!Java presents

Schedule Master v1.0

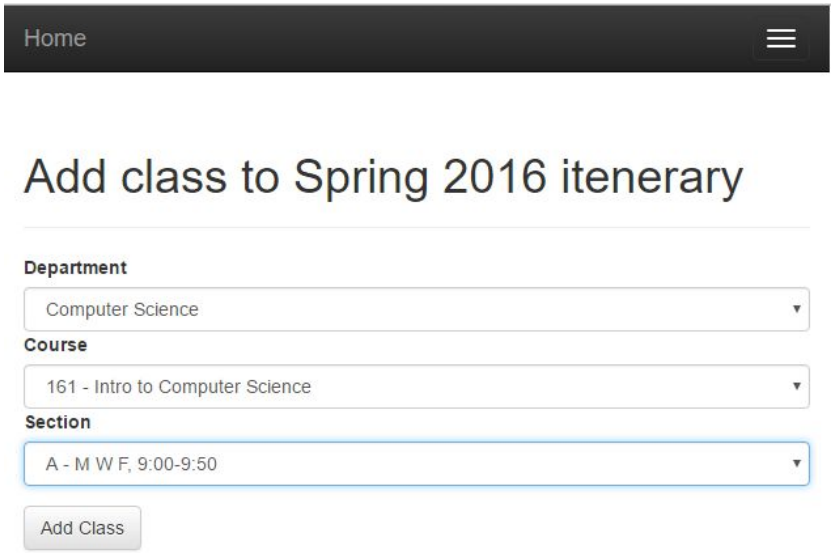
Sam Burdick
Bridger Dunn
Hunter Kurtis
Jahrme Risner

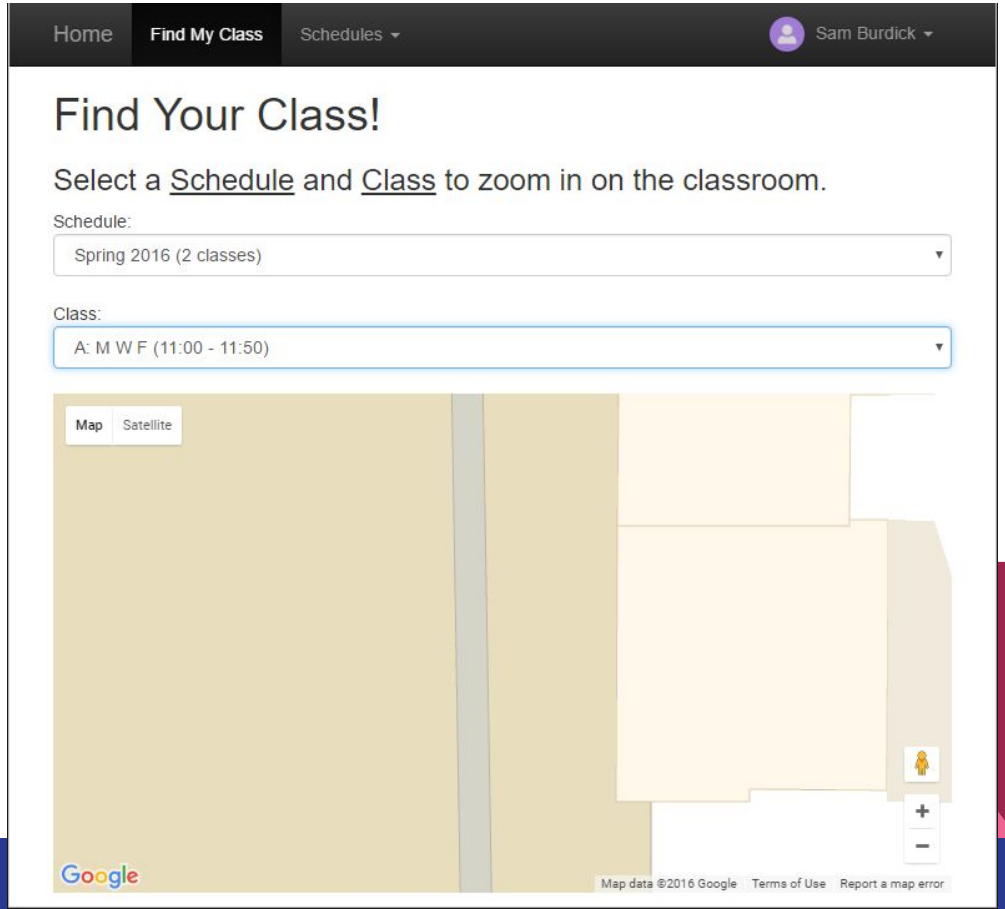
The Problem

- Students mindlessly wandering Thompson hallways
- Lack of knowledge of classroom locations
- Disorganized first year students

What is Schedule Master?

- It is an SAAS that allows students to create schedule, add classes
- Then, students can view their classes on an interactive map.





Development Stories

As a user, I want to…

1. Create Schedule, Add Class
2. View Class on Map
3. Share Schedule
4. View Schedule on Calendar

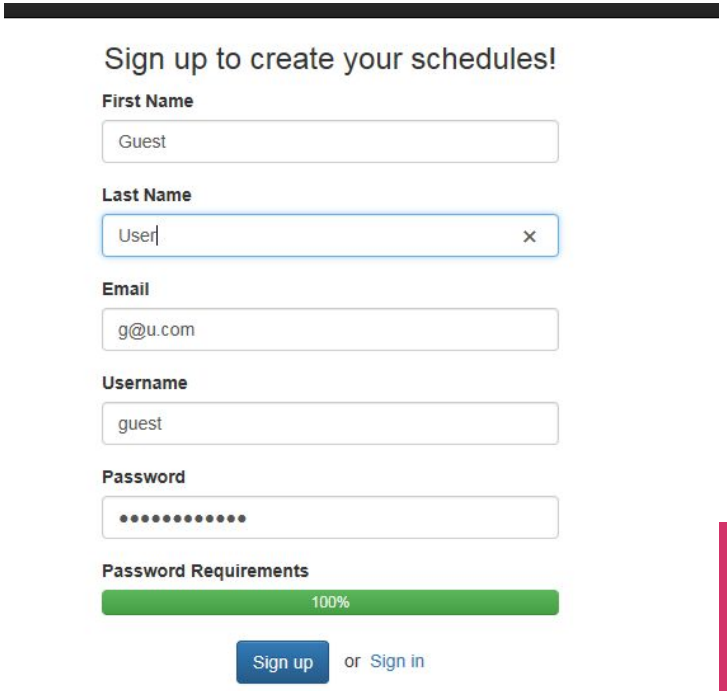
Developer story:

- All of the term and class data is stored in a Mongo database.
- This data is subject to change, so I want a flexible, modular database structure.

App Demo

Volunteer?

Url: http://not-java.herokuapp.com/



Password is:

Password123!

Create New Schedule, Add Class

Home

New schedule

Term

Spring 2016

Create

Home

Add class to Spring 2016 itenerary

Department

Computer Science

Course

161 - Intro to Computer Science

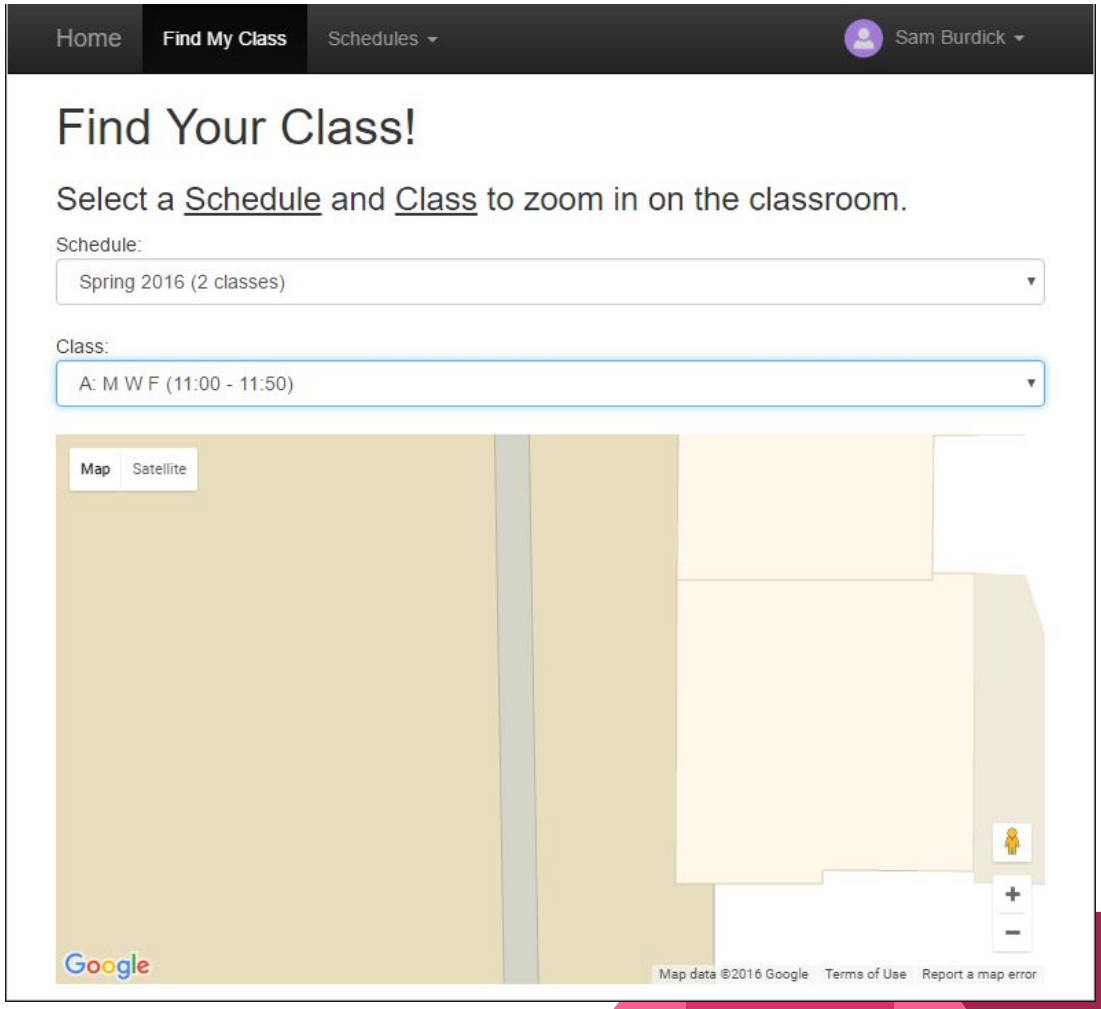
Section

A - M W F, 9:00-9:50

Add Class

Map

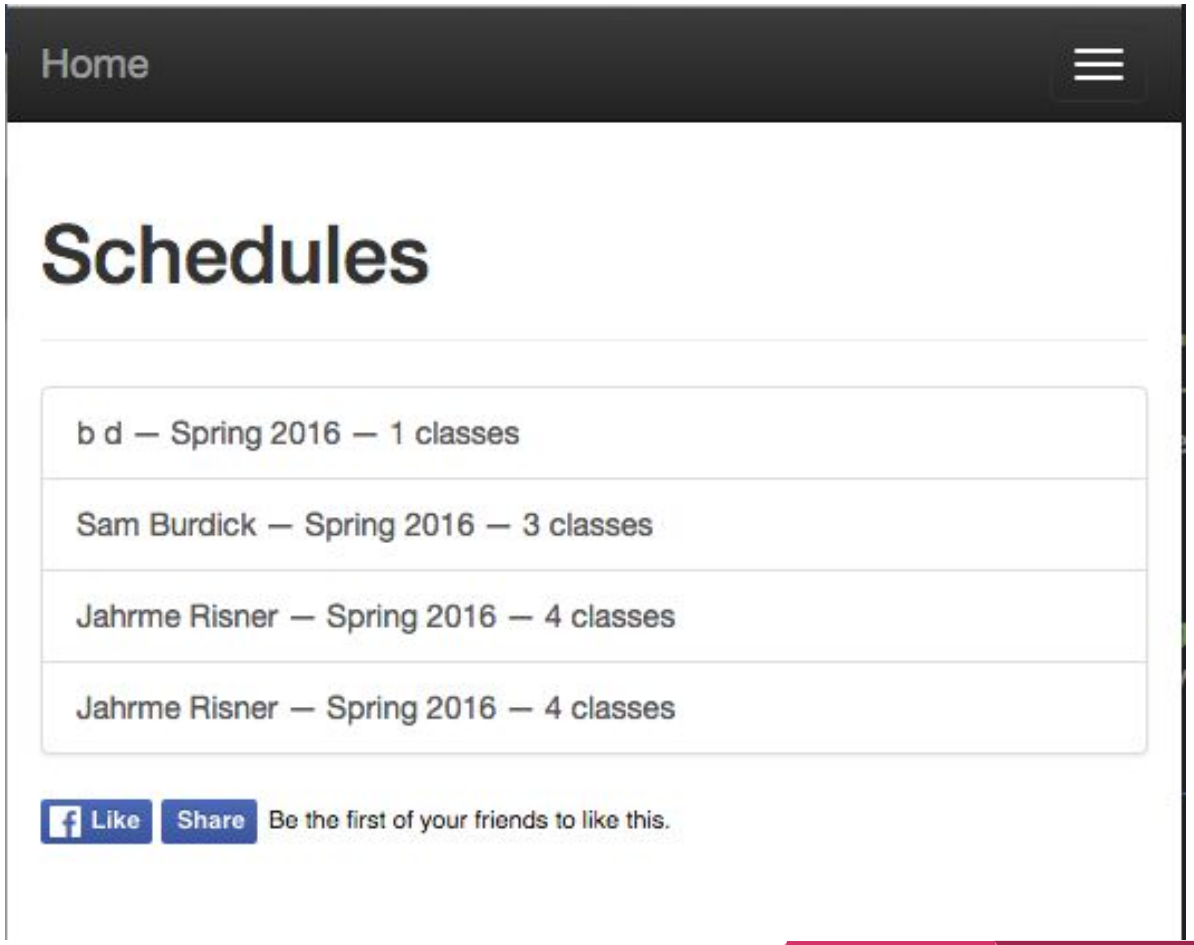
In order to find their classroom, the user can select their schedule, and upon choosing a class, the map zooms in on the location associated with the class.



Share Button

Allows users to share their schedule on Facebook

Sometimes appears on the bottom of the View Schedule page



MongoDB: ObjectIds

In the database, documents reference other documents by their ObjectId.

Observe that departments are referenced in this way: (as viewed in RoboMongo)

Key	Value	Type
(1) ObjectId("5724e418e1822056a9...	{ 3 fields }	Object
_id	ObjectId("5724e418e1822056a943a0db")	ObjectId
termName	Spring 2016	String
departments	[1 element]	Array
[0]	ObjectId("5724e4b0e1822056a943a0dc")	ObjectId

In this example, 'ObjectId("5724e4b0e1822056a943a0dc")' references the Computer Science department.

The upshot: if one attribute of a document is changed, then other documents referencing it can link to the new data. That way we don't have to 'hard-code' every attribute in the documents, just each attribute's ID.

The Database

Hierarchy: Term > Department > Course > Section > Location, Instructor

Term requires: departments

Department requires: courses

Courses require: sections

Section requires: locations, instructors

Location/Instructor: stored separately

Computer Science Department & Courses Table

Key	Value	Type
(1) ObjectId("5724e4b0e1822056a9...	{ 4 fields }	Object
_id	ObjectId("5724e4b0e1822056a943a0dc")	ObjectId
name	Computer Science	String
courses	[13 elements]	Array
[0]	ObjectId("5724d166e1822056a943a0bd")	ObjectId
[1]	ObjectId("5724d347e1822056a943a0bf")	ObjectId
[2]	ObjectId("5724d44fe1822056a943a0c3")	ObjectId
[3]	ObjectId("5724d553e1822056a943a0c6")	ObjectId
[4]	ObjectId("5724d604e1822056a943a0c8")	ObjectId
[5]	ObjectId("5724d64ce1822056a943a0ca")	ObjectId
[6]	ObjectId("5724d6f9e1822056a943a0cd")	ObjectId
[7]	ObjectId("5724d767e1822056a943a0cf")	ObjectId
[8]	ObjectId("5724d7e8e1822056a943a0d2")	ObjectId
[9]	ObjectId("5724d867e1822056a943a0d4")	ObjectId
[10]	ObjectId("5724da4ce1822056a943a0d6")	ObjectId
[11]	ObjectId("5724dae9e1822056a943a0d8")	ObjectId
[12]	ObjectId("5724db36e1822056a943a0da")	ObjectId
term	ObjectId("5724e418e1822056a943a0db")	ObjectId

Computer Science Course table

db.getCollection('courses').find({})

local localhost:27017 mean-dev

```
db.getCollection('courses').find({})
```

courses 0.001 sec. 0 50

Key	Value	Type
(1) ObjectId("5724d166e1822056a9...	{ 4 fields }	Object
(2) ObjectId("5724d347e1822056a9...	{ 4 fields }	Object
(3) ObjectId("5724d44fe1822056a9...	{ 4 fields }	Object
(4) ObjectId("5724d553e1822056a9...	{ 4 fields }	Object
_id	ObjectId("5724d553e1822056a943a0c6")	ObjectId
name	Software Engineering	String
courseNumber	240	Double
sections	[1 element]	Array
(5) ObjectId("5724d604e1822056a9...	{ 4 fields }	Object
(6) ObjectId("5724d64ce1822056a9...	{ 4 fields }	Object
(7) ObjectId("5724d6f9e1822056a9...	{ 4 fields }	Object
(8) ObjectId("5724d767e1822056a9...	{ 4 fields }	Object
(9) ObjectId("5724d7e8e1822056a9...	{ 4 fields }	Object
(10) ObjectId("5724d867e1822056a...	{ 4 fields }	Object
(11) ObjectId("5724da4ce1822056a...	{ 4 fields }	Object
(12) ObjectId("5724dae9e1822056a...	{ 4 fields }	Object
(13) ObjectId("5724db36e1822056a...	{ 4 fields }	Object

Software Engineering Section table

Key	Value	Type
(1) ObjectId("5723cab2e1822056a9...	{ 5 fields }	Object
(2) ObjectId("5724d077e1822056a9...	{ 5 fields }	Object
(3) ObjectId("5724d09ae1822056a9...	{ 5 fields }	Object
(4) ObjectId("5724d31ae1822056a9...	{ 5 fields }	Object
(5) ObjectId("5724d3a8e1822056a9...	{ 5 fields }	Object
(6) ObjectId("5724d3d4e1822056a9...	{ 5 fields }	Object
(7) ObjectId("5724d400e1822056a9...	{ 5 fields }	Object
(8) ObjectId("5724d518e1822056a9...	{ 5 fields }	Object
_id	ObjectId("5724d518e1822056a943a0c5")	ObjectId
name	A	String
instructor	ObjectId("57238ea4e1822056a943a0b3")	ObjectId
location	ObjectId("5724d499e1822056a943a0c4")	ObjectId
times	{ 2 fields }	Object
(9) ObjectId("5724d5dde1822056a9...	{ 5 fields }	Object
(10) ObjectId("5724d638e1822056a...	{ 5 fields }	Object
(11) ObjectId("5724d6b5e1822056a...	{ 5 fields }	Object
(12) ObjectId("5724d734e1822056a...	{ 5 fields }	Object
(13) ObjectId("5724d7bbe1822056a...	{ 5 fields }	Object
(14) ObjectId("5724d83ce1822056a...	{ 5 fields }	Object
(15) ObjectId("5724d8a1e1822056a...	{ 5 fields }	Object
(16) ObjectId("5724dab3e1822056a...	{ 5 fields }	Object
(17) ObjectId("5724db17e1822056a...	{ 5 fields }	Object

Location table

Key	Value	Type
(1) ObjectId("57239012e1822056a9…	{ 3 fields }	Object
(2) ObjectId("57239035e1822056a9…	{ 3 fields }	Object
(3) ObjectId("57239065e1822056a9…	{ 3 fields }	Object
(4) ObjectId("5724d499e1822056a9…	{ 3 fields }	Object
_id	ObjectId("5724d499e1822056a943a0c4")	ObjectId
name	Thompson 399	String
coords	[2 elements]	Array
[0]	47.263439	Double
[1]	-122.482846	Double
(5) ObjectId("5724d69ce1822056a9…	{ 3 fields }	Object
(6) ObjectId("5724d795e1822056a9…	{ 3 fields }	Object

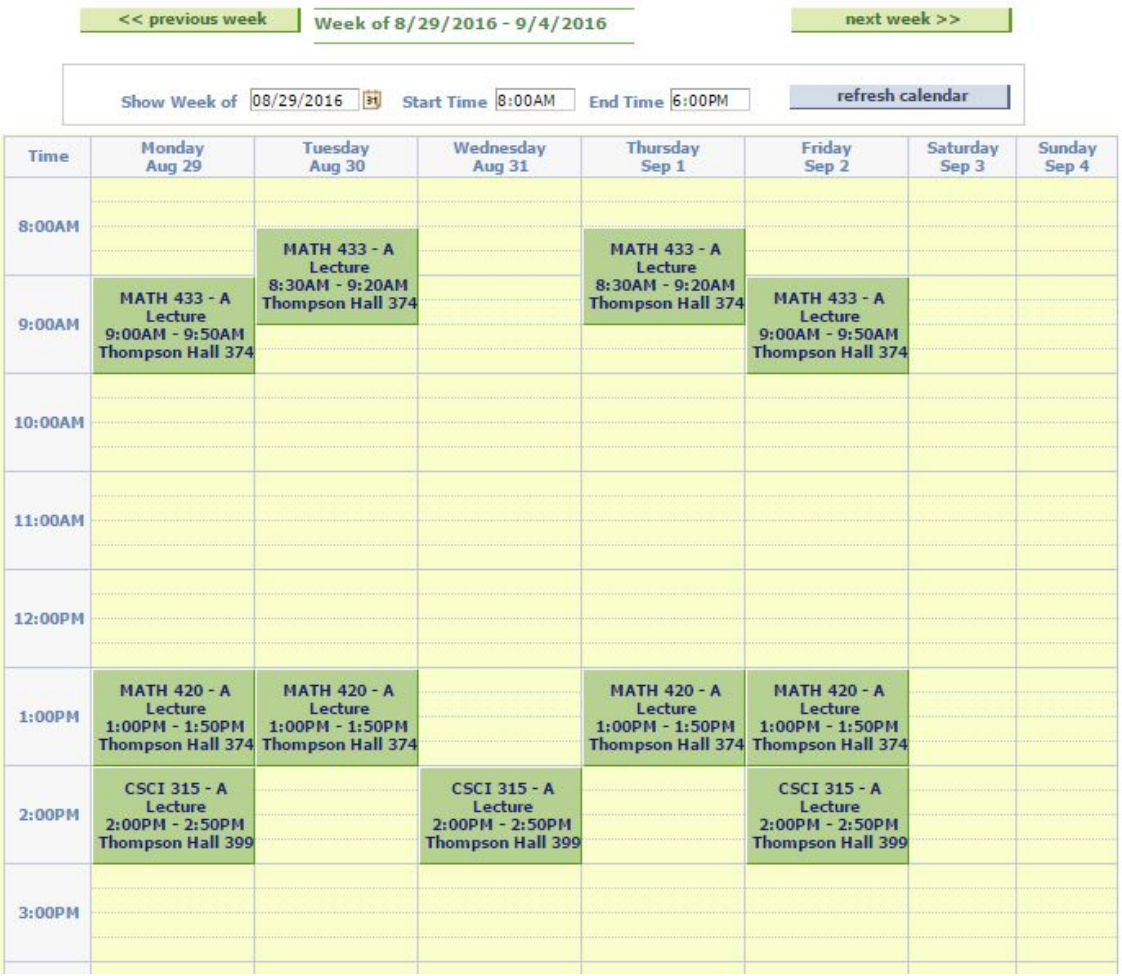
Instructor Table

Key	Value	Type
(1) ObjectId("57238de4e1822056a9...	{ 2 fields }	Object
(2) ObjectId("57238ea4e1822056a9...	{ 2 fields }	Object
_id	ObjectId("57238ea4e1822056a943a0b3")	ObjectId
name	Anthony Mullen	String
(3) ObjectId("57238eb4e1822056a9...	{ 2 fields }	Object
(4) ObjectId("57238edbe1822056a9...	{ 2 fields }	Object
(5) ObjectId("57238eebe1822056a9...	{ 2 fields }	Object

Calendar

A calendar like this one would be placed next to the map.

The user should be able to select one of his/her classes (in a box), making the location of that class appear on the map.



Difficulties

1. Consistency of Facebook share button.
   - Page refresh issues - it does not always appear

2. Ultimately, we could not find an appropriate Angular calendar module or implement our own.
3. Creating the database structure.
   - Routing the data from backend to frontend and displaying it properly

What worked well: Specialization, Agile

- Development of multiple individual modules that could be independently tested
- Frequently tested finished code, schemas
- Calendar: early feature, but scrapped towards the end
- Putting it all together

Questions?