Backend conceptual design.

1. Once the user searches for parking, we get the user's co-ordinate,

then we query the database for all parkings within 1km radius.

2. We return an array of cordinates starting from the closest to the furthest.

3. We then search for images of the given cordinates using Google maps API or the CCTV feeds

4. Supply the images to the machine learning model determine if there are any free spaces.

5. Return true or false, then if true, give the cordinate to the user.

Then continue searching the coordinates to reduce lag time.

Pseudocode:

AUTHENTICATION OF USER

START

INPUT email, password

IF not authenticate(email, password):

Send 403 message

Direct to sign up page

ELSE:

Direct to search page

STOP

SEARCH ALGORITHM

START

INPUT location

if not location:

ask for location

else:

search DB for closest parkings

return array cordinates of closest parking in order of shortest distance

if cordinates:

free\_parking = null

while free\_parking=== null:

fetch images from either google maps, CCTV footages or hidden cameras.

if images:

pass images to machine learning model

if free space:

free parking = cordinate

else:

get next\_cordinate

else:

return no cordinates found

STOP

History Algorithm

START

visits = null

get visits based on user's details

if visits:

return visits

else:

return no visits yet

STOP

Maps Algorithm

START

input location

get cordinates of parking from db

if cordinates:

return cordinates

else:

return no parkings found

STOP