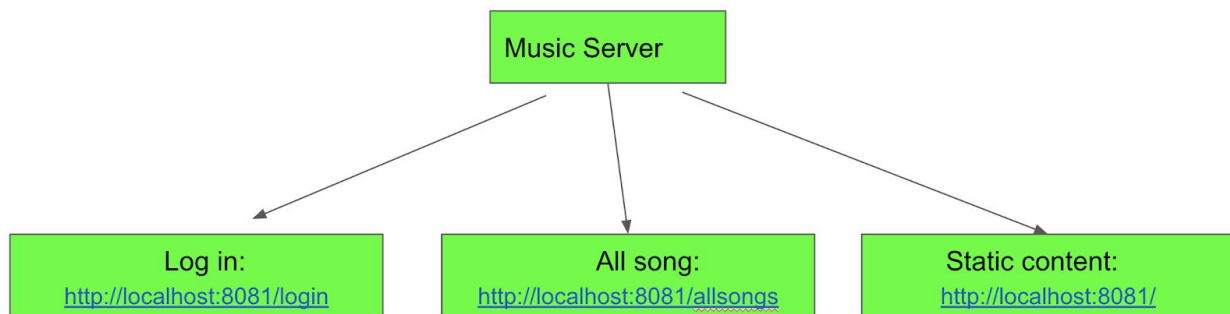


task 2: (10 points) Designing your app. In this task, you'll sketch out the components of your music player and their interaction. Create a diagram that helps you determine the answers to these questions:

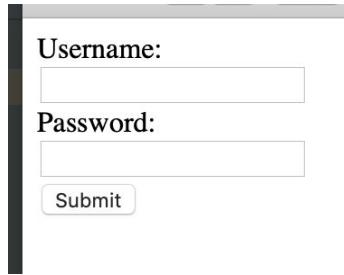
1. Name at least three tasks that your user might perform. For each task, what screens or pages will they see?
2. What is the interaction between the client and server for each task? What URL does the client use? What servlet does this map to? What does the servlet return? What happens if there is an error? What if a task requires more than one request/response?

Here are the general design:



Here are some details for each task:

Task 1: login. URL: <http://localhost:8081/login>. Attached is the screenshots for visualization.



Username:

Password:

It maps to the loginServlet. In the init stage, the server will load the user dataset which contains username and password pairs and store them in the hashmap. When the user enters username and password and presses submit, it will check the database to see if it's correct. If it's not, it will print out "no such user or user password is not correct". If it's correct, it will print out "login success" followed by the username and password (See the figure below). And then a cookie will be created for this user.



login success
meimei
0715

Task 2: allSong. URL: <http://localhost:8081/allSongs>. Attached are the screenshot for visualization.

please log in

[log in](#)


It will map to allSongsServlet. When the user sends the get all songs request to the server. The server will send back the link to ask the user to log in first if not yet. Then the user will be directed to the loginServlet. After the user log in, it will check the cookies and print out the username and all the songs in the library.

it's gege
AA Mei Mei 0 07:15
BB Mei Mei 0
gege


Task 3: serve the static content. URL: <http://localhost:8081>. Attached are the visualization images.

It maps to a resource Handler. This way, our server can handle static contents like images, file directories and files.

Directory: /

Name 	Last Modified	Size
.DS_Store	Nov 22, 2020, 11:55:39 AM	8,196 bytes
images/	Nov 22, 2020, 11:55:56 AM	96 bytes

Directory: /images/

Name 	Last Modified	Size
Parent Directory	-	-
bag.jpg	Nov 10, 2020, 5:50:03 PM	5,780,546 bytes