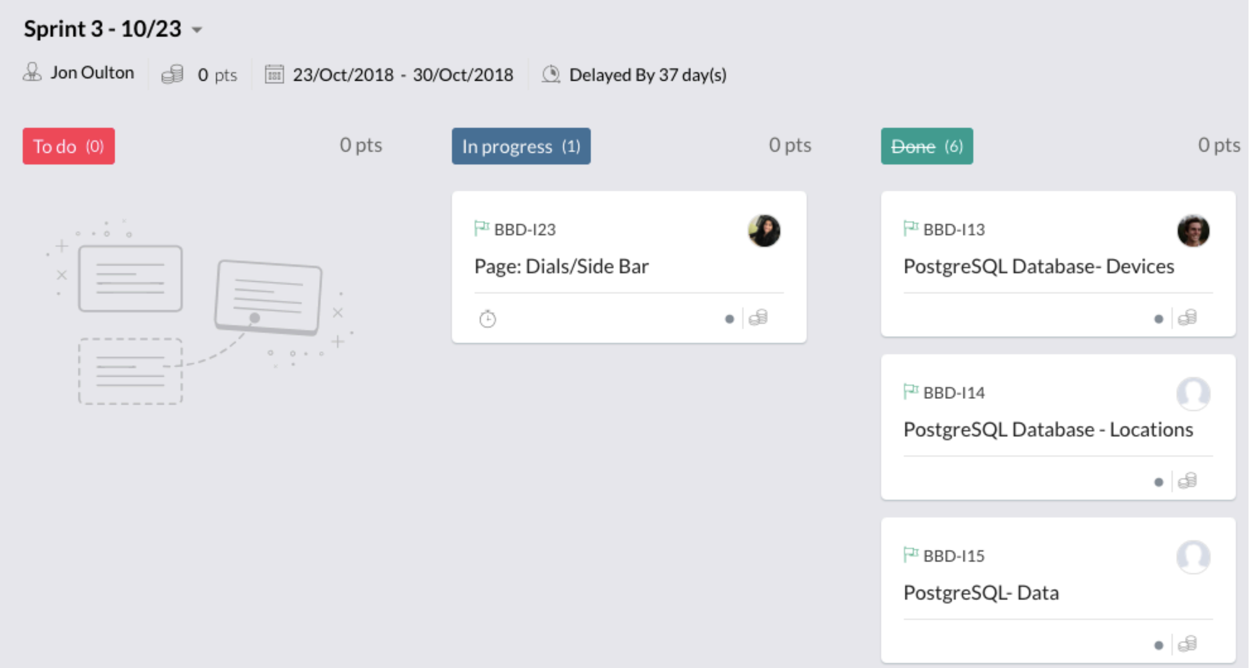
- Title: Beats by David
- Who: David Kopala, Jon Oulton, Johnson Cheung, Baljot Kaur, Alex Ho, muyu deng
- Project Tracker : Zoho



Sprint 3 - 10/23

Jon Oulton | 0 pts | 23/Oct/2018 - 30/Oct/2018 | Delayed By 37 day(s)

To do (0) — 0 pts

In progress (1) — 0 pts

BBD-I23
Page: Dials/Side Bar

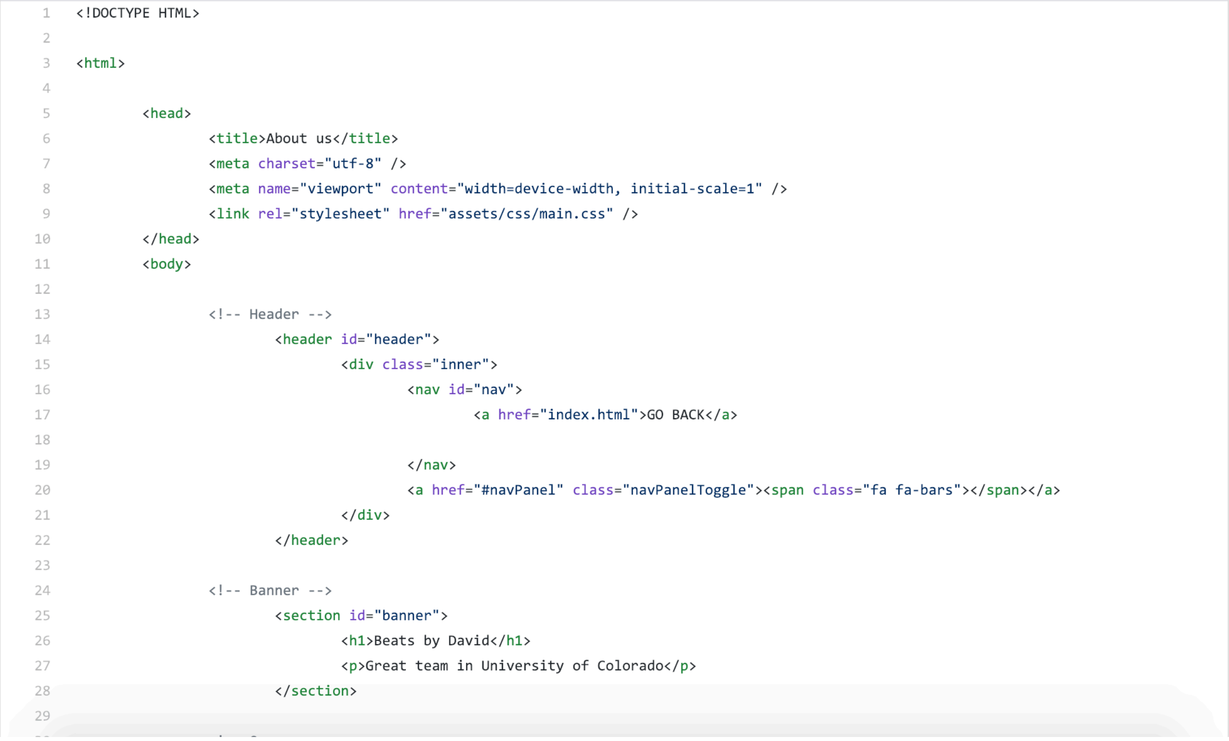
~~Done~~ (6) — 0 pts

BBD-I13
PostgreSQL Database- Devices

BBD-I14
PostgreSQL Database - Locations

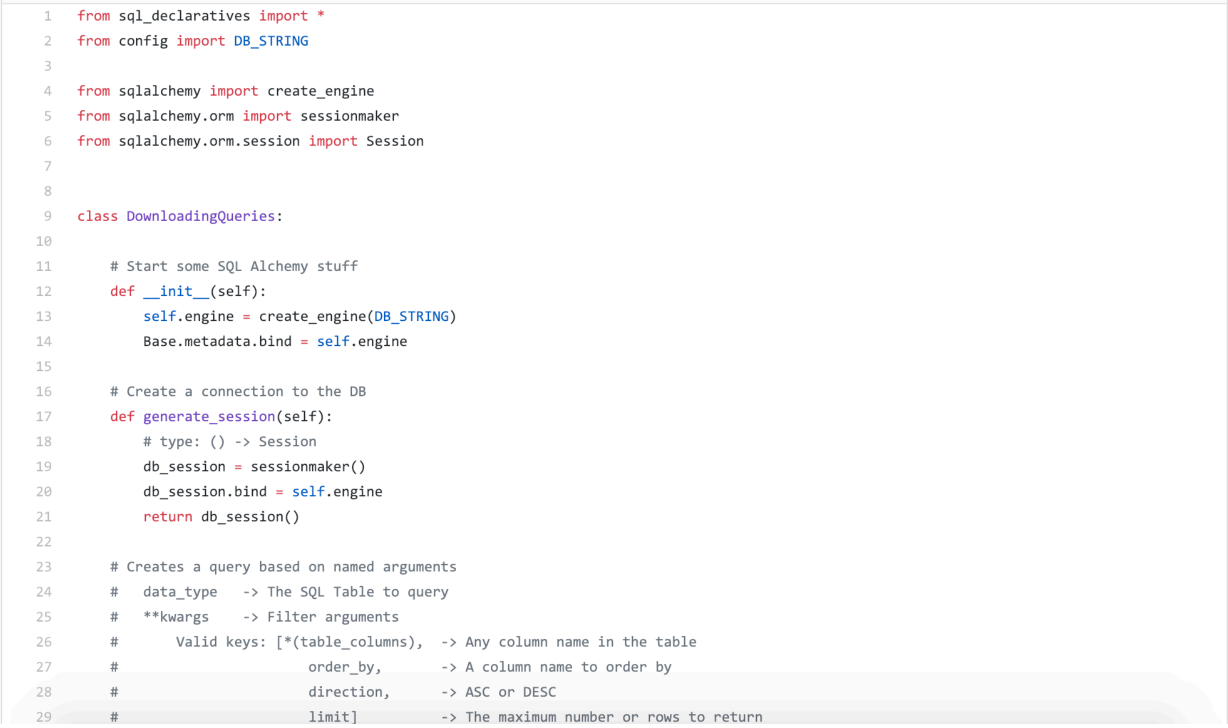
BBD-I15
PostgreSQL- Data

- VCS: https://github.com/BeatsByDavid



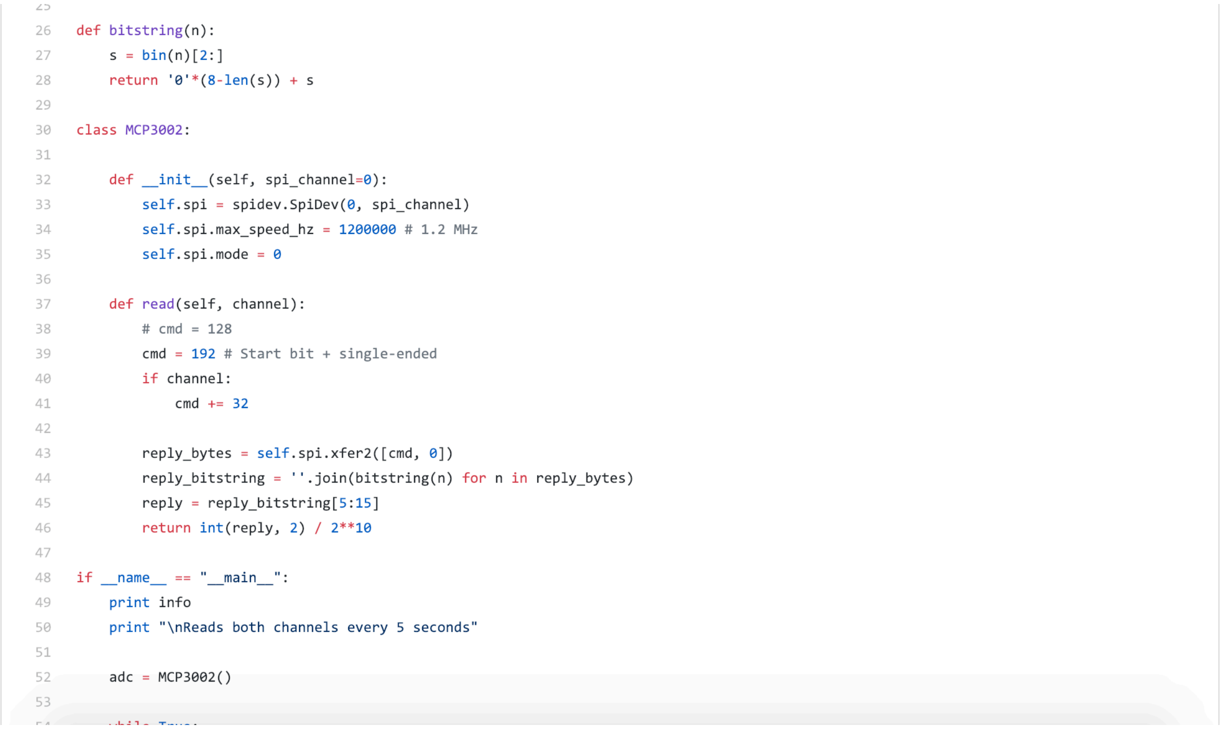
```html
<!DOCTYPE HTML>

<html>

        <head>
                <title>About us</title>
                <meta charset="utf-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1" />
                <link rel="stylesheet" href="assets/css/main.css" />
        </head>
        <body>

                <!-- Header -->
                        <header id="header">
                                <div class="inner">
                                        <nav id="nav">
                                                <a href="index.html">GO BACK</a>

                                        </nav>
                                        <a href="#navPanel" class="navPanelToggle"><span class="fa fa-bars"></span></a>
                                </div>
                        </header>

                <!-- Banner -->
                        <section id="banner">
                                <h1>Beats by David</h1>
                                <p>Great team in University of Colorado</p>
                        </section>

```

Baljot and muyu deng, worked on the front end and the user interface of the project.



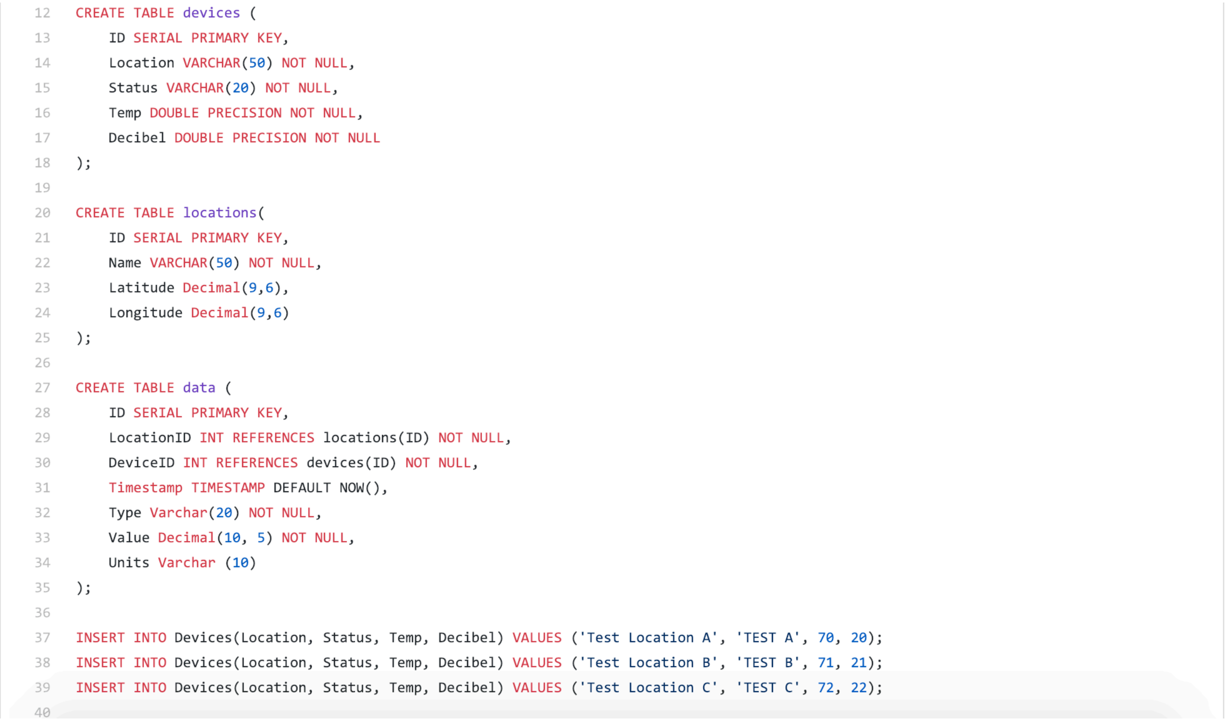
```
from sql_declaratives import *
from config import DB_STRING

from sqlalchemy import create_engine
from sqlalchemy.orm import sessionmaker
from sqlalchemy.orm.session import Session


class DownloadingQueries:

    # Start some SQL Alchemy stuff
    def __init__(self):
        self.engine = create_engine(DB_STRING)
        Base.metadata.bind = self.engine

    # Create a connection to the DB
    def generate_session(self):
        # type: () -> Session
        db_session = sessionmaker()
        db_session.bind = self.engine
        return db_session()

    # Creates a query based on named arguments
    #   data_type   -> The SQL Table to query
    #   **kwargs    -> Filter arguments
    #       Valid keys: [*(table_columns),  -> Any column name in the table
    #                   order_by,       -> A column name to order by
    #                   direction,      -> ASC or DESC
    #                   limit]          -> The maximum number or rows to return
```

Johnson Cheung, contributed the middle layer, which has the information communicate to the backend and have a history and trend.



```
def bitstring(n):
    s = bin(n)[2:]
    return '0'*(8-len(s)) + s

class MCP3002:

    def __init__(self, spi_channel=0):
        self.spi = spidev.SpiDev(0, spi_channel)
        self.spi.max_speed_hz = 1200000 # 1.2 MHz
        self.spi.mode = 0

    def read(self, channel):
        # cmd = 128
        cmd = 192 # Start bit + single-ended
        if channel:
            cmd += 32

        reply_bytes = self.spi.xfer2([cmd, 0])
        reply_bitstring = ''.join(bitstring(n) for n in reply_bytes)
        reply = reply_bitstring[5:15]
        return int(reply, 2) / 2**10

if __name__ == "__main__":
    print info
    print "\nReads both channels every 5 seconds"

    adc = MCP3002()

    while True:
```

David Kopala, created the program for the hardware to extrapolate data.



```sql
CREATE TABLE devices (
    ID SERIAL PRIMARY KEY,
    Location VARCHAR(50) NOT NULL,
    Status VARCHAR(20) NOT NULL,
    Temp DOUBLE PRECISION NOT NULL,
    Decibel DOUBLE PRECISION NOT NULL
);

CREATE TABLE locations(
    ID SERIAL PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Latitude Decimal(9,6),
    Longitude Decimal(9,6)
);

CREATE TABLE data (
    ID SERIAL PRIMARY KEY,
    LocationID INT REFERENCES locations(ID) NOT NULL,
    DeviceID INT REFERENCES devices(ID) NOT NULL,
    Timestamp TIMESTAMP DEFAULT NOW(),
    Type Varchar(20) NOT NULL,
    Value Decimal(10, 5) NOT NULL,
    Units Varchar (10)
);

INSERT INTO Devices(Location, Status, Temp, Decibel) VALUES ('Test Location A', 'TEST A', 70, 20);
INSERT INTO Devices(Location, Status, Temp, Decibel) VALUES ('Test Location B', 'TEST B', 71, 21);
INSERT INTO Devices(Location, Status, Temp, Decibel) VALUES ('Test Location C', 'TEST C', 72, 22);
```

Jon Oulton and Alex Ho, helped create and test the SQL for the application.

- Deployment: http://beats.davidkopala.com/