External Libraries used in team project:

flask:

* Flask
* render\_template
* request
* session
* redirect
* url\_for
* jsonify
* escape

flask\_socketio:

* SocketIO
* emit
* join\_room
* leave\_room

PIL

* Image

**flask library - version 1.0.2**

**Flask:**

Script location: flask.app

A Flask is a flask object. It implements a WSGI application and acts as the central object.

A WSGI application on server side is similar to the client request / server response in our homework.

For example:

*from HTTP response/Request.ppt,* [*1\_3\_HTTP\_Request\_Response (cse312.com)*](https://cse312.com/static_files/slides/1_3_HTTP_Request_Response.pdf)

Request: Response:

GET / HTTP/1.1 HTTP/1.1 200 OK

Host: cse312.com Content-Type: text/plain

…… Content-Length: 5

……

The flask object takes the name of module or package as parameter, then it performs the corresponding action.

In our code, we use ‘app’ as the variable name by declare a flask object by

app = Flask(\_\_name\_\_)

@app.route(rule,\*\*options) (flask.app.Flask - line 1288)

def action():

……

We mainly using this script to get clients’ request by passing URL rule and methods as arguments. The argument <rule> is the URL rule that contains path of our application content. And, the argument <options> can be the request type, it can be ‘GET’, ‘POST’, etc. This is identical to the request header parsing on server side in homework.

When the client sends a request to server, the flask object ‘app’ will parse the request. We just need to specify the URL rule and request method type in .route().

The procedure of parse request is going to look like this:

server receives request

↓

@app.route(rule, method)

↓

call function add\_url\_rule(rule, method) (line 1178) in .route()

↓

in add\_url\_rule(), pass parameters rule and method to function url\_rule\_class()

Script location: werkzeug.routing.Rule – line 650

The URL rule parses will take place in function url\_rule\_class()

If the client requested URL rule and request method type match the parameters in .route(), then it will execute the function / actions that is below @app.route(). Basically, all of our web application response that send back to clients will take place in this kind of functions.

**render\_template:**

Script location: flask.templating. - line 125