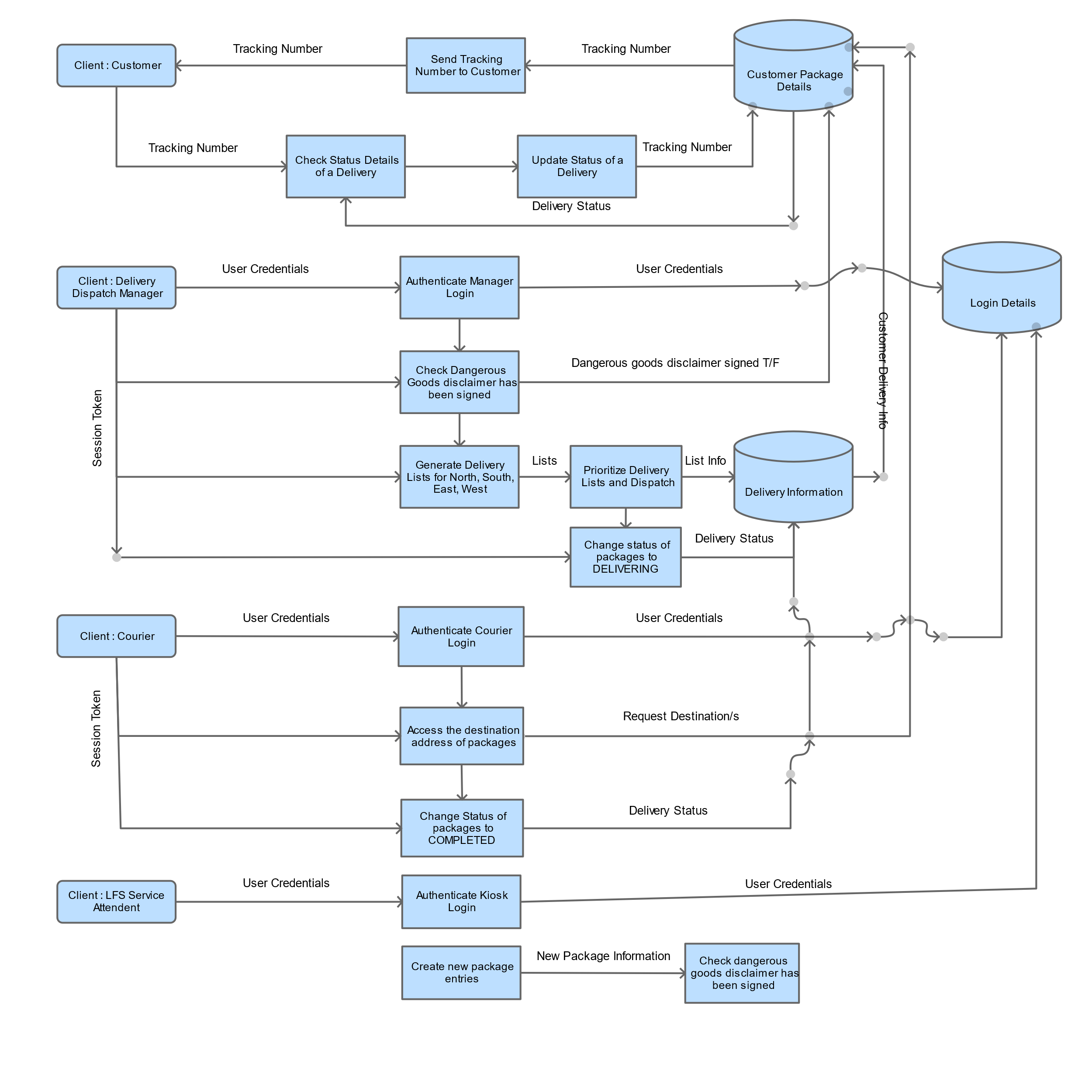
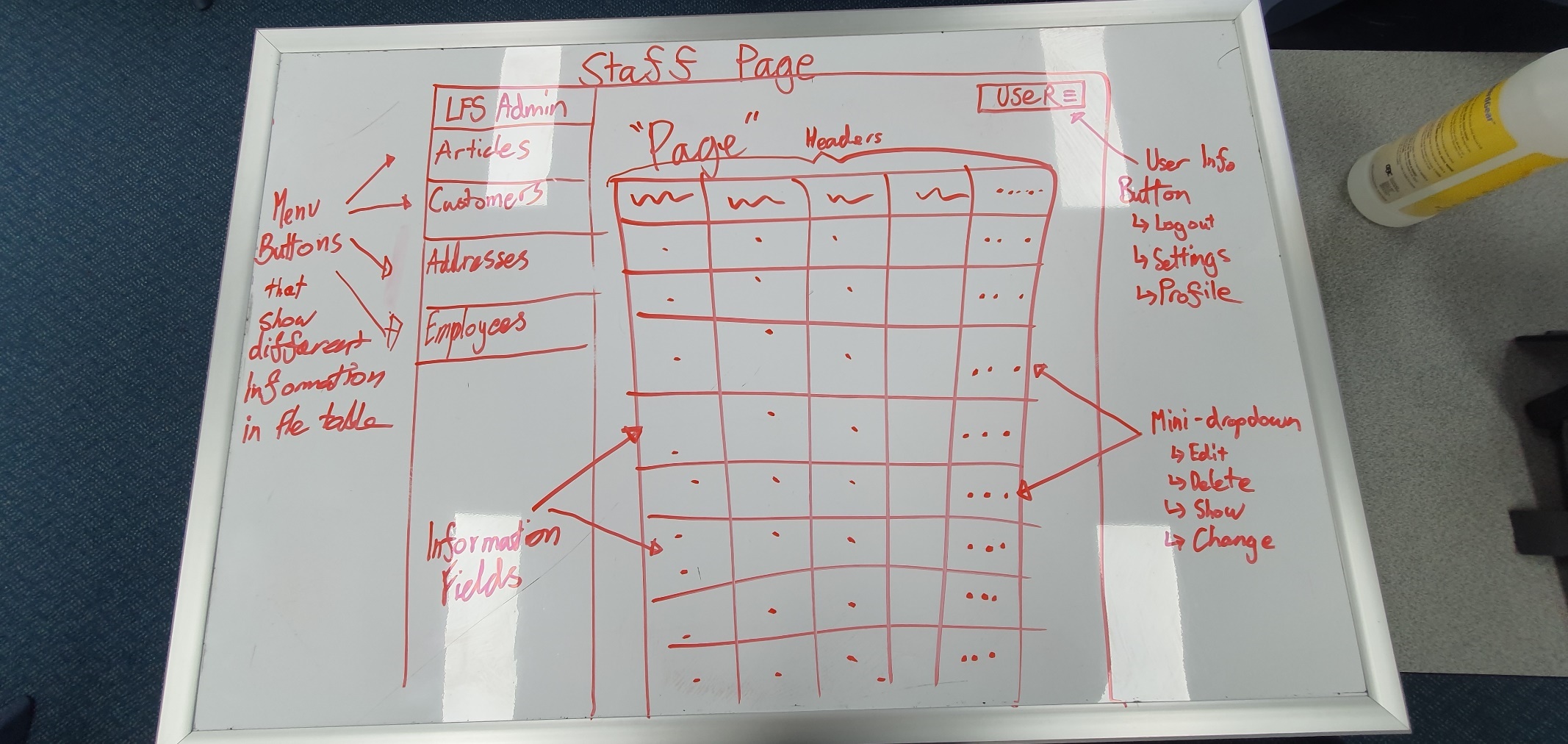
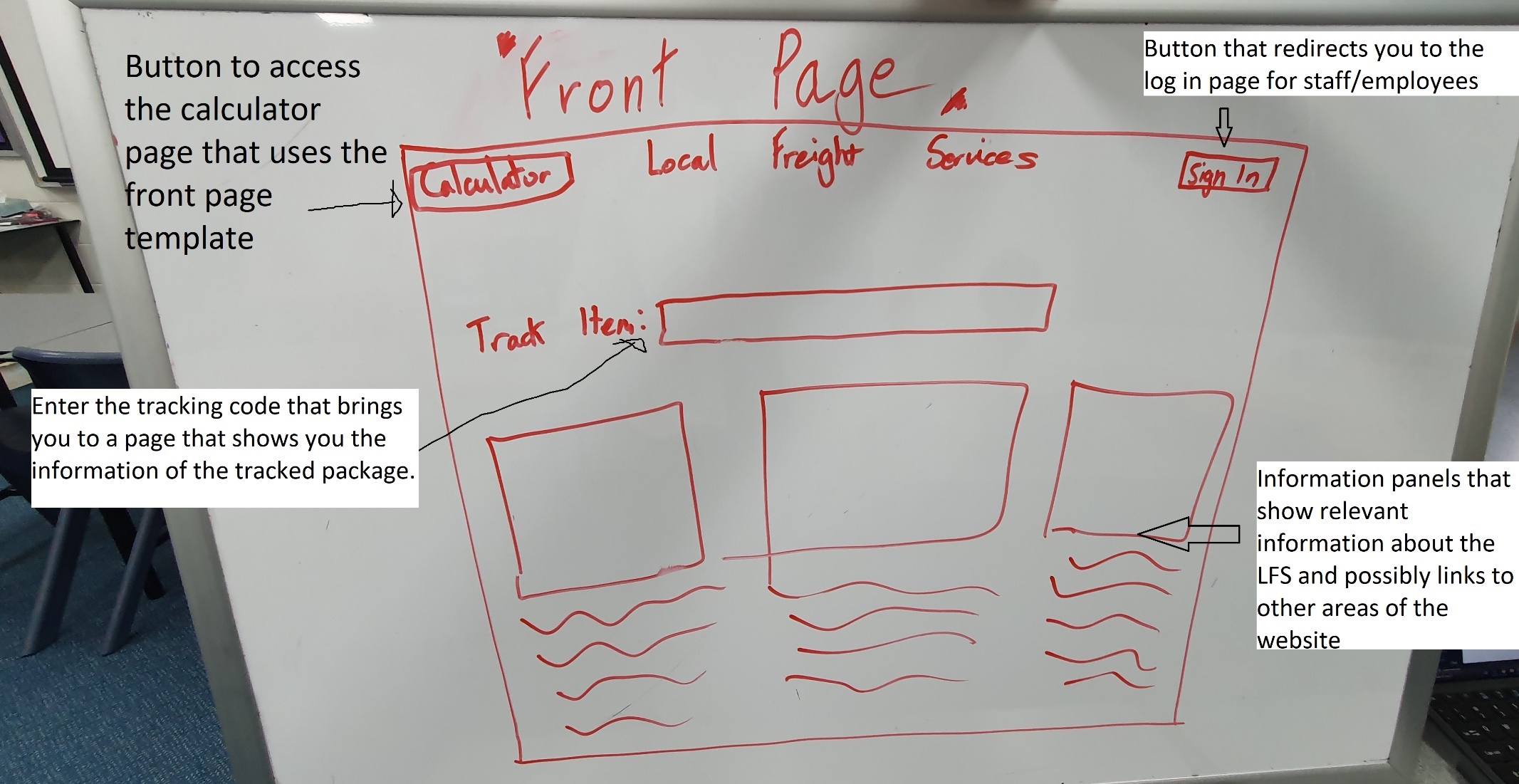
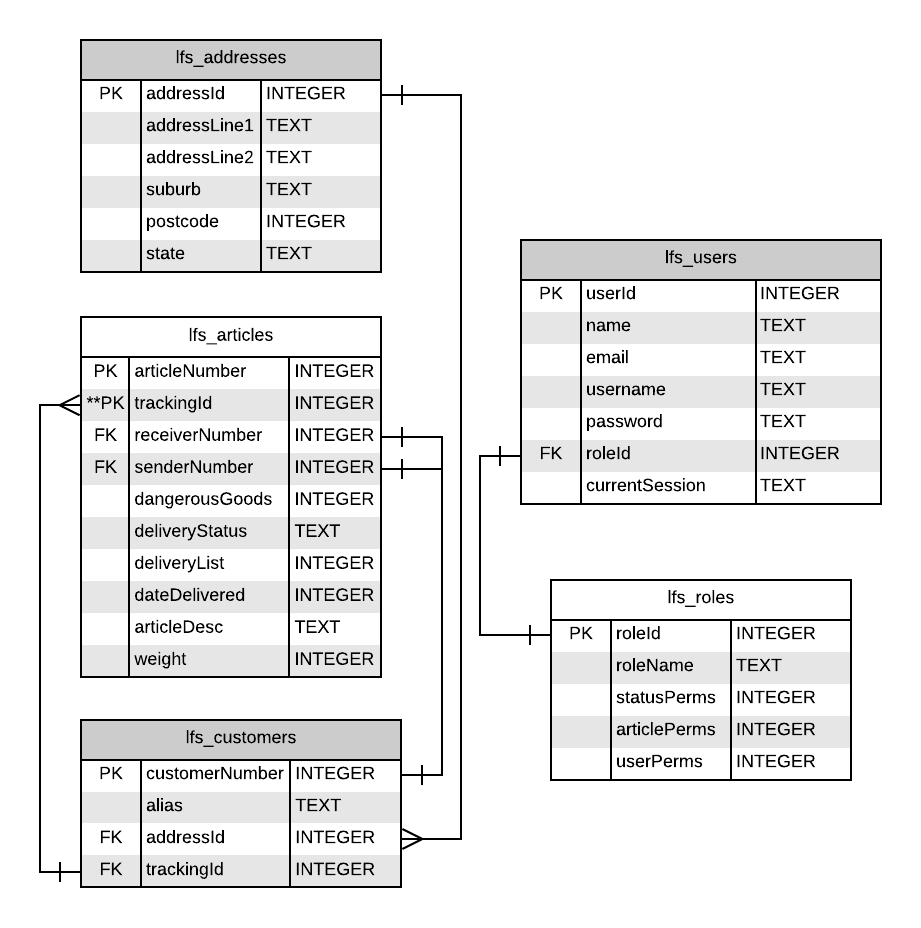
Trial Mode 
XMind:ZEN 
Generate the Tracking Code, Identify the Customer 
and Send the Code 
Associate the tracking code with a package and the 
delivery status from the database 
Receiver Name and Address 
Sender Name and Address 
Description of the package 
Weight of the package (Kilogram, rounded to 1 
decimal place. E.g 3.2kg) 
Declaration of dangerous goods. (Simple YIN data. 
Or True/False) 
Package delivery direction based on the destination 
address in North, South, East, West 
Accept and Calculate payments for postage 
Accept goods and ensure the dangerous goods 
disclaimer is signed 
Tag a package as either North, East, South, West 
based on the destination address 
Check status of delivery with the inclusion of 
relevant data about the package/delivery 
Track the delivery live through a map provider 
Receive a code for the delivery to access the status 
LFS Service Attendants 
Customers 
Access to the Tracking Number 
Website Clients 
Tracking Code 
Local Freight Services ( 
Database/Data 
Package/Customer Information 
Courier Information 
Tracking Number/Code 
Delivery Details 
Courier Name 
Delivery Driver Name 
Contact information of Driver/Courier 
Delivery Status (REQUESTED, DELIVERING, 
COMPLETED) 
Time and Dates of different status points and 
expected status changes (Received request, Package 
picked up, Delivery Completed) 
Name of the Courier or Relevant courier information 
Double check whether the dangerous goods 
disclaimer is signed and taking actions to get it 
signed if not 
Generate delivery lists for North, South, East, West 
Prioritizes and dispatches the generated lists of 
packages 
Change the status of a package or multiple packages 
to DELIVERING 
Delivery Dispatch Manager/Management 
Access the destination address of the assigned 
packages 
Couriers 
Change the status of a package to COMPLETED 









User Session Key Verification on a Staff Webpage:

SELECT \* FROM lfs\_users WHERE currentSession = ?

“SELECT” targets a specific, multiple or all columns that the SQL Statement is requesting. For example \* would select every column and return that to the code to be used later on.  
“FROM” specifies which table the program needs to look into to select the fields from. For example, lfs\_users would be the table that the program would look in to “SELECT \*” (select all) from.

“WHERE” helps determine specifically which row is wanted by specifying the requirements a row must have to be returned to the code. For example, username = “Joe” would find any rows that have the column field of “Joe” in the column username; meaning that all usernames with “Joe” in it will be the ones that will be returned.

“INSERT” allows for the following parameter to be made into a new row in a table. This virtually adds a new entry into the table therefore creating new users in the case of lfs\_users

“UPDATE” enables the program to update an entry in a database to allow for simple changes of a field in a row.

Example: SELECT \* FROM lfs\_users WHERE currentSession = 1726e9c2-ee45-11e9-ad38-9cda3e860d6b



Validating Logins:

**START**

**SET** \_email Form Email Input

**SET** \_password = Form Password Input

**IF** the password associated with \_email is the same as encrypted \_password

**THEN** return session cookie and record the session in the database

**ELSE**

return client to login screen

**END**

Tracker Page:

**START**

**SET** \_id Form Tracking Id/Code

**IF** id is invalid or nothing

**THEN** return to the main page

**ELSE**

**SET** \_article everything associated with the \_id (tracking id) in the database  
 return \_article to the client and display appropriate data

**END**

Staff Articles Dashboard:

**START**

**SET** \_session User’s Session Key Cookie

**SET** \_verify from all users associated with \_session key

**IF** \_verify is not nothing or invalid

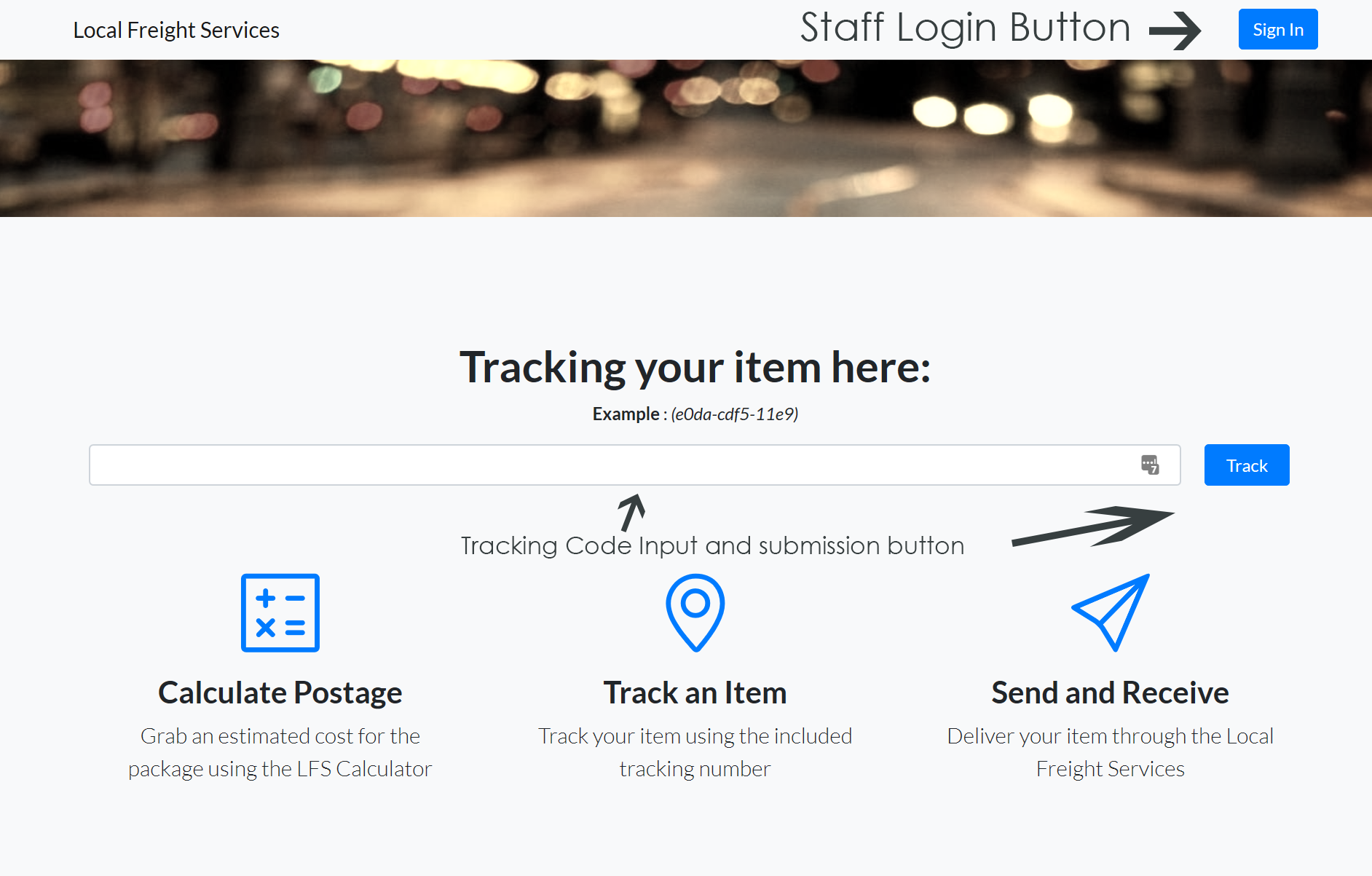
**THEN BEGIN**

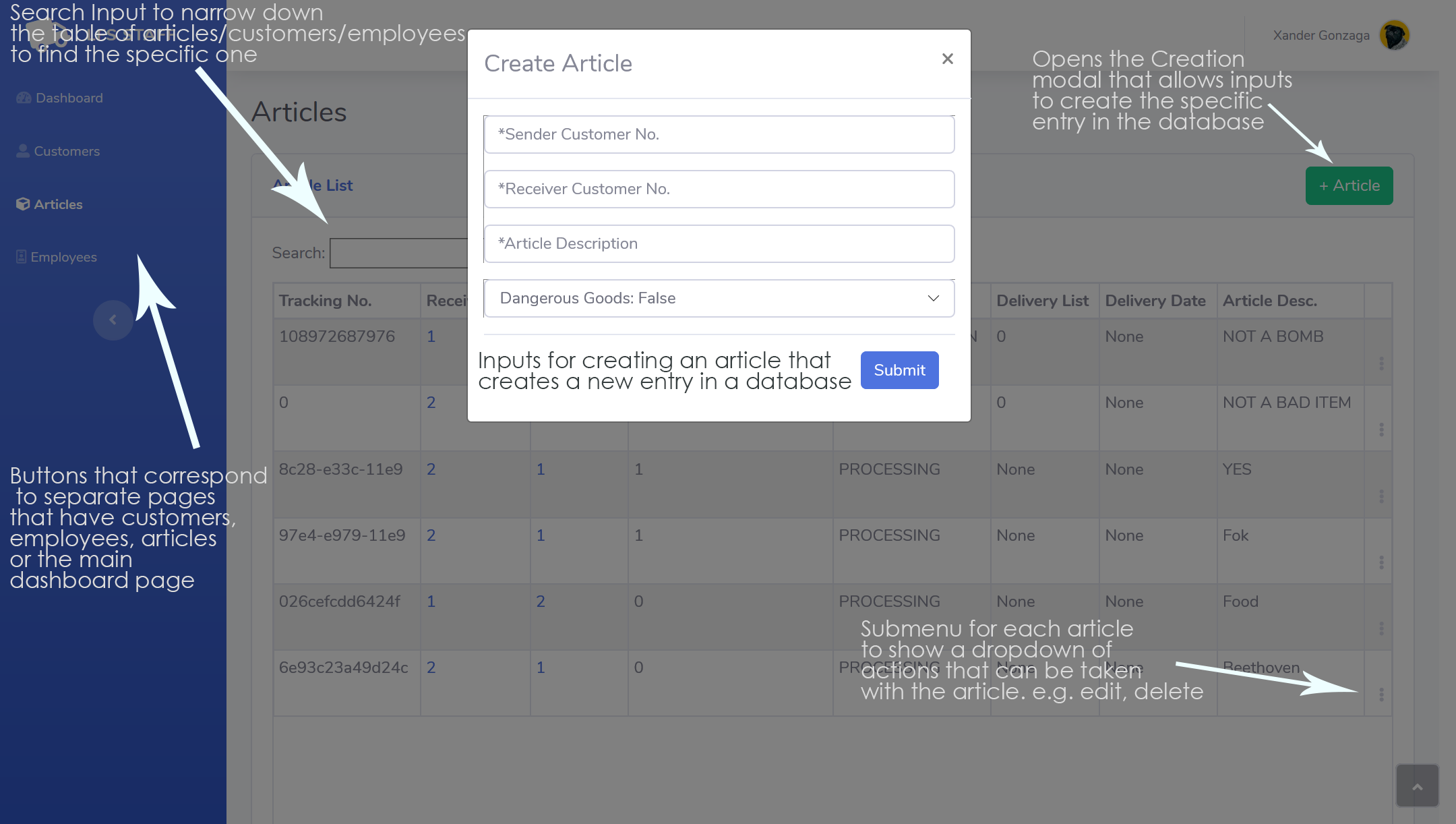
SET \_a GET every article from the database

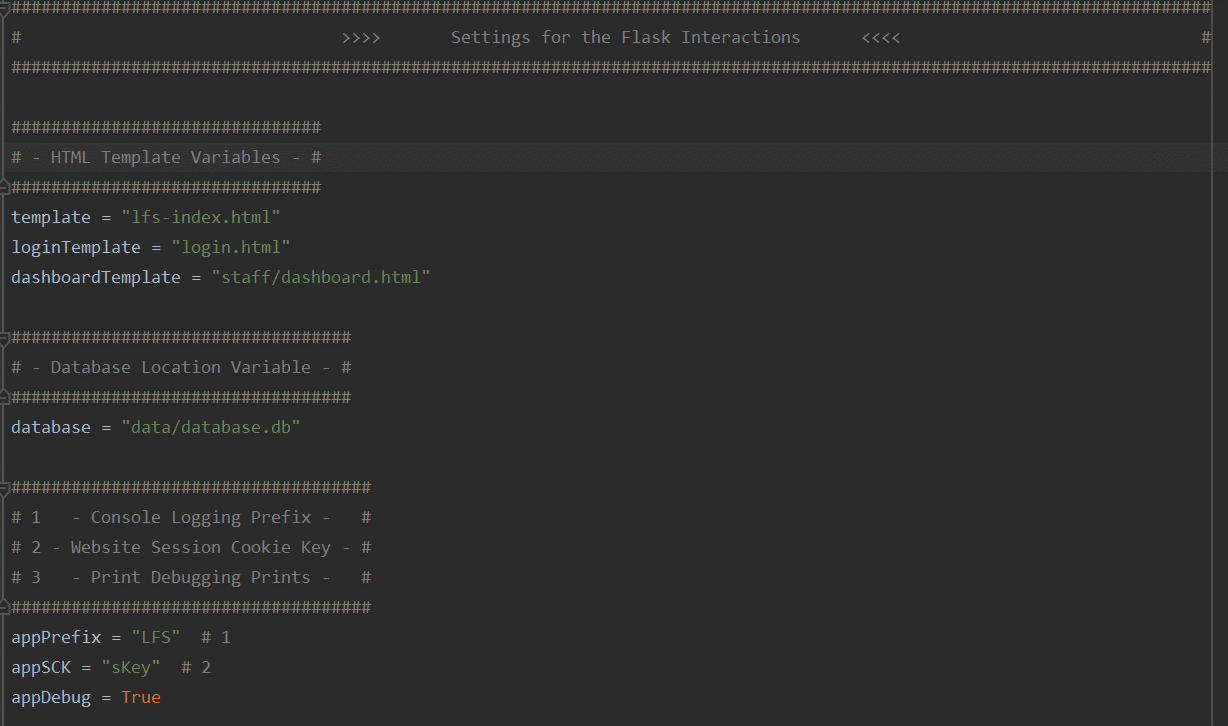
**EXCEPTION**

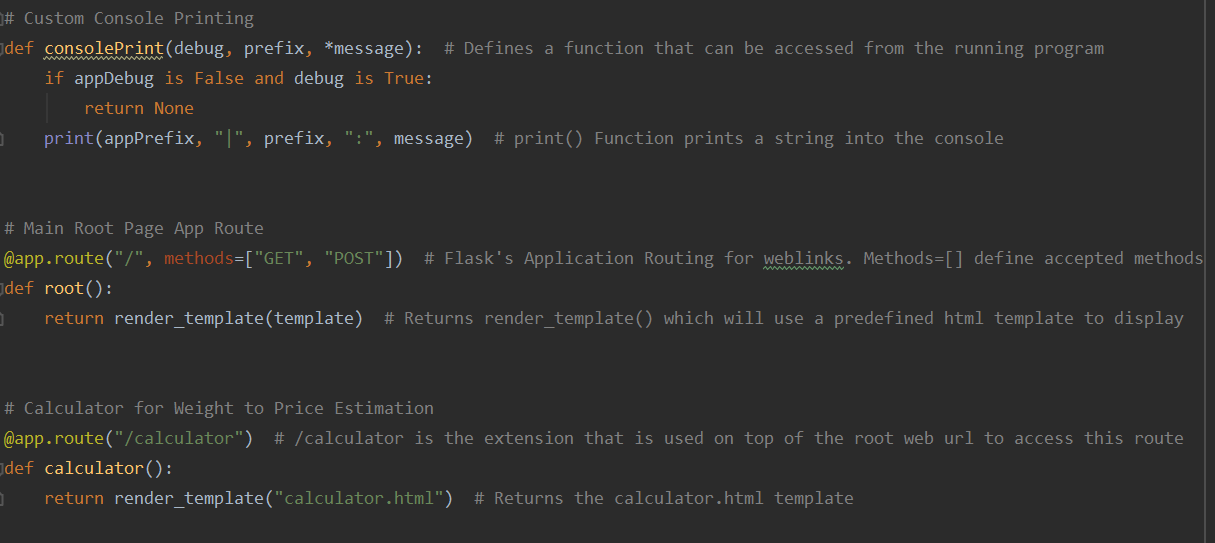
Return to a blank articles staff dashboard page  
 Return to an articles page and pass through \_a to the page

**END**

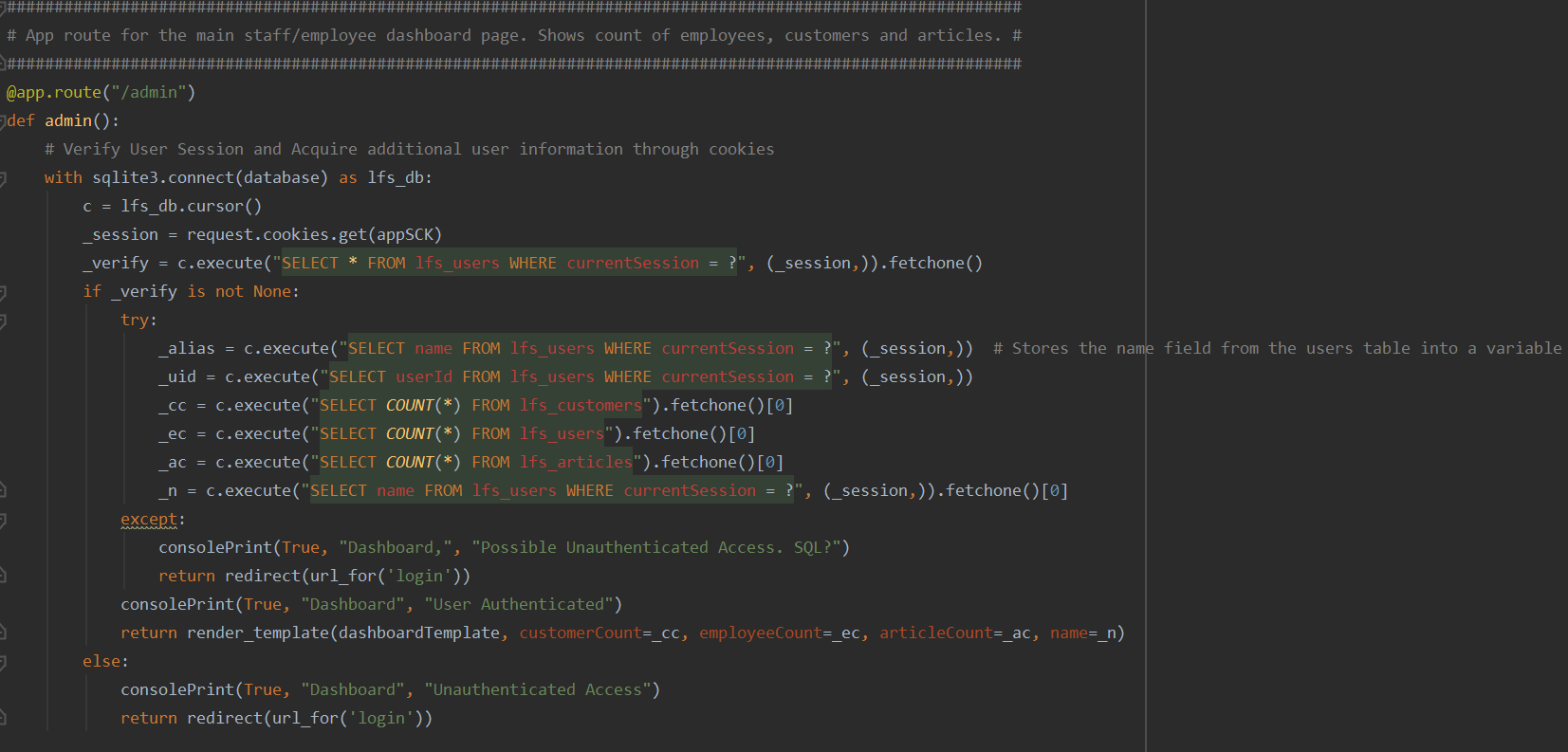


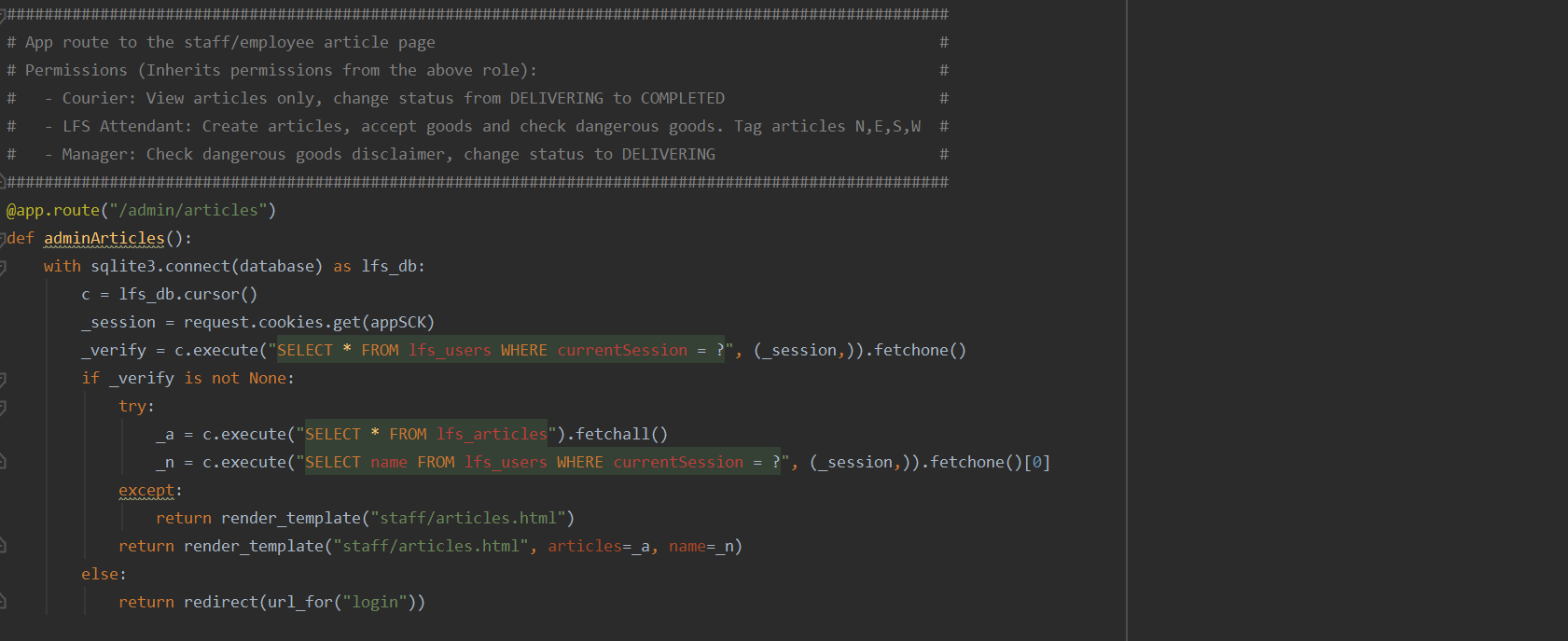


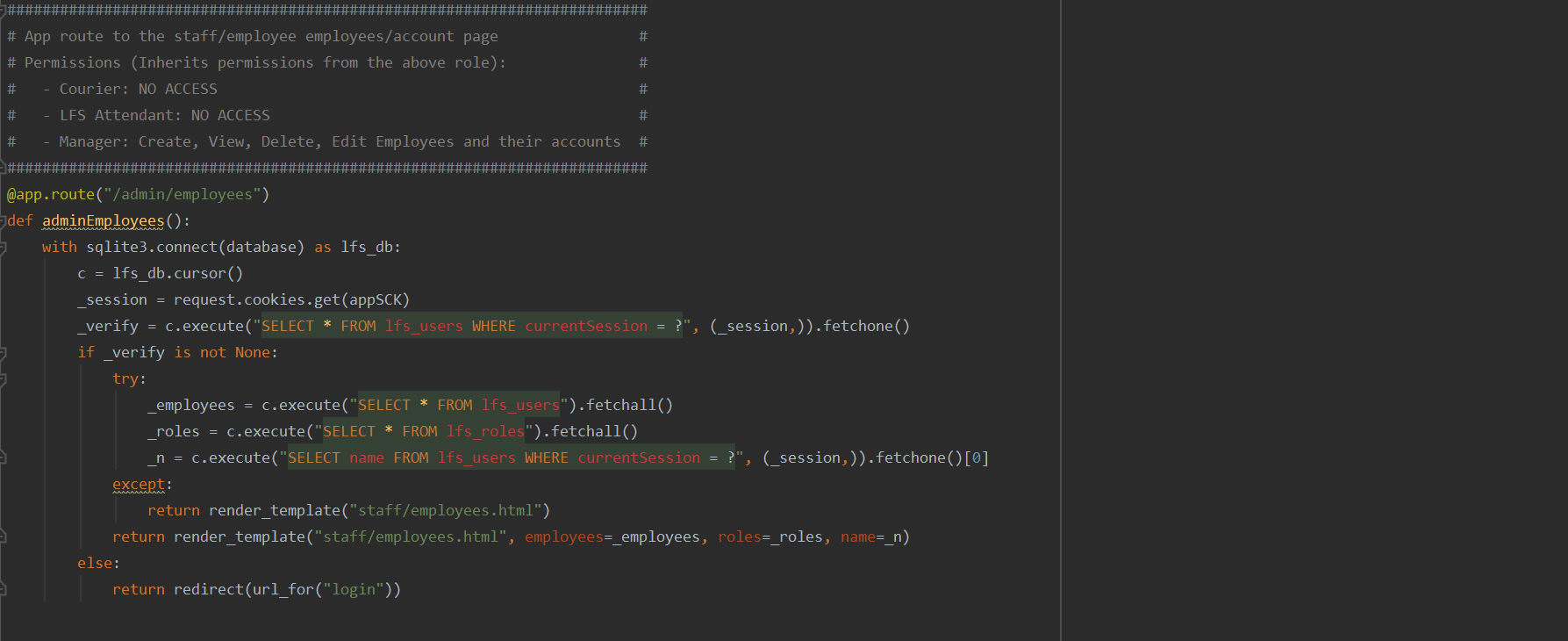


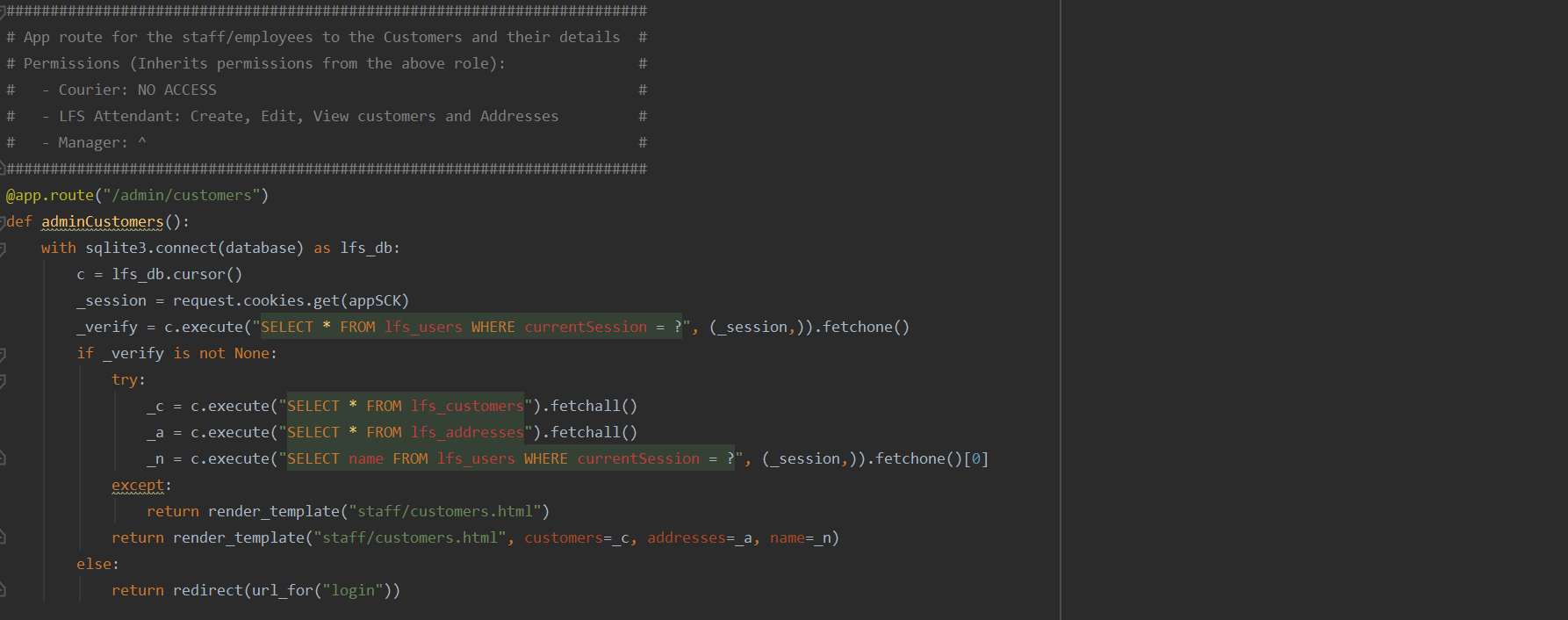


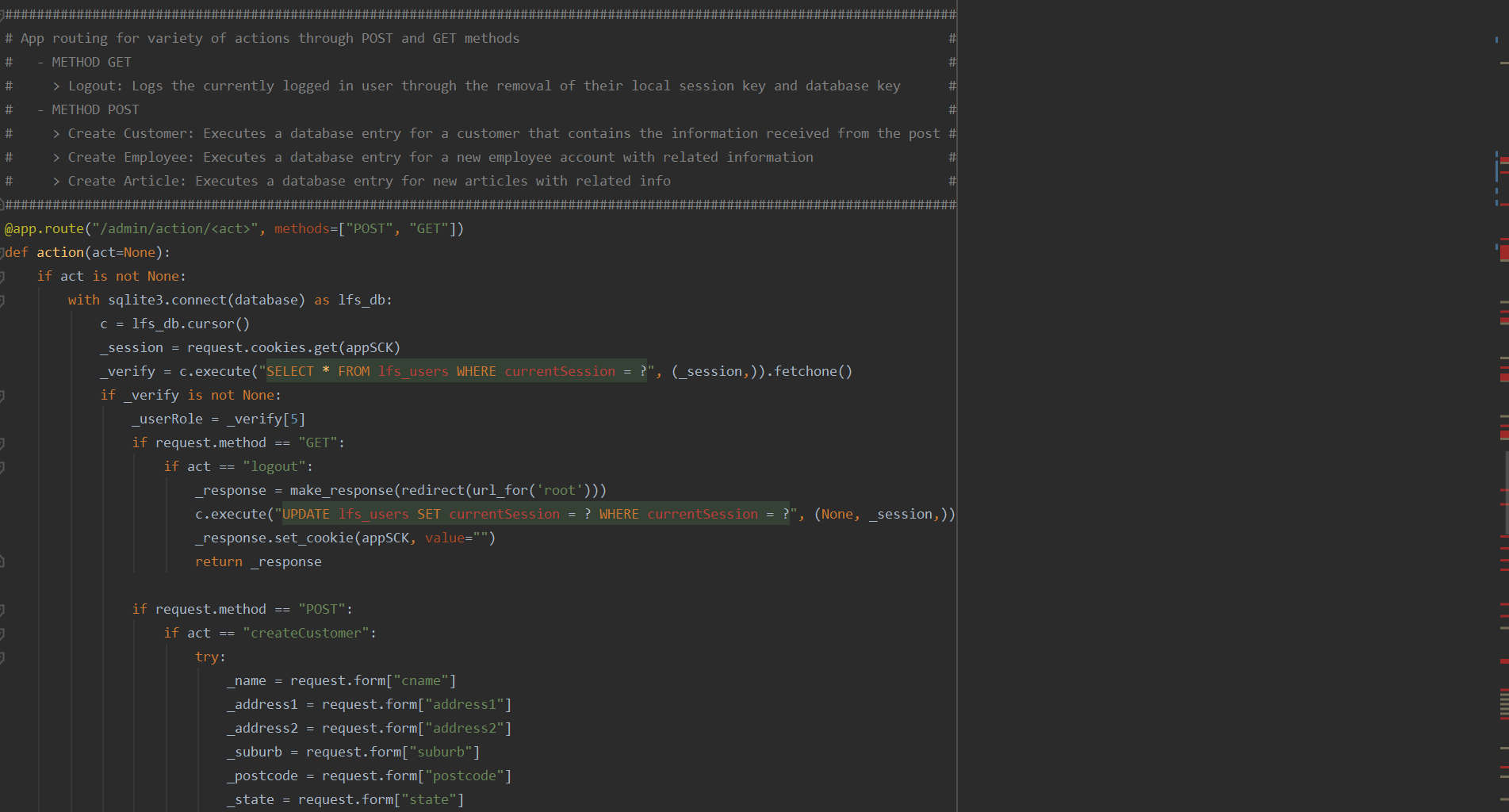


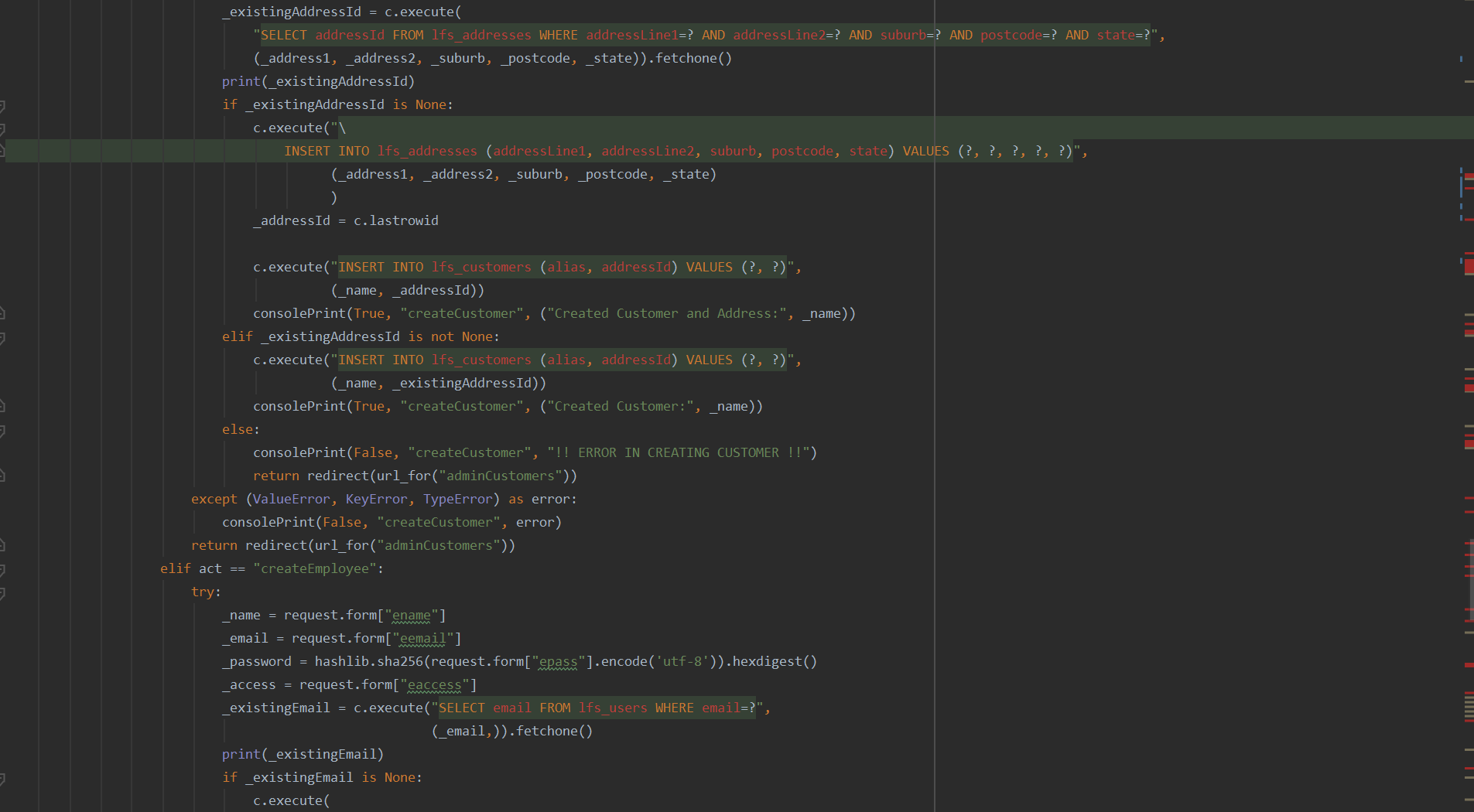


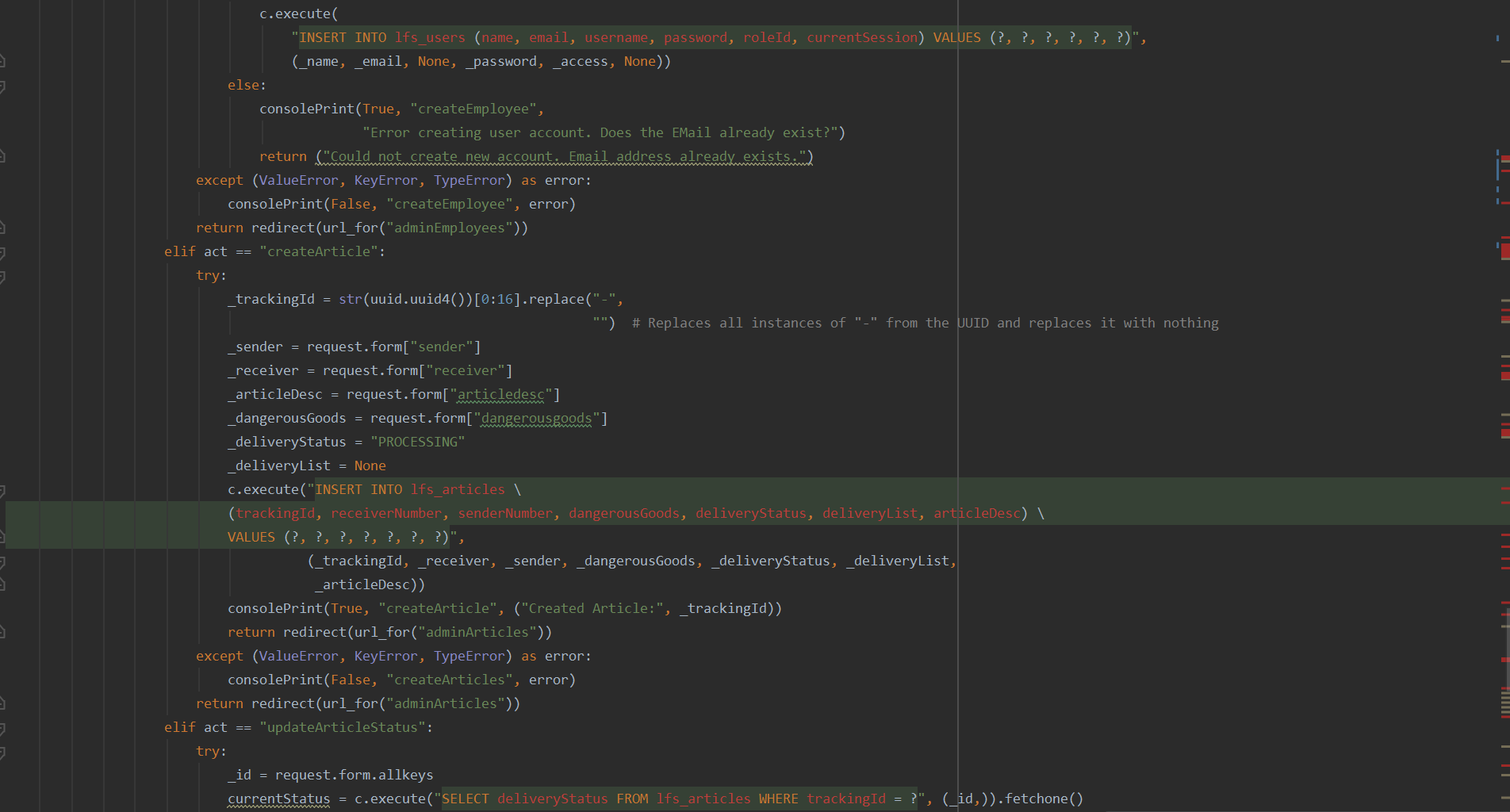


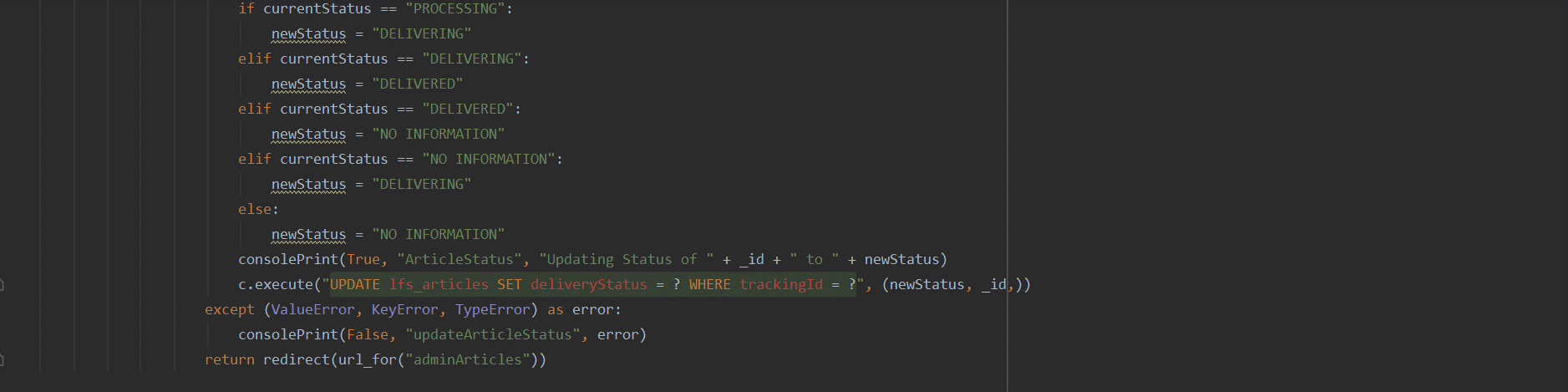




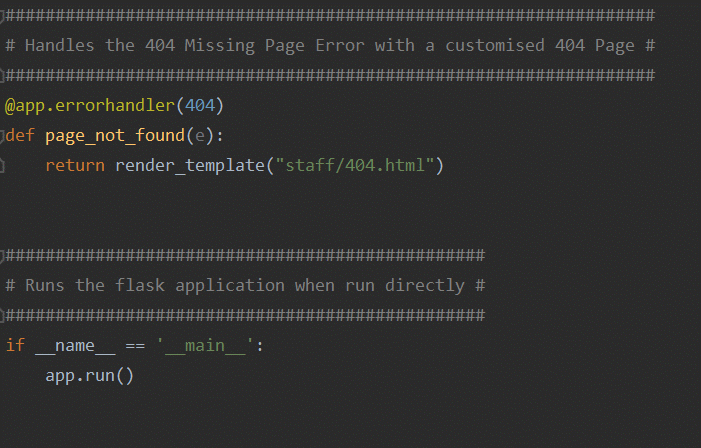


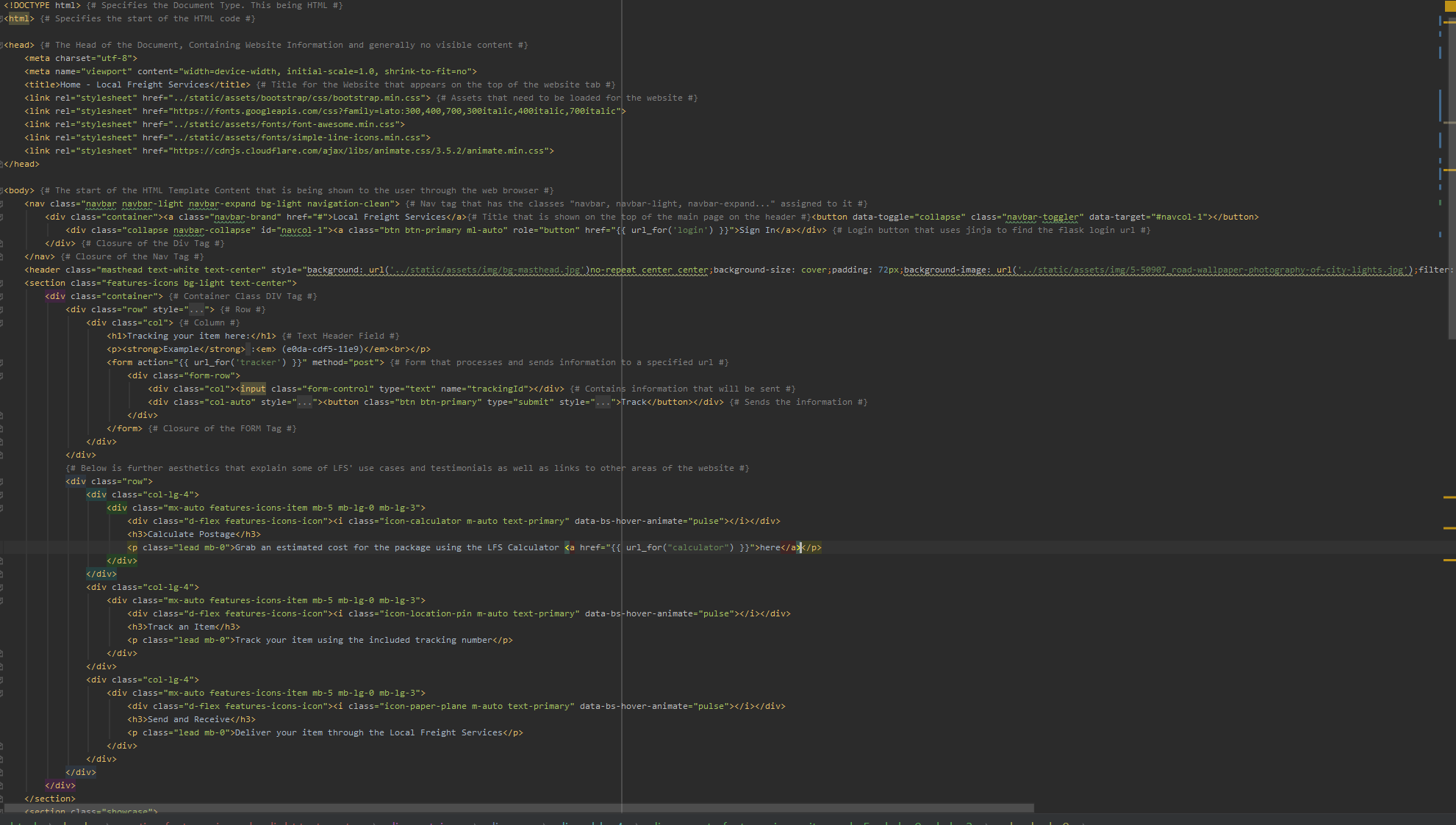


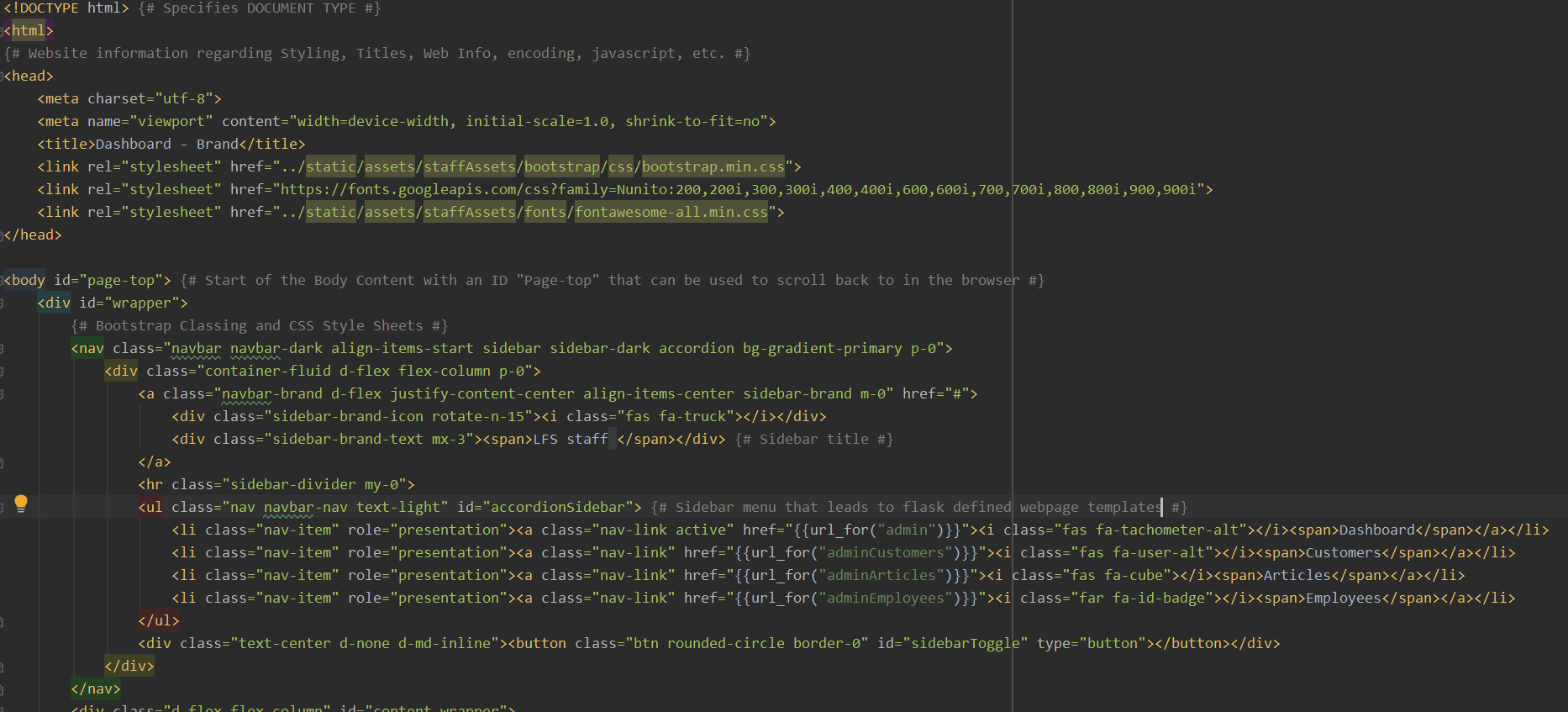












|  |  |  |  |
| --- | --- | --- | --- |
| Personal Impacts | Social Impacts | Economic Impacts | Legal Impacts |
|  |  | * Costly to train each courier driver on how to use the system that allows for changing status of each individual article when delivered | * Loss of tracking for the individual articles could leave unwanted legal disputes |

The web-application prototype for the Local Freight Services is a rough draft of a possible design for the website’s front and back-ends. However, the prototype lacks all the features that would have been beneficial for the consumers, employees and overarching company. The code that was used for the initial prototype does not contain efficient programming which may cause unnecessary strain on the client or server whether its due to over processing or unnecessary code. Some improvements for the Local Freight Services web-application would be better public navigation to a separate tracking page, and weight calculation page. Signing In and the staff dashboard could be revamped to allow for a more secure process for logging in as the current system that processes the login is very basic and may be easily overwhelmed or possibly infiltrated. Over time some future improvements may include accepting non-tracked packages to relieve the delay between changing statuses of each article, public page being more user-friendly, and notifications for consumers who would like to have instant notifications for their tracking.