Log In

**Inputs username, password**

**Outputs** authToken

**Pseudo-Code** Get user SID where username and password match input values.

If user is found then

• Generate AuthToken

• Update user AuthToken and set AuthTokenExpireDateTime per application settings (i.e., take current time and add sliding expiration time offset)

• Return AuthToken

If user is NOT FOUND then

• Clear user AuthToken and AuthTokenExpireDateTime

• Return NULL

Ans:-

SELECT \* FROM sitedb ,AunthenticateUsers

username VARCHAR(20);

password VARCHAR(20);

loginresult INTEGER OUTPUT;

IF

EXISTS

(SELECT (user,username) FROM (sitedb,user) = username AND (user,password) = password);

(UPDATE(user,userauthentication)FROM (sitedb,user));

Declare currenttime DATETIME;

SET currentime =GETTIME;

Declare expirationtime datetimeoffset;

SET expirationtime = GETTIME;

IF

loginresult = 1;

ELSE

loginresult = 0

View User

**Inputs authToken**

**Outputs** User information and siteSID

**Pseudo-Code** Get user SID where authToken matches input value

If user is found then

• Get user name and email

• Get site for user

• Update user AuthTokenExpireDateTime per application settings

(i.e., take current time and add sliding expiration time offset)

If user is NOT FOUND then

• Return NULL

Ans:-

SELECT\*FROM user

WHERE authenticationusers = user;

IF

EXISTS

SELECT \* FROM user,username

SELECT \* FROM use,site

UPDATE(Declare,currenttime= DATETIME;

SET currentime =GETTIME;

Declare expirationtime = datetimeoffset;

SET expirationtime = GETTIME);

ELSE

loginresult = 0

View Site Services

**Inputs siteSID**

**Outputs** Site information; FoodBankSID; ShelterSID; SoupKitchenSID, FoodPantrySID

**Pseudo-Code** Get Short Name, Primary Contact Phone and Location for site where SID matches input value

If site FOUND then

• Get Services for site

o Set FoodBankSID to service SID for food bank (-1 if no food bank at site)

o Set ShelterSID to service SID for shelter (-1 if no shelter at site)

o Set SoupKitchenSID to service SID for soup kitchen (-1 if no soup kitchen at site)

o Set FoodPantrySID to serviceSID for food pantry (-1 if no food pantry at site)

• Update user AuthToken and set AuthTokenExpireDateTime per application settings (i.e., take current time and add sliding expiration time offset)

• Return site details and SIDs for services

If site NOT FOUND then

• Return NULL

Ans:-

SELECT\*FROM site

UPDATE site

SET shortname

INSERT INTO site

VALUES(roy,rahil);

SET primarycontactphone

INSERT INTO site

VALUES(9512345678,9516785678);

SET location

INSERT INTO site

VALUES(riverside,corona);

IF

EXISTS

SET foodbankId

INSERT INTO site

VALUES(1,2);

SET shelterId

INSERT INTO site

VALUES(2,3);

SET SoupkitchenId

INSERT INTO site

VALUES(1,6);

SET foodpantryId

INSERT INTO site

VALUES(3,6);

SELECT \* FROM user,site

UPDATE Declare,currenttime= DATETIME;

SET currentime =GETTIME;

Declare expirationtime = datetimeoffset;

SET expirationtime = GETTIME);

ELSE

loginresult = 0

Delete Site Service

**Inputs authToken, site service SID**

**Outputs** ErrorCode (0 for successful execution)

**Pseudo-Code** Get user SID where authToken matches input value

If user is found then

• Get site SID for user

• Get site SID for service SID

• If site SID’s are NOT EQUAL then

o Return ErrorCode “User can only delete for his or her own site”

• Get number of services for site

• If number of services is ONE then

o Return ErrorCode “User cannot delete the last service at a site”

• DELETE service SID

If user is NOT FOUND then

• Return NULL

Ans:-

DELETE\*FROM sitedb,user

WHERE authenticationusers = user;

IF

EXISTS

SELECT (user,site) FROM (sitedb,user)

UPDATE(siteid,site)

SELECT (user,site) FROM (sitedb,user)

UPDATE(serviceId,site)

ELSE

siteId = ERROR

GET numberofservices = result;

IF

EXISTS

result = 1;

ELSE

result = ERROR

DELETE\*FROM sitedb,serviceId

ELSE

Result = 0