# URLs

|  |  |  |  |
| --- | --- | --- | --- |
| Page | URL | Name | Notes |
| Login | / | “login\_register” |  |
| Register |  |  |  |
| Login\_submit | /login\_submit | “login\_submit” | * URL for login POST request |
| Register\_submit | /register\_submit | “register\_submit” | * URL for register POST request |
| Home | /home | “home” | * Redirected to from “login\_register” for logged-in users |
| Search | /search?query=query | “search” |  |
| Food | /food/<id> | “food” |  |
| Recipe | /recipe/<id> | “recipe” |  |
| Equipment | /equip/<id> | “equip” |  |
| New\_item | /new\_item | “new\_item” | * May need to split in to one for each of type of item * Both for displaying form an submitting |

# Templates for Pages

* All templates also require the context:
  + “page\_name” – i.e. Home, Login, Food\_name, Search: Query
  + “page\_type”: “home”, “search”, “login\_register”, “food”, “recipe”, etc
  + “user” – the logged in user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Template Name | Use | Context Keys | Context Value | Notes |
| login.html | Login | “form” | Login form |  |
| Register.html | Regster | “form” | Register form |  |
| hs\_main.html | Home screen/search results | “foods”  “recipies”  “equipments” | array of items  array of items  array of items | * the plurals are not proper english, but we need to maintain the single-time/array convention * for home is most recent ~5 of each, for search its the results * Average cooking time and average rating should be calculated from the reviews, in the view, and dynamically added to the object as properties “extra”, and “stars”, respectively |
| “item\_main.html” | For individual food, recipe and equipment pages | “item”  “type” | The model object |  |
| new\_item\_main.html | For making a new item | “item\_form” | Item form | * One form for all items, determine which with dropdown, or many (might need multiple templates)? |

* Items can have optional extra propertied added at runtime – things like the amazon link, average cooking time, etc.
  + We should go over this in person to figure out the details
* We need to detail how pictures are going to be stored/retreived