```python
from bottle import request, run, post, response
import bottle

from ServerRequestHandler import ServerRequestHandler

server_request_handler = ServerRequestHandler()


@post('/post')
def post():
    """
    Primary function called when a post request is made to the server. Firstly ensures that the user
    is logged in using the session management cookie, then passes the request to the server request
    handler.

    If the user is not logged in (according to the session management cookie), they are sent a
    response requiring a login. This redirects them within the app to the login page.
    """
    if request.get_cookie('username'):
        # User is logged in under their username, so their updates go to the correct
        # place.
        json = request.json
        if request.json['type'] == 'student.get_info' or request.json['type'] == 'student.update_location':
            json['args']['username'] = request.get_cookie('username')
        return server_request_handler.handle_request(json)

    elif request.json['type'] == 'universal.login':
        # When a user attempts to login to the system. If the login is successful, then
        # a session management cookie is granted and sent back in the header of the
        # HTTP response.
        login_attempt = server_request_handler.handle_request(request.json)
        if login_attempt['successful']:
            response.set_cookie('username', request.json['args']['username'])
        return login_attempt

    elif request.json['type'] == 'universal.create_account':
        # When a user attempts to create a new (student) account.
        return server_request_handler.handle_request(request.json)

    # If it is the MacOS application attempting to make changes, no log in is necessary.
    elif 'macos' in request.json['type']:
        json = request.json
        # Remove the macos from the request -- it will not be recognized
        if not 'get_beacons' in json['type']:
            json['type'] = request.json['type'][6:]
            print(json['type'])
        return server_request_handler.handle_request(json)

    else:
        # If the user is attempting to make a request without being logged in.
        print('dsdsf')
        return {
            'successful': False,
            'login_necessary': True,
            'reason': 'You need to login.'
        }

application = bottle.default_app()
run(application, host='172.20.10.2', port='8080')

```