

Meals on Wheels Staff and Volunteer Web Portal



Max Patek, Nate Strawser, Josh Santana, Alex Hicks, Kyle Leisure,
Michael Benos, Kevin Naddoni

Meals on Wheels -- Requirements Overview

- Customers
 - Receive meals on a delivery route
- Volunteers do jobs
 - Volunteer have recurring jobs (deliver on a route, packing meals, etc)
 - Volunteer can fill and request substitutions
- Staff
 - Assign jobs to volunteers and coordinate substitutions
 - Packing and delivering meals
 - Manage volunteer and customer data (payments, diets, pets)
 - Generate various reports (billing, daily meal counts, food labels)
- *(and many more specific requirements)*

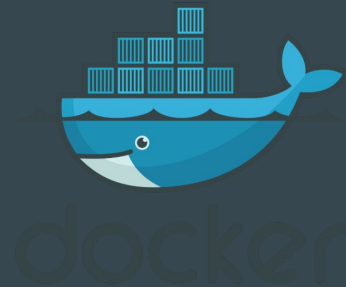
Technologies

- Django
- Python
- Docker
- Docker-Compose
- Bootstrap
- Gunicorn
- Nginx
- Letsencrypt
- Postgres
- Autoformatters
- Google Maps API



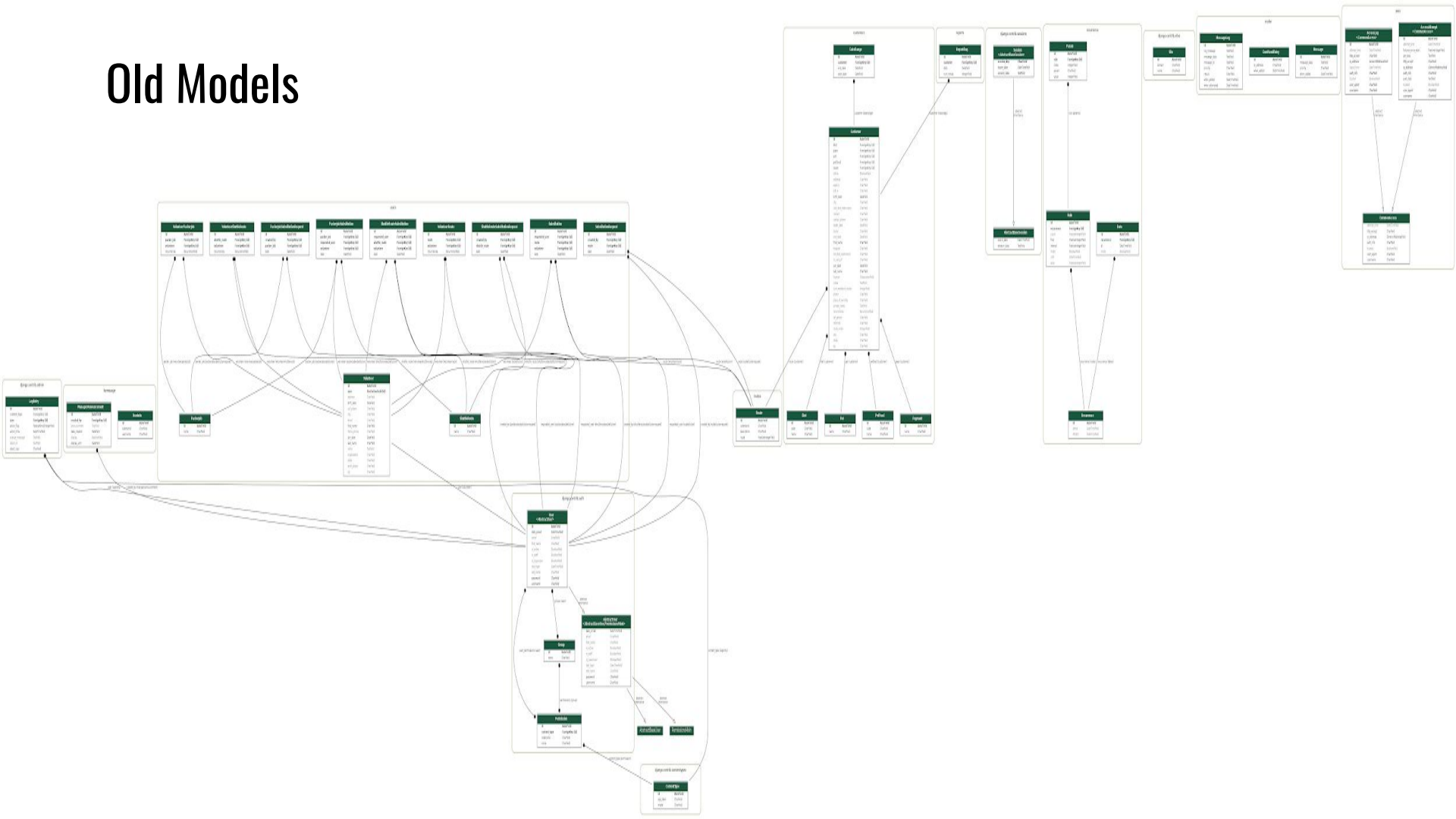
Tools

- Travis CI
- Github Actions
- Github Issues and Github Projects
- Amazon Web Services
- Make
- Coverage.py

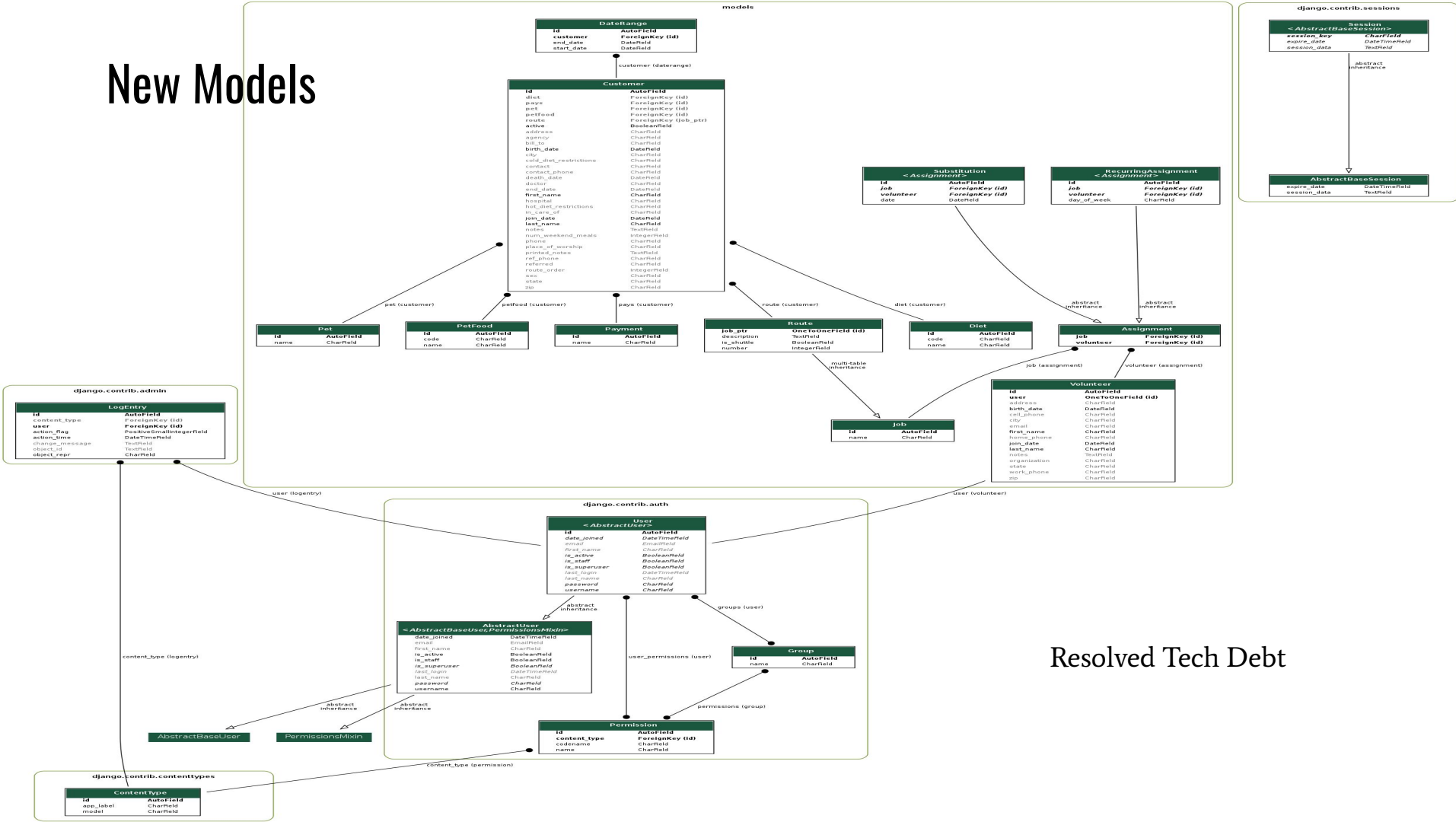


What is good?
(wins)

Old Models



New Models



Fewer Pages (demo)

- More concise models → fewer pages
- All job types (route, packer, shuttle) can now be represented on a single “Manage Jobs” page
- Substitutions and assignments are more generalized (no separate pages for filled/unfilled, or job types)
- Fewer reports for volunteers to generate each day
- Site navigation is much easier

Google Maps Integration (demo)

- Route Maps
 - Route manipulation
 - Directions
 - Google Maps app support
- Addresses
 - Autocomplete



Assignment Management (demo)

[Meals on Wheels](#)[Customer Management ▾](#)[Volunteer Management ▾](#)[Manage Jobs](#)[Manage Substitutions](#)[Manage Assignments](#)[Reports ▾](#)[Logout](#)[✉ Email shown volunteers](#)[Create Assignment](#)

Manage Assignments

Filter:

Volunteer ⬆⬇⬆

Job ⬆⬇⬆

Assignment(s)

Volunteer 166

Route 1

3rd Mon

Volunteer 422

Route 1

Wednesdays

Volunteer 246

Route 1

Fridays

Volunteer 393

Route 1

1st Mon, 2nd Mon, 4th Mon, 5th Mon

Volunteer 380

Route 2

1st Mon, 1st Wed, 2nd Wed, 3rd Mon, 5th Mon

Volunteer 240

Route 2

3rd Wed, 4th Wed, 5th Wed

Volunteer 531

Route 2

Fridays

Reports (demo)

- Hardcopy PDFs
 - handed to volunteers (*food labels, route sheets*)
 - given to the board (*billing, daily count*)
 - kept for historical records (*job overview*)
 - Hourly cron job creates up-to-date records → guarantees historical accuracy
- Faster report generation
 - Leverage batch processing in cron job
 - Django's query aggregation

What was hard?

(challenges)

Old Portal

- Combination of greenfield and brownfield development
 - Worst of both worlds
 - Had to understand old codebase while writing our webapp from scratch
 - 12k+ lines of Python
- Logistical difficulties of working with an existing live application
 - Track down old API keys
 - DNS Records
 - Data migration

Migrating

- Old Data
 - Fragile server
 - Risky data dumps
 - Incompatible Django & fixtures
- New Relation Model
 - Complex migration scripts
 - Topological sort by foreign keys
- Strip Unused Data
 - Open Route
 - 100,000s rows of outdated data
- Migration Utility
 - `python3 manage.py importlegacy --no-coordinates`
 - `python3 hack_fixtures.py`

```
class LegacyRoute(models.Model):
    route = models.IntegerField()
    calendarId = models.CharField(db_column="calendarId", max_length=200)
    description = models.CharField(max_length=400)

    def _import(self):
        route_type, _ = JobType.objects.get_or_create(name=ROUTE_TYPE_NAME)
        Route.objects.create(
            name=f"Route {self.route}",
            number=self.route,
            description=self.description,
            job_type=route_type,
        )
```

Models Changes

- Overly optimistic in the very early design predictions
- September to October customer meetings → important to learning their system
- Understanding MOW users workflow
 - Customers on specific days
 - Routes on specific days
 - Jobs with multiple volunteers
 - Other exemptions, exceptions that needed to be designed into new system
- Proactively incorporated customer input → ultimately not a major development challenge

Reports Changes

- Strict formatting requirements
- Labels were printed on a specific sized paper
- Working with a new library whtmltopdf

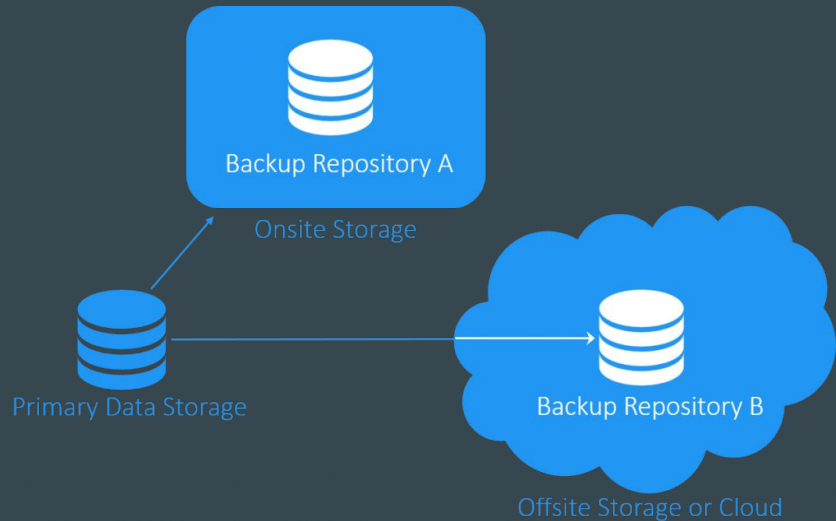
Recurrence Representations

- Customers only receive meals sometimes
 - Active?
 - DateRange defined?
 - Recurrence field?
- Jobs only run on certain days
 - Recurrence field?
 - Anyone actually assigned?
- Volunteers only work certain days
 - Every Monday?
 - Fourth Monday?
 - Fifth Monday?
 - *Last Monday?*
- Existing django packages
 - No support for restrictions
 - Not maintained
 - Conflict with others
 - Like django-address
- Built our own
 - Use existing for customers
 - Use existing for routes
 - Use custom for volunteers

What's next?

Future Work

- Maintainability
- Data Security
- Mobile friendliness
 - Functional, but not optimized



 **amazon** web services™ | **Glacier**



Thank you!

We'd like to thank the course staff and the team at Meals on Wheels (specifically Allie and Susan) for all of their help.

Any questions?