. |

**Data design**

There are two databases used: comp3334, comp3334\_userkeys. Their structure is as follow:

comp3334

|  |  |  |  |
| --- | --- | --- | --- |
| **Value** | **Type** | **Description** | **Example** |
| id | int | An integer value assigned to identify the user | 0 |
| username | varchar(100) | User name of the user, used to identify themselves to other users in chat | Bob |
| password | varchar(8000) | Hashed value of password | def502005816ad9fd9764423f8c32fb07379ce7ce2a3f505e6593a63bf7221e447999f7550117a6108b07483162e2eefa3f80aeee2684d7d89ef27f1960a8978d952b7755153698b105bffc29b39c42db6bc85dd0a1145d46fd543 |
| email | varchar(255) | Contact information of use, used in registration and other contacts upon needs | [Bob@hello.com](mailto:Bob@hello.com) |
| activate | varchar(32) | Random string of length 32 used to activate the account It will be set to ‘1’ after successfully activated. | 81e5f81db77c596492e6f1a5a792ed53 |

comp3334\_userkeys

|  |  |  |  |
| --- | --- | --- | --- |
| **Value** | **Type** | **Description** | **Example** |
| username | varchar(255) | Username of the user, used to identify themselves to other users in chat. | Bob |
| userkey | varchar(8000) | User RSA key of the user, used with another database(comp3334) | def0000067dda2ef1d7298a903c1c9659c63c32324eaf831665217797457d748445d7f0bdce0009e9b024072f9bb2c8d300a4787d379349758f5901bf907d511911abe29 |
| passkey | varchar(255) | User RSA key of the user, used with another database(comp3334), indicate for passing key | def0000067dda2ef1d7298a903c1c9659c63c32324eaf831665217797457d748445d7f0bdce0009e9b024072f9bb2c8d300a4787d379349758f5901bf907d511911abe29 |

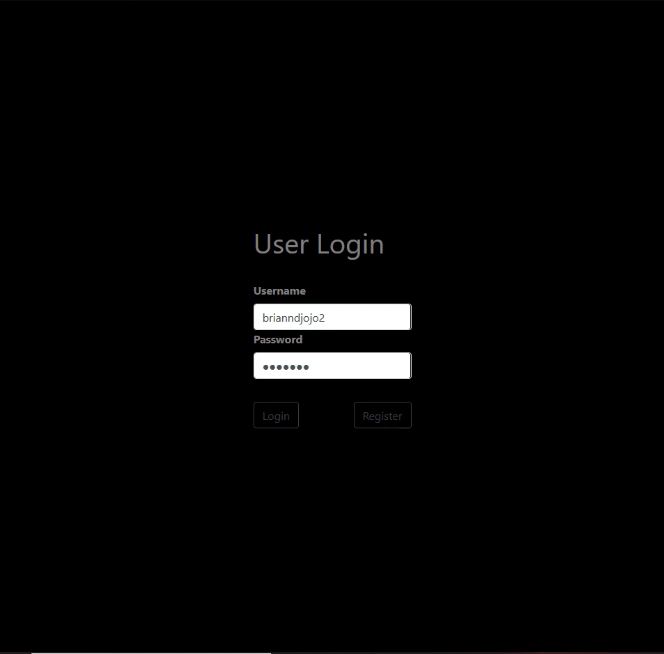
**Interface design**

Login page:

Two text fields for inputting username and password respectively.

A button call function verify the inputted value of the two text fields.

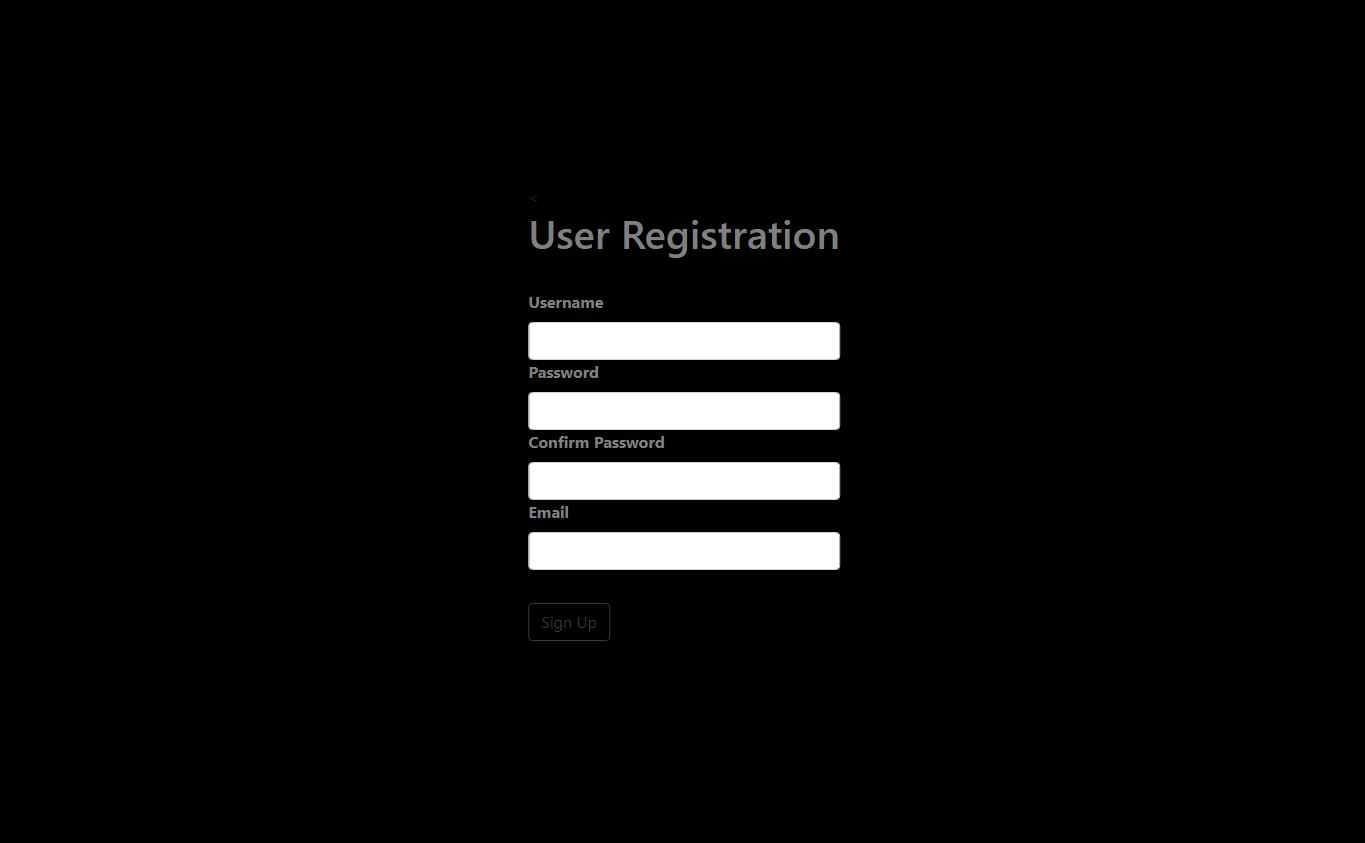
A button redirect user to register page.



Register page:

Four text fields for inputting username, password, email respectively.

A button call function to process the inputted value and do registration.

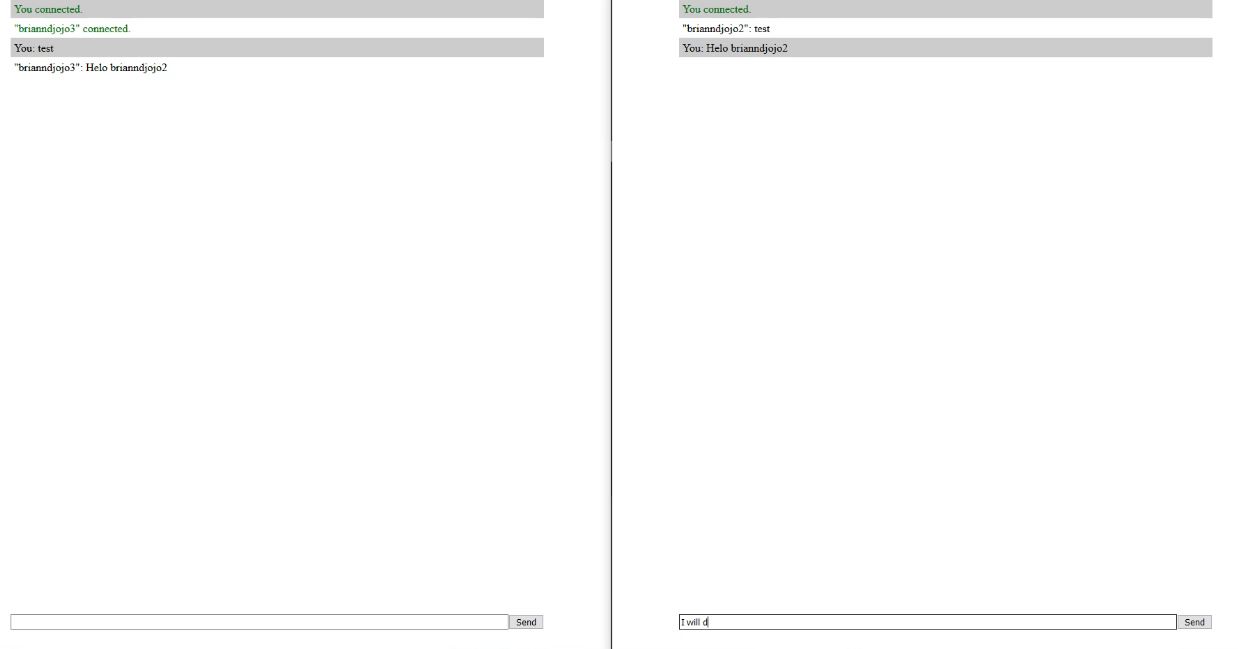


Chat interface:

A message part show system messages (e.g. notification of user join) and chat messages (indicates who sent the message by showing username).

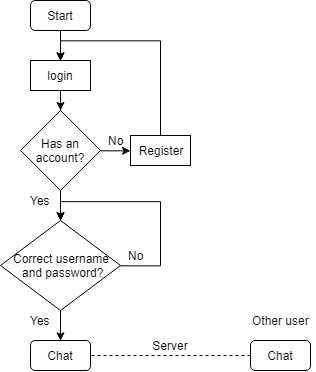
A text field for users to input message

A button for users to send inputted message



**Procedural design**

Users first login (if they don’t have an account, register new account). After verifying their username and password, user is then connected to the server and start chat with another user.



## d. Bibliography

[1] Web Dev Simplified, Build a Real Time Chat App With Node.js And Socket.io, May. 25, 2019. [Video file]. Available: https://www.youtube.com/watch?v=rxzOqP9YwmM [Accessed Dec. 8, 2019]

[2] Why SSL? The Purpose of using SSL Certificates, 2019. Available: <https://www.sslshopper.com/why-ssl-the-purpose-of-using-ssl-certificates.html>

[3] User-made libraries: Credits:

* Defuse (for AES encryption)
  + Source: <https://github.com/defuse/php-encryption>
  + Authors: Created by Taylor Hornby and Scott Arciszewski.

# Peer Evaluation Form

**Group Number:**  9

**Name:**

**Student No:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Group Members**  **(Name and Student Number)** | **Effectiveness\*** | | | | |
|  | **Not at all** | **Poorly** | **Adequately** | **Well** | **Extremely**  **Well** |
| <Me> |  |  |  |  |  |
| Briann DJOJOMITRO  17086405d |  |  |  |  |  |
| Vincent DJOJOMITRO  16094577d |  |  |  |  |  |
| LAW Yin Chun  17048449d |  |  |  |  |  |
| TUET Ka Ho  17052432d |  |  |  |  |  |
| YAU Yuet Man  17056093d |  |  |  |  |  |

\* Put  in appropriate boxes

**Additional Comments:**