

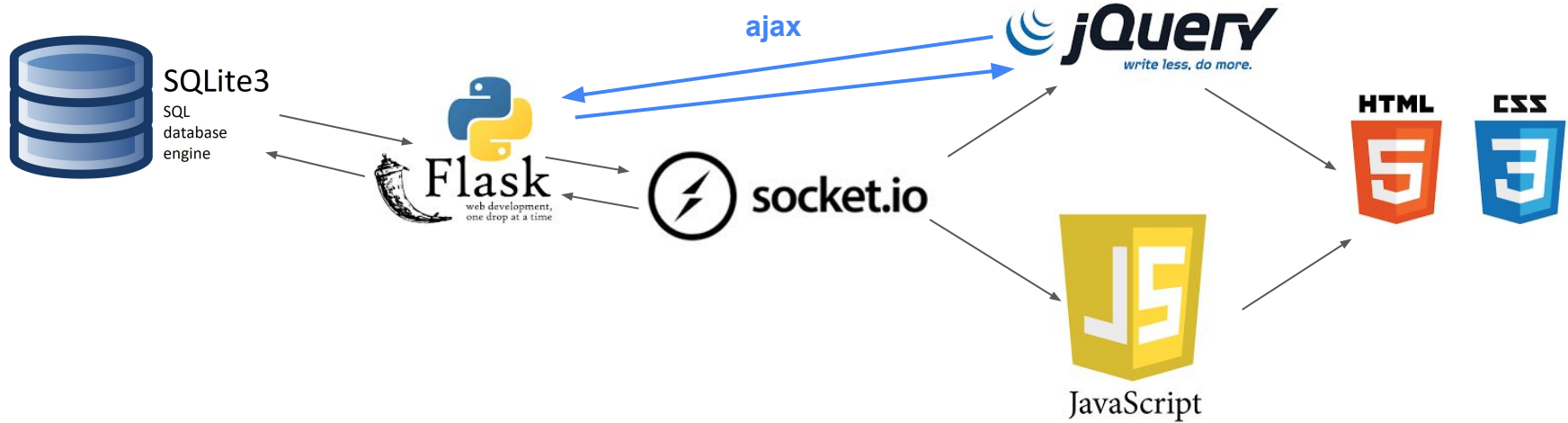
Computer Networking Chat App Project

Gavin Jin & Alan Chao

Tools Used

Server:

Client:

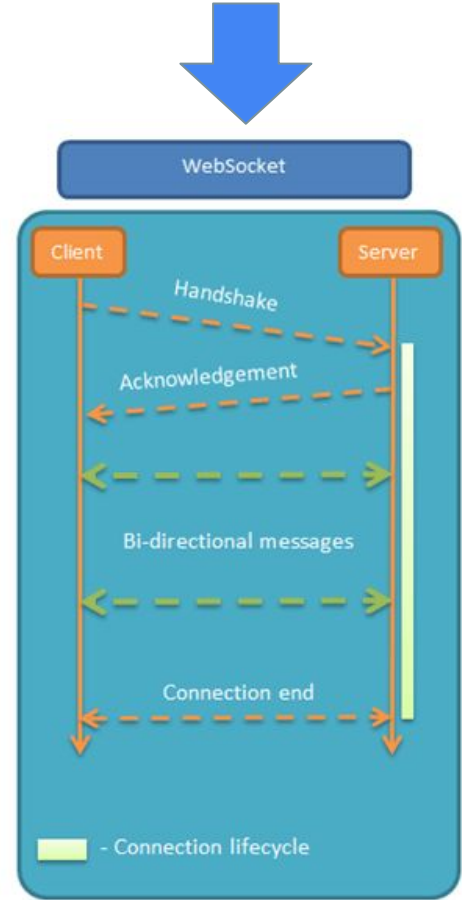
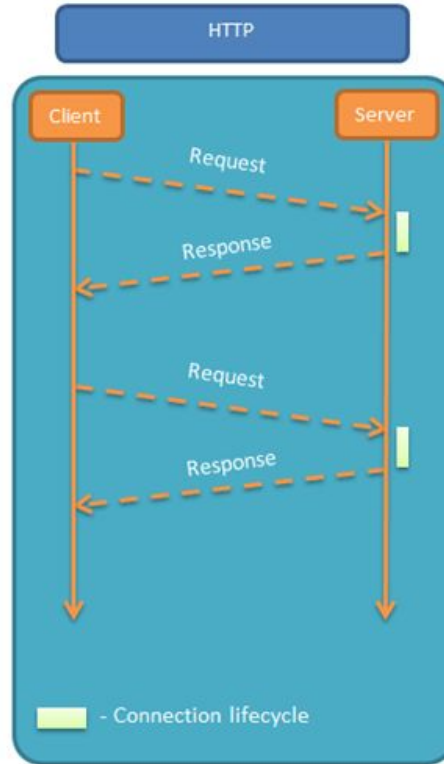


Message Connection

Socket.io



Based on the usage of **WebSocket**:
bi-directional message, good for
building real-time chat applications



Socket Implementation

Client

```
$(document).ready(function () {  
    socket = io.connect('http://127.0.0.1:5000');
```

```
const meInfo = {  
    "Type": "Send", "From": currentUserNane,  
    "To": currentThread.name, "Content": msg,  
    "Chat": currentThread.type, "is_image": 0  
};  
  
socket.send(JSON.stringify(meInfo));
```

Server

```
if __name__ == "__main__":  
    socket.run(app, host="127.0.0.1",  
              port=5000, debug=True)
```

```
@socket.on("message")  
def handle_message(msg):  
    msg = json.loads(msg)
```

Client Functions

1. Register and Login
2. Get list of Users and Groups
3. Private Messaging / Group Messaging Selection
4. Creating Group
5. Joining a Group / Private chat
6. Deleting Group
7. Emoji Sending
8. Image Sending
9. Responsive Sound and Color of the UI

Server Functions

1. Check database with [password hashing](#)
2. [Return client online Users and Groups](#)
3. Update database and forward to the right destination
4. Check name and initialize database
5. Update status and send history
6. Check username and update database
7. Same as the text <-> UTF-8
8. Save as a BLOB type in database
9. Basic prevention from users' cheating behaviors

Customized Message Sending Protocol

Connect

```
{  
  
  Type: "Connect",  
  
  Id: user_socket.io_id,  
  
  Username: username  
}
```

Send

```
{  
  
  Type: "Send",  
  
  From: from_user,  
  
  To: destination,  
  
  Content: content,  
  
  Chat: "Private" | "Group",  
  
  is_image: 0 | 1  
}
```

Join

```
{  
    Type: "Join",  
    Chat: "Private" | "Group",  
    From: username,  
    To: groupname/username,  
}
```


Create

```
{  
  
    Type: "Create",  
  
    Name: group_name,  
  
    From: username,  
  
    List: [user1,user2,....]  
  
}
```

Delete

```
{  
  
    Type: "Delete",  
  
    Name: group_name,  
  
    From: username  
  
}
```

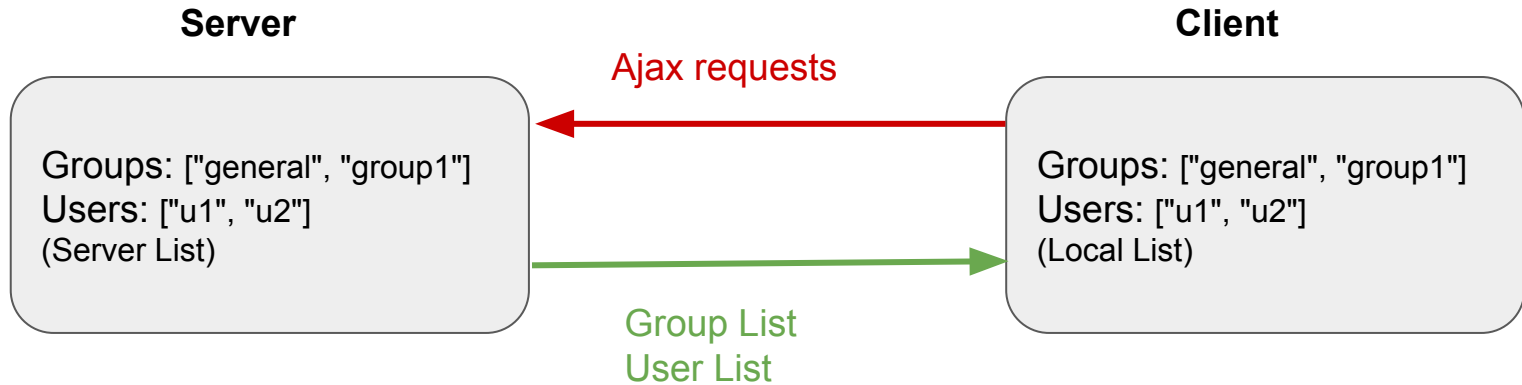
Thank You

:)

Client Side Update

Update the list of active users and groups every 5 seconds

Using **AJAX**



Example, all users register with a password:
123456.

While in database:

username	password
user1	sha256\$5l6ljtTa\$09435ed52b314e052427e373fc09686f991bc0fa747bd62a08393c2716e7bce2
user2	sha256\$1gOfE0SE\$34b1b0ef66c60a7a8c71499e21ee8bceb2ef7e3c98b5d5e34898837cc7c1b365
user3	sha256\$fkagIXN8\$b1ea26b5766d5dbcdf048087904faa8c1c14268371dcc97c993af069f0fcf327
user4	sha256\$PRH7OYwy\$4dbfa22d8b5950c846bc3ce5d40a81adea6e728e0a98c685cfa961879a2f994e