**Explaination of Archetectural Structure of the Prototype:**

For the architectural structural of the prototype, the client/server architectural design is best suited. It separates pieces of the system that needs to use a particular function (the client) from parts of the system that provides those functions (the server). This allows for decoupling between the client (the user) and the server, allowing developers to advance the prototype in the future

How the Client/Server architectural structure works, goes as follows:

* The User sends a web request to the Web Server.
* The web request is sent to the python package which executes the request and retrieves the data needed from the data base and the Jinja template.
* The python package has routes which connects the modules i.e.

1. **login.py which is called by routes.py and renders template from login.html jinja template**
2. **Post.py which is called by routes.py and renders template from postView.html jinja template**
3. **Subscription.py which is called by routes.py and renders template from Subscription.html jinja template**
4. **DiscussionGroup.py which is called by routes.py and renders template from DiscussionGroup.html jinja template**
5. **UserProfile.py which is called by routes.py and renders template from UserProfile.html jinja template**

With all Jinja files placed in a template created as shown in the uml diagram

* The route.py has functions that enables the web server to provide a requested service for the user.
* The SQLAlchemy python code file, Session is established to connect with the database and represents for all objects you’ve loaded or associated with it during the lifespan. the Session includes add() method, commit() method, delete() method. add() is use to place instances in the session. commit() is use to write changes to the database. delete() is use to place an instance into the Session’s list of objects to be marked as deleted.
* Controllers consists of the python code for each of the requirements. The routes.py file imports the methods for each of the GET and POST requests that the users asks for. This is in a folder colled Controllers and placed in the apps folder.
* Flask delegates jinja to render all templates to extend the main.html, which uses a base design for all the requirements template.
* Flask delegates the backend routes between python, jinja and SQLAlchemy.
* Bootstrap is used to help build the user interface of the prototype. It uses CSS or Cascading Style Sheet to create the design of the prototype.