**System Architecture Document**

**Functions to be implemented:**

**Client:**

* Register/Login*(for admins)*
* List shops *// GET /shops*
* Search for shops *// search?q=shop1*
* List the recently viewed restaurants *// localStorage*
* Scan qr code at the table to claim a table:
  + Scan qr code to get table info.
  + PUT request to rails API to claim the table if available.
  + Table's owner id is generated randomly on the client and saved to localStorage and sent to server too.
* List the items on the menu for a specific shop.
* Search for items on the menu.
* Add to order.
* Send order. *// POST {*
  + *"items":[*
  + *{*
  + *"name":"tea",*
  + *"price":5,*
  + *"quantity":2,*
  + *"shop\_id":1,*
  + *"table\_id":3,*
  + *"owner\_id":"asdrewtg"*
  + *}*
  + *]}*
* View the check/bill. *// GET /shops/1/tables/3/order\_items*
* FOR logged in ADMINS
  + View tables
  + View and complete current orders.
  + View waiter requests.
  + View and manipulate items.

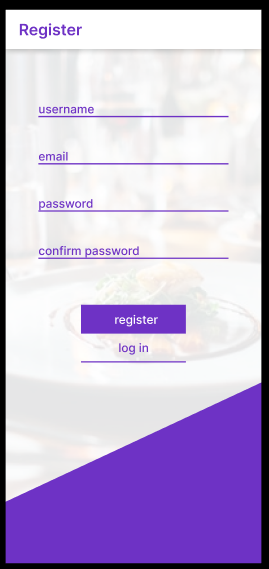
**Server:**

To communicate with the mysql database, this app will use a rails API.

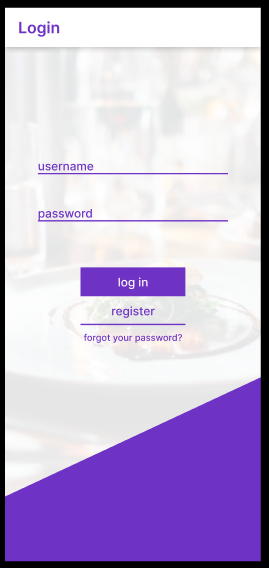
* Register/Login(for admins)
* Provide all the necessary routes for the client functions to work. Mostly CRUD operations.
* Authorize only the logged in users under admin route and to see the following pages:
  + AdminOrders, AdminTables, AdminTableDetails, AdminMenu
* Server actions to handle are:
  + **shops#search**
  + **items#index**
  + **items#create**
  + **items#show**
  + **items#update**
  + **items#update**
  + **items#destroy**
  + **requests#index**
  + **requests#create**
  + **requests#destroy**
  + **order\_items#index**
  + **order\_items#create**
  + **order\_items#update**
  + **order\_items#update**
  + **Order\_items#current**
  + **tables#index**
  + **tables#create**
  + **tables#show**
  + **tables#update**
  + **tables#update**
  + **tables#destroy**
  + **shops#index**
  + **shops#create**
  + **shops#show**
  + **shops#update**
  + **shops#update**
  + **shops#destroy**

**Pages:**

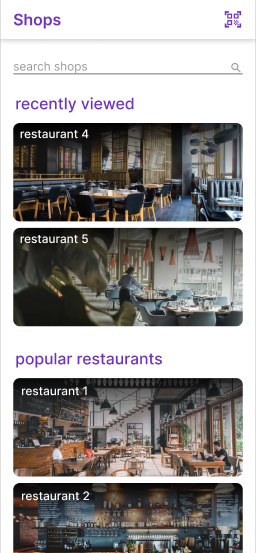
* **Customer (no need to log in)**
  + Register Page



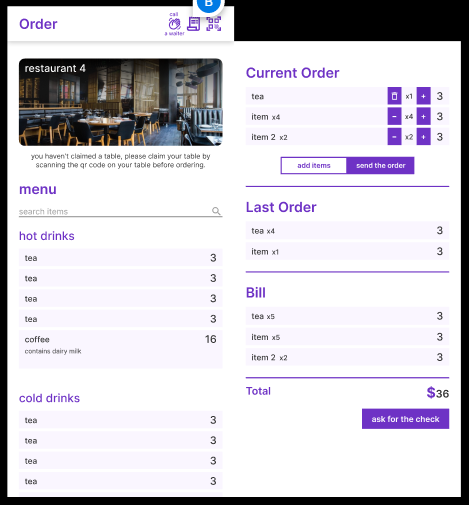
* + Login Page



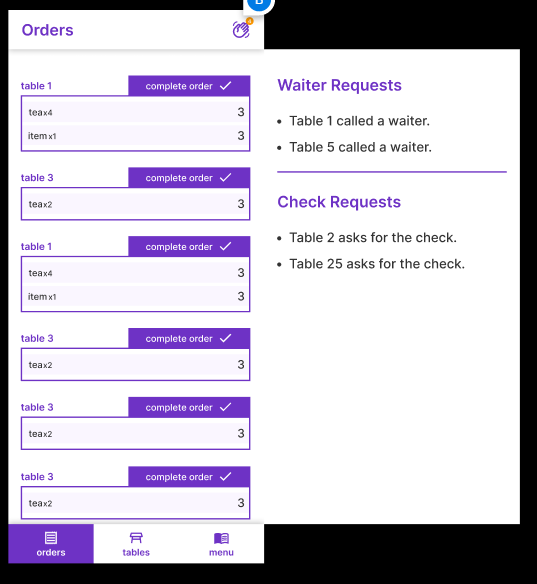
* + Shops Page



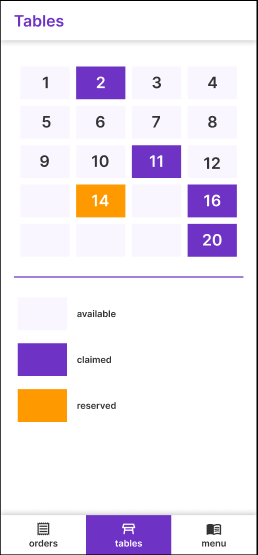
* + Order Page



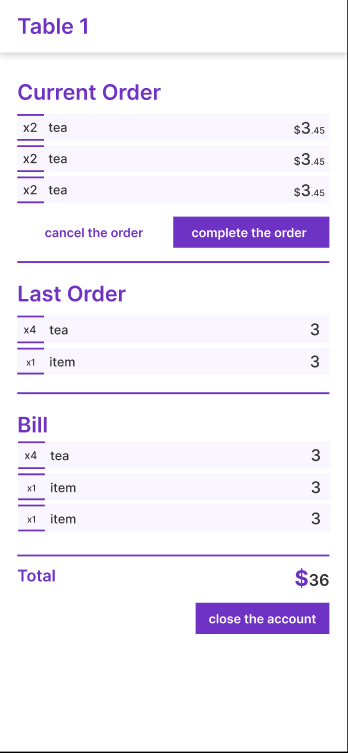
* **Admin (logged in only)**
  + AdminOrders Page



* + Admin Tables Page



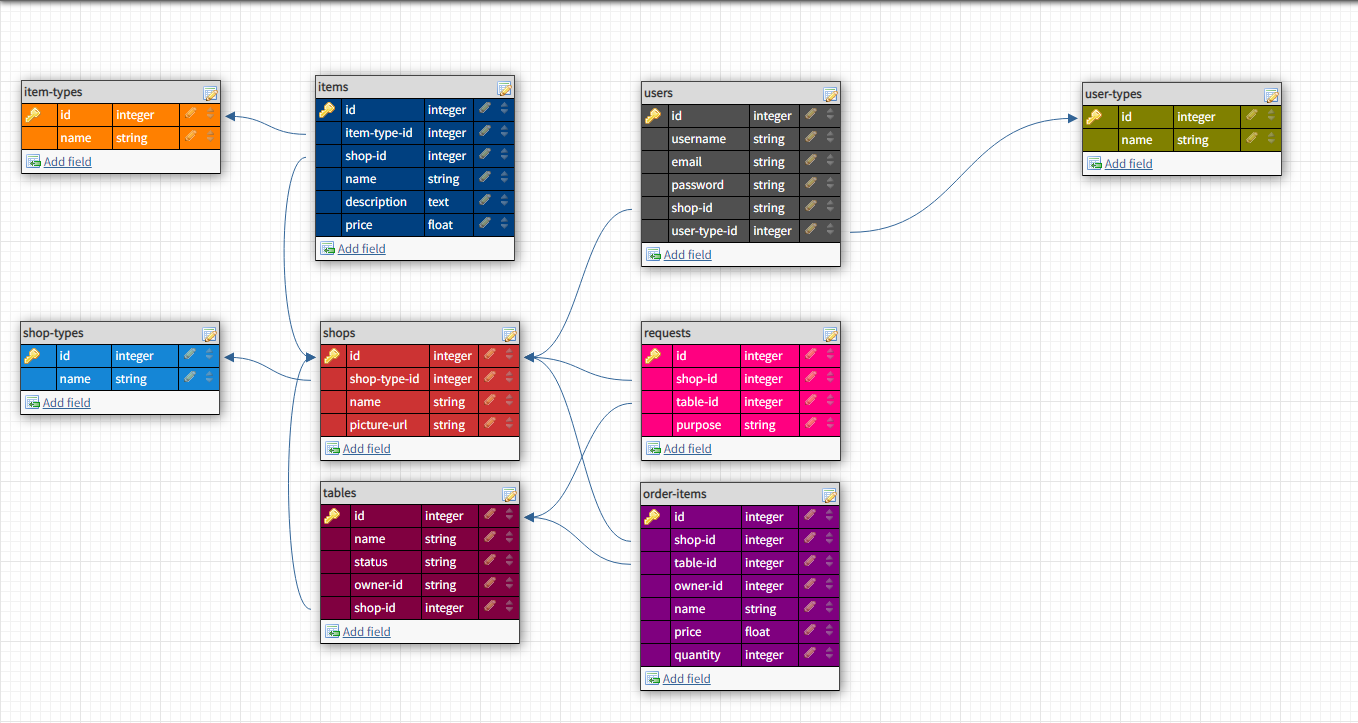
* + AdminTableDetails Page



* + AdminMenu Page



**Database:**



Architecture Overview:

|  |  |  |
| --- | --- | --- |
| **CLIENT** | **API** | **Picture Files**  **(shop pics)** |
| **DB TABLES** |