|  |
| --- |
| user\_table |
| user\_id (primary key) |
| username (TINYTEXT) |
| password (TINYTEXT) |
| user\_email(TINYTEXT) |

**Stored procedures for user\_table:**

initUserTable() – creates the table

Mapping: /user/init

CreateUser(username, password, user\_email) – insert everything into the user\_table

Mapping: /user/create

Returns:

1. for successful creation
2. for user already exists

Login(username, password) – find user by username and confirm if password matches, then return user\_id and username

Mapping: /user/login

Returns:

0 for successful login

1 for wrong password

ChangePassword(user\_id, oldpassword, newpassword) – find user by user\_id and then confirm old password matches before updating it with the new password

Mapping: /user/changePassword

Returns:

0 for successful change

1 for wrong password

|  |
| --- |
| org\_table |
| org\_id (primary key) |
| org\_name (TINYTEXT) |
| contact\_name (TINYTEXT) |
| contact\_number (TINYTEXT) |
| contact\_email (TINYTEXT) |
| notes (MEDIUMTEXT) |

**Stored procedures for org\_table:**

initOrgTable() – Creates the org table

Mapping: /org/init

CreateOrg(org\_name, contact\_name, contact\_number, contact\_email, notes) – insert everything into the org\_table

Mapping: /org/create

Returns:

0 for successful create

1 for org already exists

UpdateOrg(org\_name, contact\_name, contact\_number, contact\_email, notes) – update the table with all values

Mapping: /org/update

Returns:

0 for successful update

1 for org not created yet

|  |
| --- |
| donation\_table |
| order\_id (primary key) |
| org\_id (foreign key) |
| org\_name (TINYTEXT) |
| category (TINYTEXT) |
| weight (INT(1000000)) |
| donation (INT(1)) – 0 for outgoing/ 1 for incoming |
| date\_created(DATETIME) |

**Stored procedures for donation\_table:**

Donation(org\_name, category, weight, donation) – add entry to the donation table with incoming or outgoing donation flag

Mapping: donation/input

Returns:

0 for success

Example: localhost:8080/org/create?org\_name=org1&contact\_name=Bob&contact\_number=123-4567&contact\_email=billybob@jo.bob&notes=donated a french fry last week

ListOrg(donation) – return org\_name for all incoming or outgoing entries

Mapping: donation/listOrg

Returns:

String array of org names

Example: localhost:8080/donation/input?org\_name=org1&category=vegetables&weight=70&donation=0

ListInfo(org\_name, donation) – return org\_name, category, weight, date\_created

Mapping: donation/listInfo

Returns:

List of donations

**Stored procedure to generate report data:**

GetReport(donation, start\_date, end\_date,)

Mapping: donation/report

\*\*\*Need to create a table for the user model and stored procedures to query table\*\*\*

Widget Donation object:

String org\_name

Int rank

Int donation

String deli

String dairy

String meat

String produce

String pantry

String bakery

InitWidget()

Mapping: donation/initwidget

Returns: Three WidgetDonation objects

WidgetDonation(org\_name, rank, donation, deli, dairy, meat, produce, pantry, bakery)

Mapping: donation/widgetdonation

Returns: WidgetDonation object

Stored procedures left to do for the widget:

hasEntered(timestamp) – return whether data has been entered or not

freshList(username) – adams large crazy algorithm

staleList(username) – get back the updated list

add username column to donation table