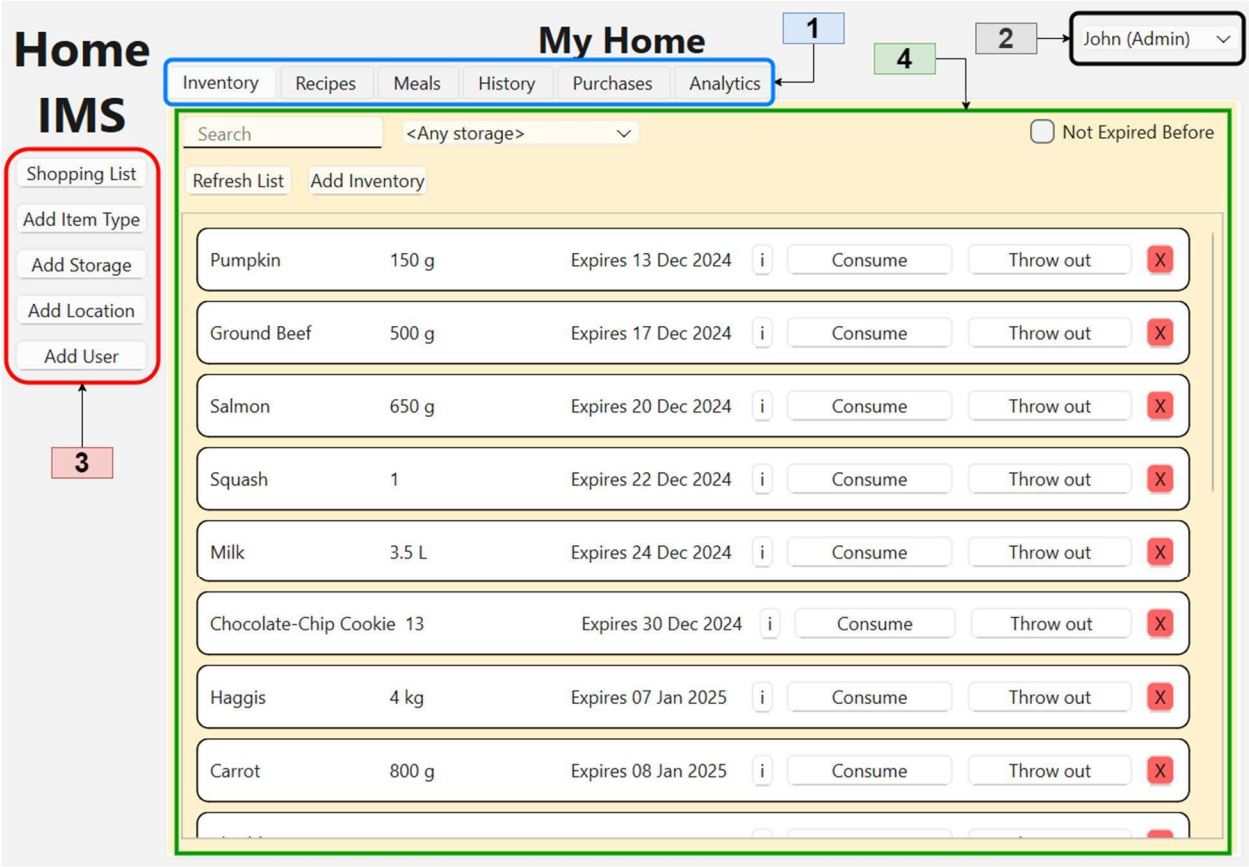
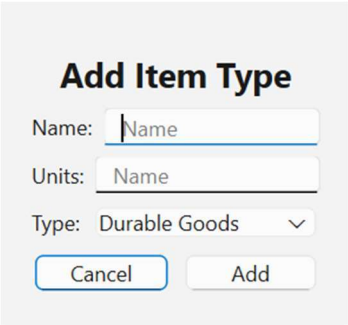
User Manual

Overview

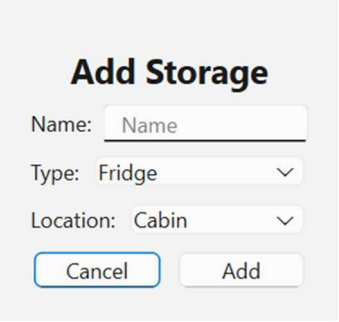


- All screens in this interface, except for small popup windows, have the same general structure as the picture above.
- The coloured numbers and boxed on the picture are annotations and correspond to the numbered descriptions on the next page.
- Areas labeled 1, 2, and 3 are viewable from all tabs.
- You *must* select a user (see 2) before using the interface. All clickable elements will be greyed out if you do not.
    - Parent users (admins) can use every part of the application.
    - Regular users cannot click on any **Add** button (greyed out), and they cannot schedule or consume meals.

1. Tab Selector
   - Allows you to change the view to the other tabs by clicking. Please see the respective tab's section for an explanation of its functionality.
2. User Selector
   - Drop down menu to select the user who is currently using the application. There is no access control for selecting users at this time. Users must have already been created to be selected (see **Add User** button on the sidebar).
3. Side Bar Buttons
   - Shopping List
     - Auto generates a copyable shopping list for ingredients needed to make the meals scheduled in the next seven days.
   - Add Item Type
     - Allows the user to create a new item type. This does *not* add anything to inventory. An item type is what the item is, not a specific instantiation of it.

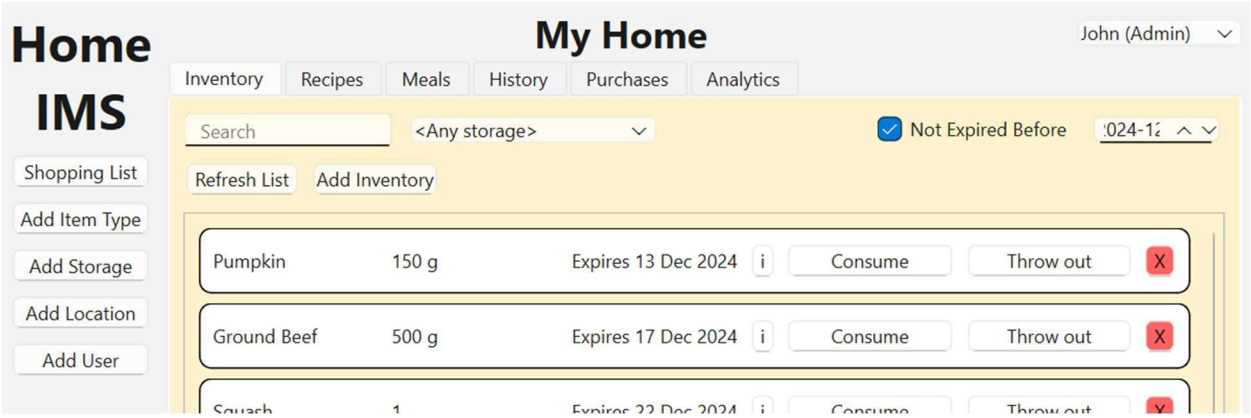


   - Add Storage
     - Allows the user to create a new storage. The desired location must already be defined (see below)
   - Add Location
     - Allows the user to create a new location. Only name is needed.
   - Add User
     - Allows the user to create a new user. Checking the **Parent** box will make the new user an admin.

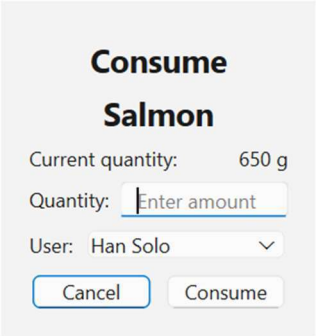


4. Tab Display
   - Shows the interface for the tab selected. Please see the respective tab's section for an explanation of its functionality.

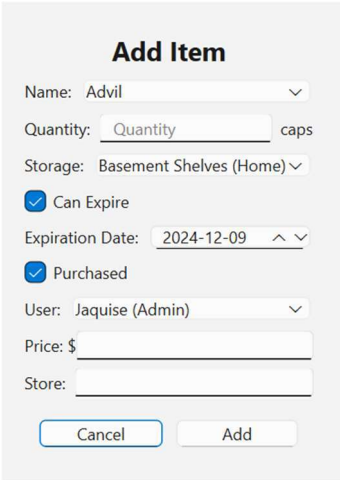
Inventory



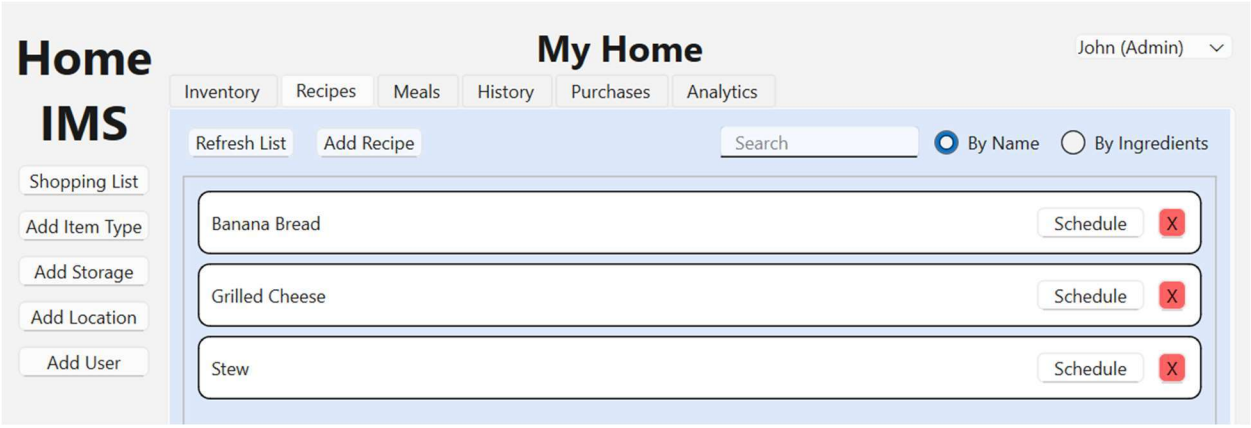
- The inventory tab displays items in the inventory are horizontal cards.
    - Cards display Lists the name, quantity and expiration date of an item.
    - The **i** information button launches a popup window to display more information about the item, such as its storage location.
    - The **Consume** and **Waste** buttons each launch a popup window to specify how much of the item was consumed/wasted, and who consumed/wasted it. Consumed and wasted items are recorded differently in History and affect Analytics.
    - The **X** button deletes the inventory entry without recording it in History. This is for error correction.



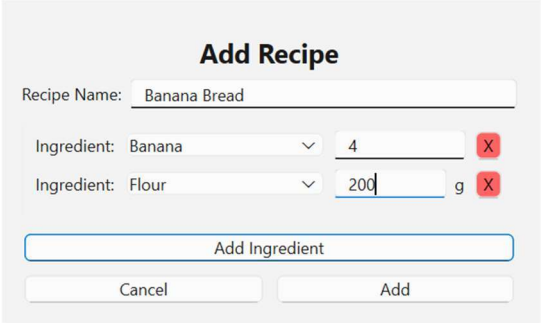
- Filtering criteria for the inventory items is places above the cards.
    - After applying any filter criteria, you must click the **Refresh List** button.
- The **search bar** allows you to search for inventory items by name.
- The **storage dropdown** menu will show items in the selected storage.
- The **Not Expired Before** checkbox and date entry allows you to filter for items that will have not expired before the chosen date if the box is checked.
- The **Add Inventory** button launches a popup window to add an item to inventory.
    - You can only add items of an *existing item type*.
    - Units for the item are determined by the existing item type.
    - Optionally add the items expiration date.
    - Optionally record this inventory entry as a purchase.



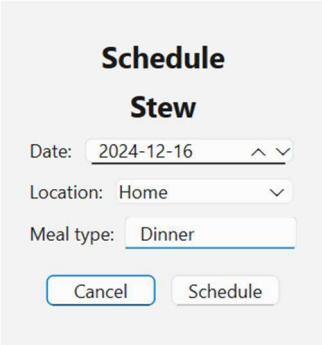
Recipes



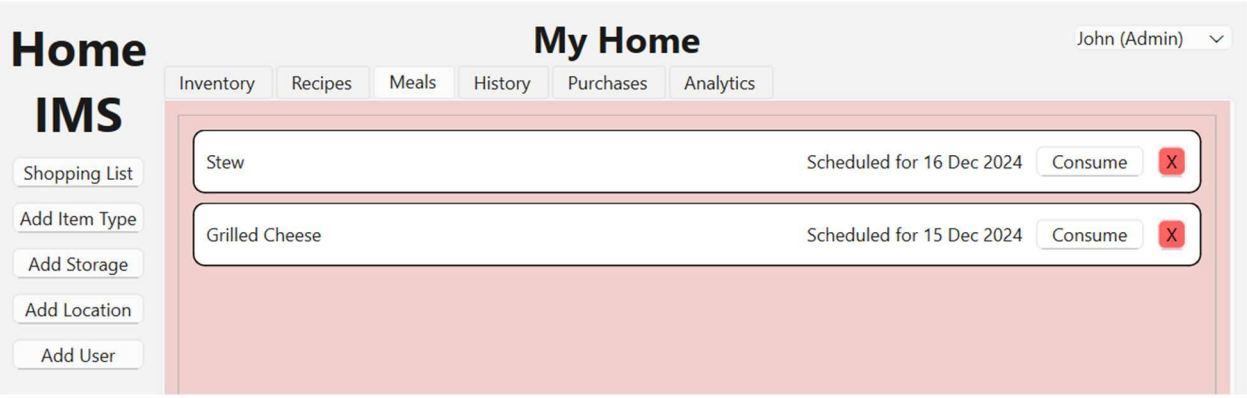
- Recipes can be searched **By Name** or **By Ingredients** in the recipe. **Refresh List** must be clicked to show the search.
- **Add Recipe** launches a popup window to create a new recipe. Ingredients must be existing item types, but they do not have to be in the inventory.
    - Clicking **X** removes an ingredient
    - Clicking **Add Ingredient** adds another ingredient entry row.



- Recipes are displayed as horizontal cards, similar to Inventory.
    - **Schedule** launches a popup window to schedule a recipe as a meal. The date, location, and meal type is specified. This constitutes the Meal Planner feature of this application.
    - Clicking **X** will delete with recipe from the database. If that recipe is scheduled as a meal, those meals will also be deleted.

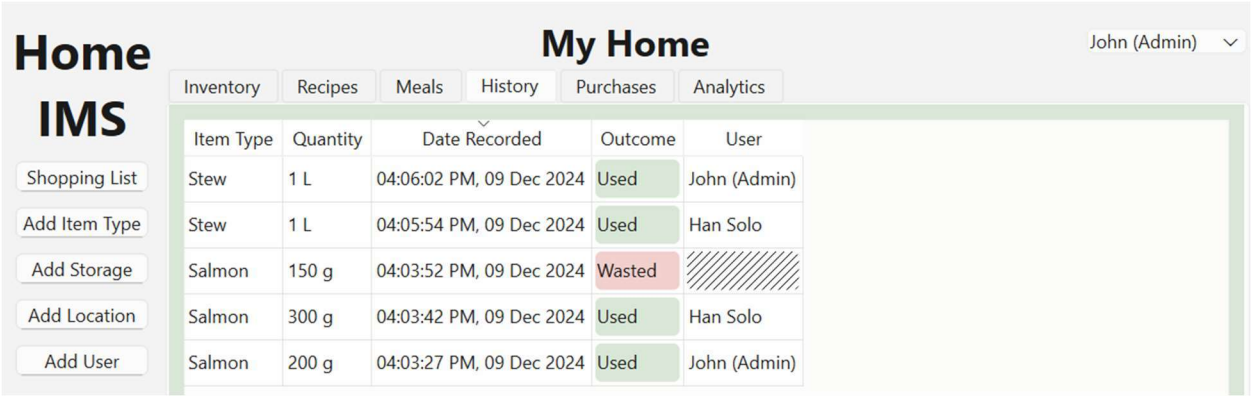


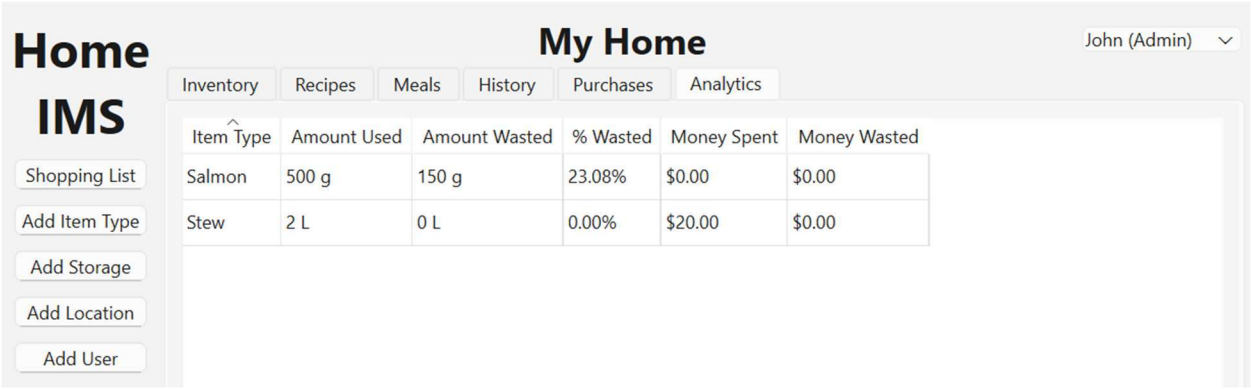
Meals



- The Meals tab shows your upcoming meals as horizontal cards by descending date.
- The only way to add a meal is to schedule an existing recipe (see Recipes).
- The **X** button will delete the meal from the database.
- The **Consume** button will consume the ingredients needed for that meal from the inventory, and it will be logged as consumed by the current user.

History, Purchases and Analytics

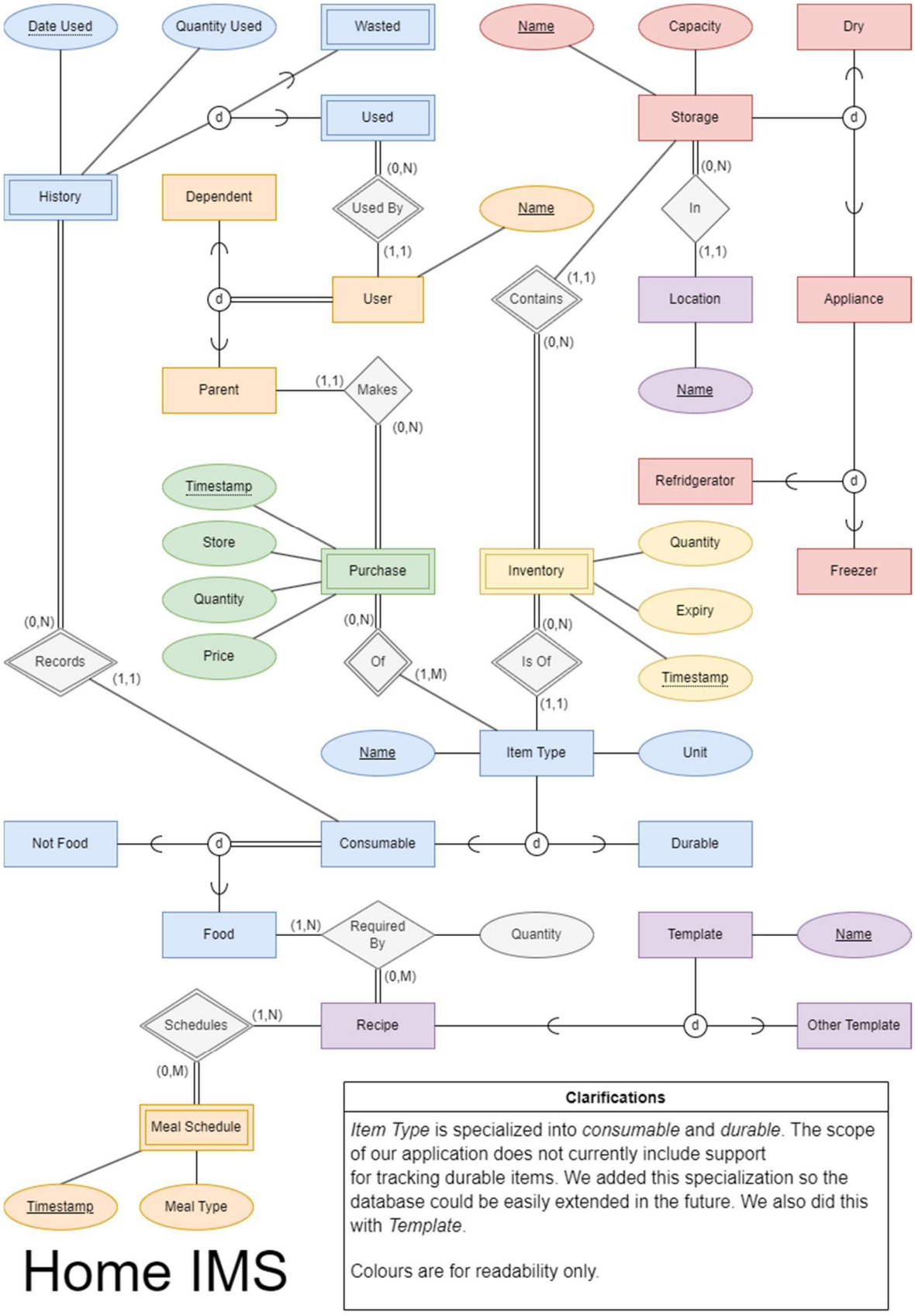




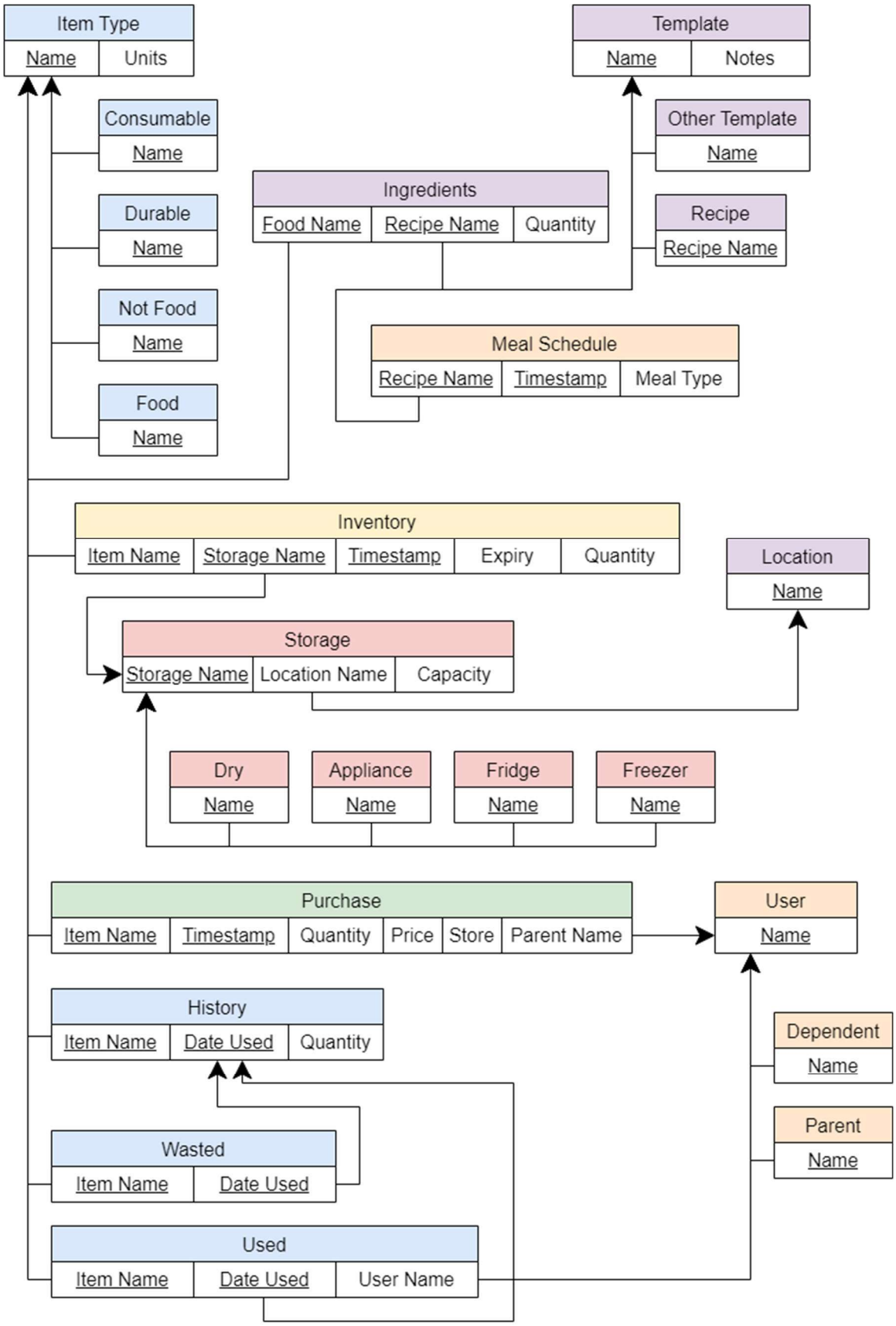
- The History, Purchases and Analytics tabs are shown above in that order.
- Each tab is structurally the same except they show a different table of data.
- Each column in the tables can be clicked to sort by ascending or descending order.

Appendix A – Diagrams

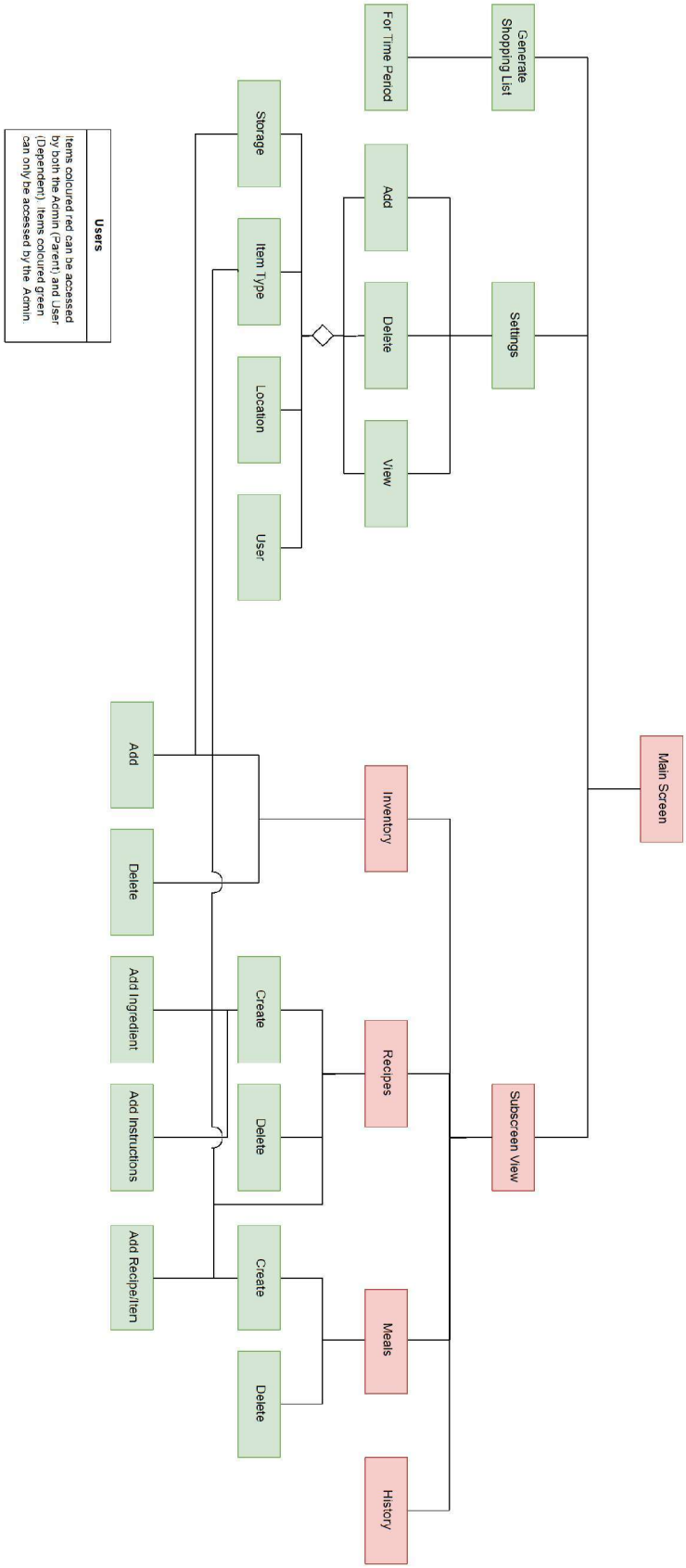
Entity Relation Diagram



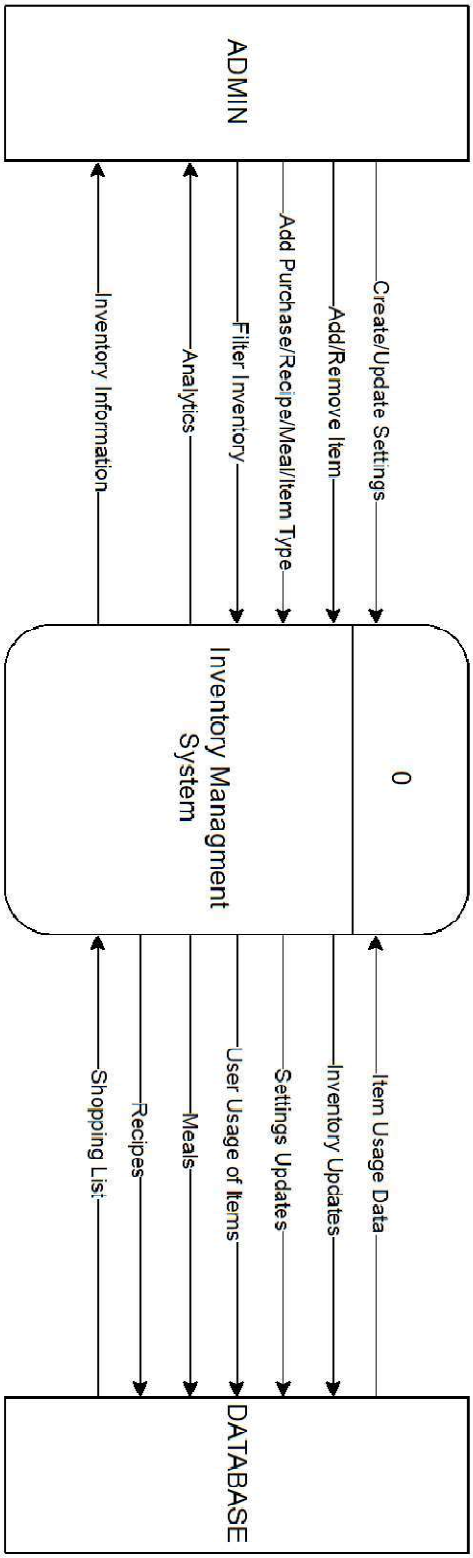
Relational Model



HIPO Model


Main Screen
Subscreen View
Generate Shopping List
For Time Period
Settings
Add
Delete
View
Storage
Item Type
Location
User
Users
Items colcured red can be accessed by both the Admin (Parent) and User (Dependent). Items colcured green can only be accessed by the Admin.
Inventory
Recipes
Meals
History
Add
Delete
Create
Delete
Create
Delete
Add Ingredient
Add Instructions
Add RecipeItem

Data Flow Diagram



Appendix B – Sample Data

The following is code from `build_demo_database()` in `Database.py` truncated to show only the values used as sample data. Please see `Database.py` line 290 for context. Data for Recipes, Meals, History, Purchases and Analytics were generated by *using the application* with the below data i.e. data for Recipes, History, etcetera was created by clicking "Add Recipe," "Consume," etcetera.

```
foods = [
    {"name":"Banana", "unit":""},
    {"name":"Potato", "unit":""},
    {"name":"Soup", "unit":"L"},
    {"name":"Milk", "unit":"L"},
    {"name":"Squash", "unit":""},
    {"name":"Spaghetti", "unit":"g"},
    {"name":"Pumpkin", "unit":"g"},
    {"name":"Ground Beef", "unit":"g"},
    {"name":"Goldfish", "unit":"g"},
    {"name":"Watermelon", "unit":""},
    {"name":"Cheddar", "unit":"g"},
    {"name":"Salmon", "unit":"g"},
    {"name":"Haggis", "unit":"kg"},
    {"name":"Rice", "unit":"g"},
    {"name":"Chocolate-Chip Cookie", "unit":""},
    {"name":"Flour", "unit":"g"},
    {"name":"Penne", "unit":"g"},
    {"name":"Peanut", "unit":"g"},
    {"name":"Carrot", "unit":"g"},
]

notfoods = [
    {"name":"Advil", "unit":"caps"},
    {"name":"Wood glue", "unit":"L"},
    {"name":"Toilet paper", "unit":""},
    {"name":"TidePods", "unit":""},
    {"name":"Handsoap", "unit":"L"},
]

durables = [
    {"name":"Hammer", "unit":""},
]

for loc in ["Home", "Cabin"]:
    self.db_actions.add_location(name=loc)
```

```
dry_storages = [
    {"storage_name":"Cupboard", "location_name":"Home", "capacity":0.1},
    {"storage_name":"Cellar", "location_name":"Cabin", "capacity":0.8},
    {"storage_name":"Pantry", "location_name":"Home", "capacity":0.5},
    {"storage_name":"Basement Shelves", "location_name":"Home", "capacity":0.7}
]
fridge_storages = [
    {"storage_name":"Kitchen Fridge", "location_name":"Home", "capacity":0.65},
    {"storage_name":"Wine Fridge", "location_name":"Home", "capacity":0.3}
]
freezer_storages = [
    {"storage_name":"Kitchen Freezer", "location_name":"Home", "capacity":0.65},
    {"storage_name":"Deep Freezer", "location_name":"Home", "capacity":0.84}
]

parents = [ "John (Admin)", "Penny (Admin)", "Jaquise (Admin)" ]
dependents = [ "Harry", "Han Solo", "Sarah" ]

inventory_items = [
    {"item_name":"Carrot", "storage_name":"Kitchen Fridge", "quantity":800,
"expiry":dt.datetime.now() + dt.timedelta(days=30)},
    {"item_name":"Milk", "storage_name":"Kitchen Fridge", "quantity":3.5,
"expiry":dt.datetime.now() + dt.timedelta(days=14, hours=19)},
    {"item_name":"Goldfish", "storage_name":"Pantry", "quantity":9000,
"expiry":dt.datetime.now() + dt.timedelta(days=2000)},
    {"item_name":"Rice", "storage_name":"Basement Shelves", "quantity":2300,
"expiry":None},
    {"item_name":"Pumpkin", "storage_name":"Kitchen Fridge", "quantity":150,
"expiry":dt.datetime.now() + dt.timedelta(days=4)},
    {"item_name":"Hammer", "storage_name":"Basement Shelves", "quantity":1},
    {"item_name":"Toilet paper", "storage_name":"Basement Shelves",
"quantity":33},
    {"item_name":"TidePods", "storage_name":"Basement Shelves", "quantity":90},
    {"item_name":"Ground Beef", "storage_name":"Deep Freezer", "quantity":500,
"expiry":dt.datetime.now() + dt.timedelta(days=8)},
    {"item_name":"Chocolate-Chip Cookie", "storage_name":"Pantry", "quantity":13,
"expiry":dt.datetime.now() + dt.timedelta(days=21)},
    {"item_name":"Potato", "storage_name":"Wine Fridge", "quantity":1},
    {"item_name":"Salmon", "storage_name":"Deep Freezer", "quantity":650,
"expiry":dt.datetime.now() + dt.timedelta(days=11)},
    {"item_name":"Soup", "storage_name":"Deep Freezer", "quantity":1.6667,
"expiry":dt.datetime.now() + dt.timedelta(days=45)},
    {"item_name":"Watermellon", "storage_name":"Kitchen Fridge", "quantity":1,
"expiry":dt.datetime.now() + dt.timedelta(days=22)},
    {"item_name":"Spaghetti", "storage_name":"Cupboard", "quantity":800},
```

```
    {"item_name":"Haggis", "storage_name":"Kitchen Freezer", "quantity":4,
"expiry":dt.datetime.now() + dt.timedelta(days=29)},
    {"item_name":"Potato", "storage_name":"Cellar", "quantity":46467},
    {"item_name":"Cheddar", "storage_name":"Kitchen Fridge", "quantity":500,
"expiry":dt.datetime.now() + dt.timedelta(days=30)},
    {"item_name":"Advil", "storage_name":"Cupboard", "quantity":120},
    {"item_name":"Squash", "storage_name":"Kitchen Fridge", "quantity":1,
"expiry":dt.datetime.now() + dt.timedelta(days=13)},
]
```