Data plan for website.

To start with, we can list all the possible data that stored on client side and on server side.

Let us start with the client one. It is a tricky question since we basically get everything from the server. Imagine the actions of a user. He first enters the username (var char) and password (var char) in the login page, and redirect to his user page after authorization. At this point, the only possible local data is the username and password that can be stored in cookie for later use (if the user wants to). Then the local webpage gets all the info related to that specific username from database by router. What about now? How do we store this info on client side? Do we really need so? It does make the page quicker to download next time but does it worth it? One thing for sure is we definitely not using cookie for this time. And we also got a user Icon. I previously designed this function simply because most user-info pages have something similar, but how are we going to realize it? Saving the href link in database? What if the link is not available anymore?

It is clearly that we do not have personalizing site preferences and need to save anything about it. But persisting previous site activity is big thing. It is frustrating for a user to rewrite 30 lines of personal information simply because his or her cat unplugged the battery charger. This also applied to all the other pages – register/users/admins. Any page with info-editing function should be applied.

For the server side, as we just talked about, it basically saves everything. First it should have a table saving all the username and password with a Single Column Primary Key, let us call it “user-id” (int) for now. It satisfies 2NF and help us link to other tables easier and quicker. Then we can move on to user information table. The Feature Details does not talk much about it, so does my group. I will say “name” (var char), “events” (var char), “time” (var char) applied to all three kinds of user. If we want to add extra info like “ID/passport num (int)”, “birthday” (date) that only applied to users and admins, we can create a separate table. Then it is the accessibility table or authority table. It should be Boolean with users/admins 2 sections, a value of true represent the authority of this user. We can check all 2 values and have a priority that admin > user or simply change user to false and admin to true in sign-up.

Thus, in conclusion, basically every data expects the map part is linked with the Primary Key user id. The client should pass the login authentication to access data related to the user id of that username. In our approach, we are using Ajax to send HTTP requests (XMLHttpRequest) to get the corresponding data. Sending the data to the server is another story. I have checked the piazza and find MongoDB is not allowed in this project. Data can also be updated into MySQL tables by executing SQL UPDATE statement through PHP function, but I am not sure if it is allowed. I guess we will just use POST to send information to server side and let it to make SQL queries to minimize possible errors.

For the question “What processing needs to be done to make the data useful?”, I do not really get it. I my opinion there is no need to further edit the data stored in database. It might be possible in our future coding that we use Convert function in sql to change datatype making comparison passed, but I cannot imagine the specific case here.