

Data Models and more V1.1

Apr 2016

By: Mark Prucha, and Bilal Adra

This is the first draft of the data model and site architectural layout as we envision.

We have broken our data into three tables.

- User – Table to model all attributes of a user
- Pickup- Table to model all pickup related information (including scheduled/unscheduled pickups as well as donor information)
- Day- Table to model all days which have already been deemed as available for pickups.

The sections of this document are as follows:

1. User Data Model
2. Pickup Data Model
3. Day Data Model
4. Flow of Events
5. Named Routes
6. Issues

Feedback can be slacked, or you can email us at mgprucha@noctrl.edu (mgprucha on slack) and bhadra@noctrl.edu (bilaladra on slack)

Changes from 1.0 are highlighted in purple with yellow text

Red highlights mean we were not sure if we need it, but we think we should delete whatever it is

1. User Data Model

| User Data Model | |
|------------------------|---------|
| Column | Type |
| <u>id</u> ¹ | int |
| email | string |
| password_digest | string |
| username | string |
| level | enum |
| superAdmin | boolean |

- A user will have a unique integer id which will serve as the key to the User table. This is generated by Rails when a new user is added.
- The username is a unique string determined by the admin when adding the user. The username will be used at login. (i.e. mprucha).
- The user will have an email address which will be a string in the form: `<name>@<domain>` (i.e. mgprucha@noctrl.edu).
- A user's password need not be unique. Restrictions on password can be determined by the programmer in charge of user signup.
- The user's level will be an enum (stored in db as int). The correspondence of integers to level types is shown below. (more information on enums can be found here in the ruby on rails api <http://api.rubyonrails.org/> , search enum if you dont see it right away its the first one)
 - 0 -> Entry
 - 1 -> Standard

¹ Column names with an underline denote table key

- 2 -> Admin

- Super Admin

- This admin is not creatable by the users, only by actively accessing the database. It has the power to delete and manage admin accounts as well as all other admin powers

Example User:

| | | |
|-----------------|----|----------------------|
| id | -> | 1 |
| username | -> | kriegerRulez |
| user_email | -> | wtkrieger@noctrl.edu |
| Password_digest | -> | adlkf;jbin2l09 |
| level | -> | 2 |
| superAdmin | -> | true |

2. Pickup Data Model

| Pickup Data Model | |
|--|-----------|
| Column | Data type |
| <u>id</u> | int |
| day_id | int |
| donor_title | string |
| donor_first_name ² | string |
| donor_last_name | string |
| donor_spouse_name ³ | string |
| donor_address_line1 | string |
| donor_address_line2 | string |
| donor_city | string |
| donor_zip | string |
| donor_dwelling_type | string |
| donor_location_instructions ⁴ | string |
| donor_phone | string |
| donor_email | string |
| number_of_items | int |

² Broke up donor_name into 2 separate attributes: donor_first_name and donor_last_name

³ Added donor_spouse_name

⁴ Removing donor_location_instructions unless they are needed somewhere

| | |
|--------------------------|---|
| item_notes ⁵ | text |
| donor_notes ⁶ | text |
| rejected | boolean |
| rejected_reason | string |
| created_at ⁷ | datetime (added automatically by rails) |

- Each pickup will have a unique id created when the pickup is created.
- Day_ID is the id of the corresponding Day row. An unscheduled pickup has a Day_ID of null where a scheduled pickup has a Day_ID that is defined. To get the date of the pickup simply get the date from the corresponding Day in the Day table.
- Because we are not interested in keeping track of how many donations a donor has made, it is easier to just add donor information as attributes to the pickup table.
- Donor first name, donor last name, address, city, zip, dwelling type, location instructions, phone, and email are all strings entered on the /pickup/new screen.
- Donor phone will be a 10-character string of numbers (i.e. 6301234567)
- Number of items is an integer.
- Both item_notes and donor_notes are type text. This is because the default number of characters a string can hold is 255.
- The Number of Pickups defaults to 0

Example Pickup:

```

id                -> 25
scheduled         -> true
donor_title       -> Mr.
donor_first_name  -> Bob
Donor last name   -> Bobson
donor_address_line1 -> 30 N Brainard St

```

⁵ Previously item_description

⁶ Previously other_notes

⁷ Added by default in rails, only shown here for clarity's sake

| | | |
|---------------------|----|---|
| donor_address_line2 | -> | null |
| donor_city | -> | Naperville |
| donor_zip | -> | 60540 |
| donor_dwelling_type | -> | House |
| donor_phone | -> | 6301234567 |
| donor_email | -> | bob@noctrl.edu |
| number_of_items | -> | 1 |
| item_notes | -> | A 55" Sony Bravia TV |
| donor_notes | -> | Please ring the door bell, then knock, then donor will answer the door |
| rejected | -> | true |
| rejected_reason | -> | only plays this video: https://www.youtube.com/watch?v=dQw4w9WgXcQ |
| track_donor | -> | true |

3. Day Data Model

| Day | |
|-------------------|-----------|
| Column | Data type |
| <u>id</u> | int |
| date | date |
| number_of_pickups | int |

- The Day table will hold the date and number of pickups scheduled for that date and the date.
- When rails converts date to a string (ie date.to_s) it is of the form: YYYY-MM-DD, you can get variations using the strftime method whose documentation can be found [here](#)

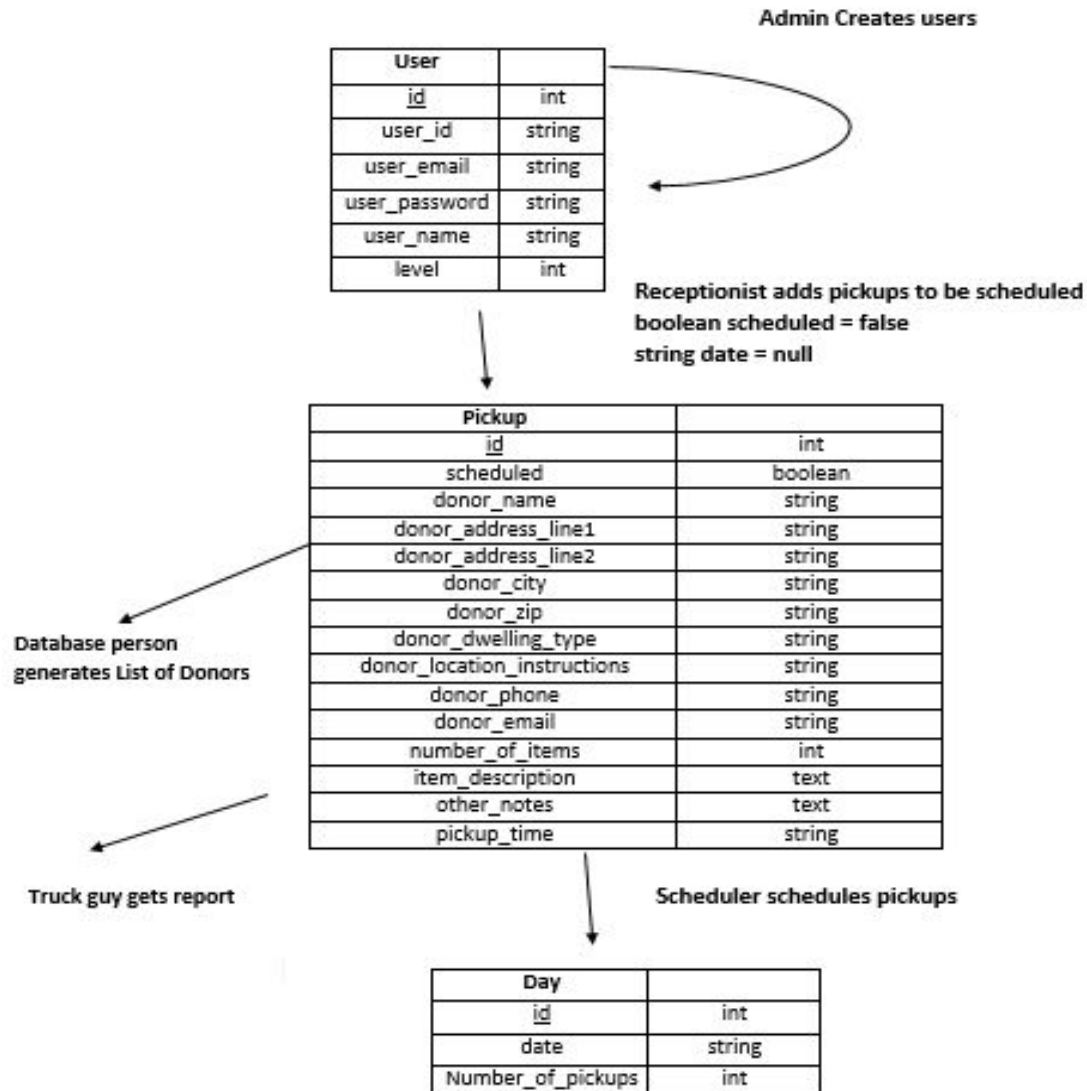
Example Day:

```
id           -> 35
date        -> 2016-05-28
number_of_pickups -> 4
```

4. Flow of Events

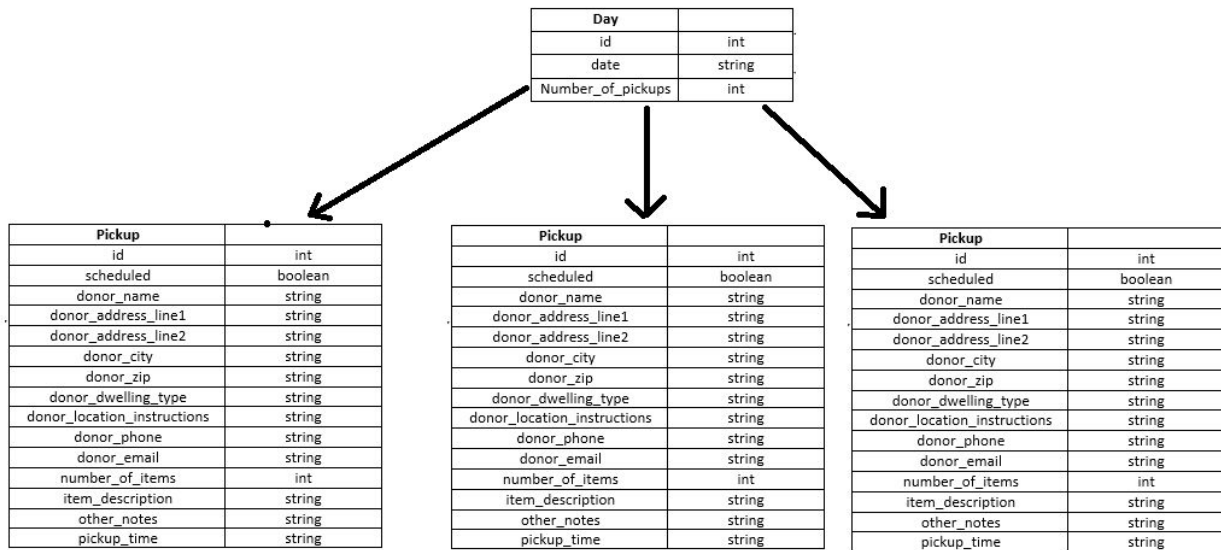
A typical flow of events and their relation to our data models is shown in the figure below.

2) Flow of Events



***Flow of events does not display the updated tables**

Day and Pickup objects have a one-to-many relationship as shown in the figure below.



Notes:

- A Day has many pickups, but a pickup has at most (belongs to) one day.
- The Bullpen consists of the all pickups whose 'scheduled' attribute is equal to true (or whose date attribute is equal to null).
- When a pickup is added to the bullpen, it is created in the pickup table with scheduled equal to false and date equal to null.
- A list of scheduled pickups can be accessed by accessing all the pickups in the pickups table whose scheduled attribute is equal to true (or whose date is not null).
- When a pickup is rescheduled for a different date, its current date must be used to access the matching date in the Day table. The number of pickups for that date must be decremented by one. Its current date must then change to the newly scheduled date.

5. Named Routes

| Name | Path | Description |
|------------|-------------------------|---|
| Login | root | The first screen |
| Home.1 | /pickups | Bullpen |
| Home.2 | /pickup/new | New pickup form |
| Home.3 | /pickup/<PickupID>/edit | view/edit/schedule pickup |
| Schedule.1 | /day/index | List pickup days |
| Schedule.2 | /day/new | New pickup days |
| Schedule.3 | /day/<DayID> | List pickups on day |
| Report.1 | /reports | List reports |
| Report.2 | /reports/donors | Generates reports for donors |
| Report.3 | /reports/truck | Generates truck report |
| Admin.1 | /users | Lists all users |
| Admin.2 | /users/new | Add a new users |
| Admin.3 | /users/<UserID>/edit | Edit a user |
| About | /about | A riveting page detailing our journey and great accomplishments |

Notes:

- <DayID> = the id of the Day object stored in the database
- <UserID> = the id of the Day object stored in the database
- <PickupID> = the id of the Pickup object stored in the database
- Home.1 will have two paths: one as the root path and one at /pickup/index

In routes.rb, we will have the following resources:

1. Users
2. Pickups
3. Days

6. Issues

Should we have a track donor field? Prof Bill is finding out.