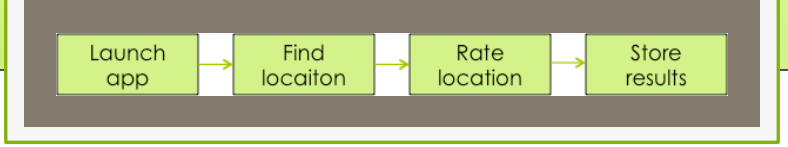
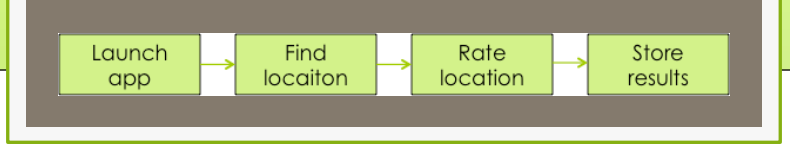
iAmbiance

Barry Tormey
Jason Maynard
Shawn Hathaway

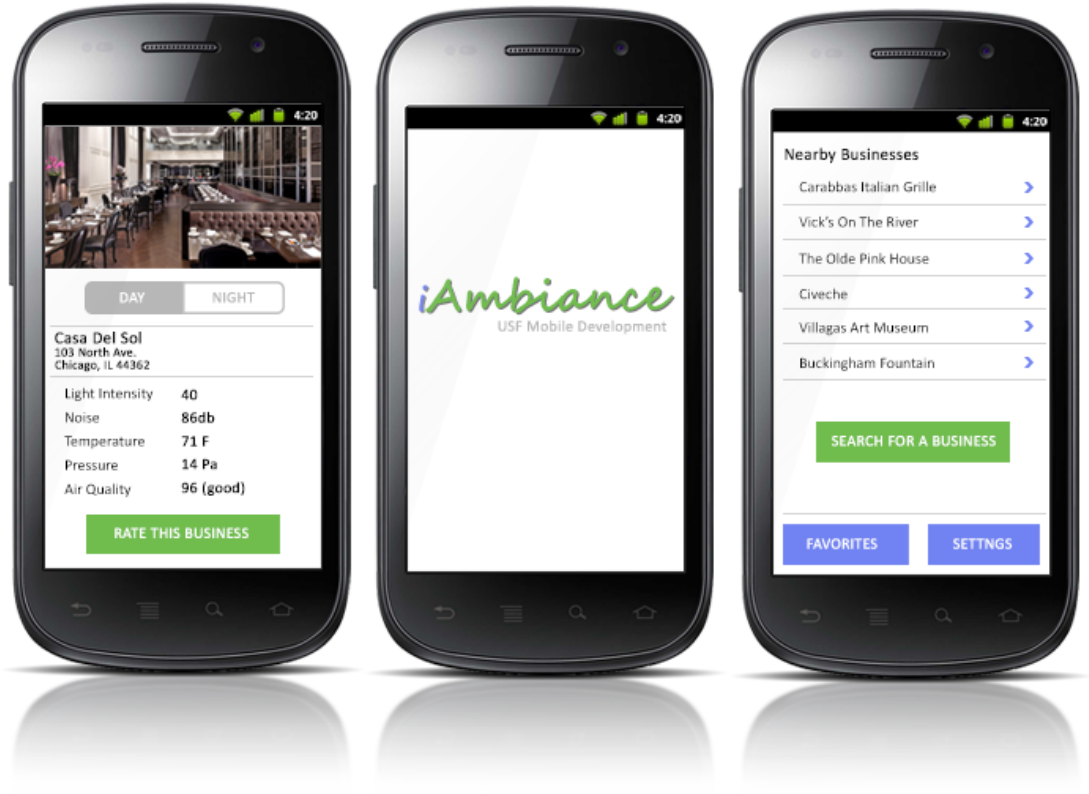


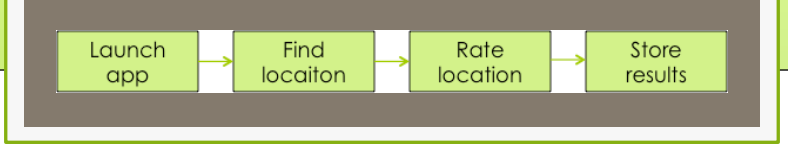
Overview

- System design / GUI - Barry
- Sensor integration - Jason
- Database integration - Shawn
- Demo - Team

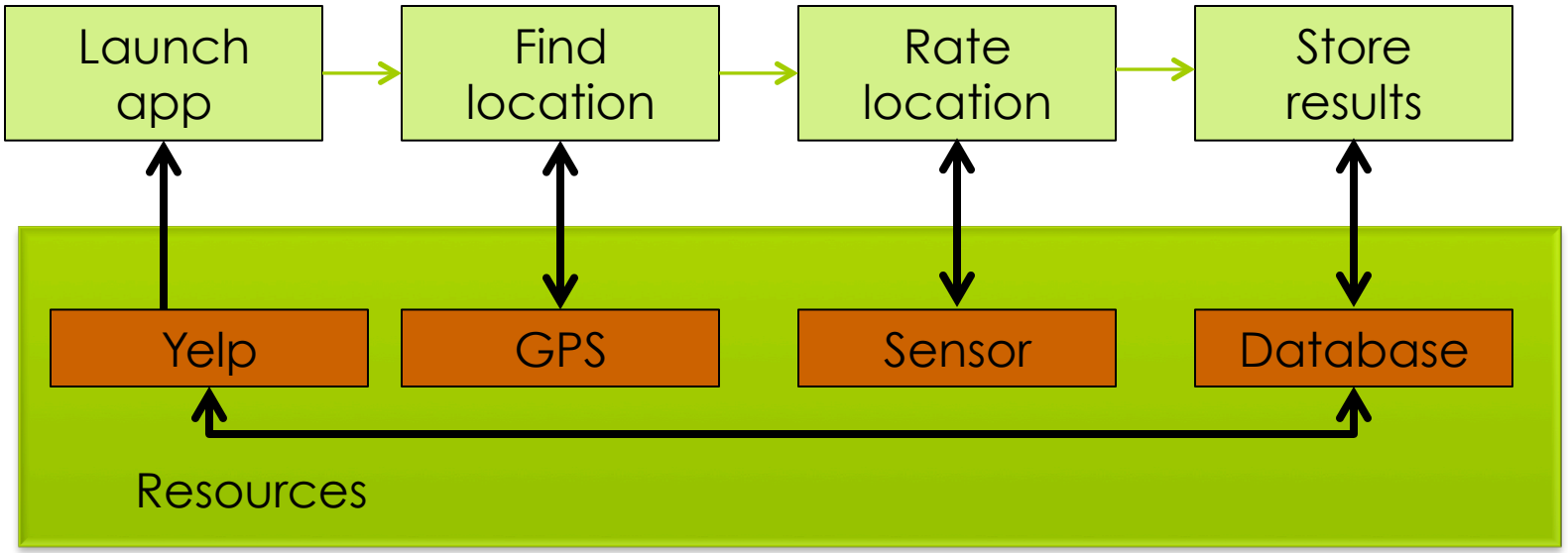


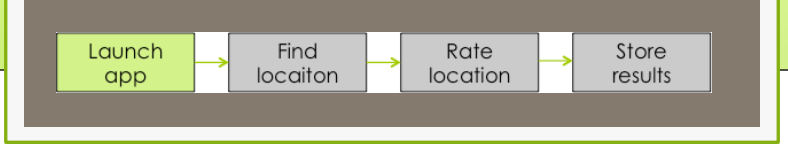
System Design / GUI





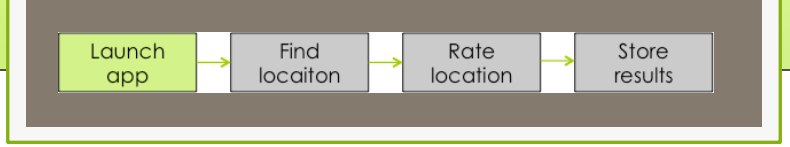
System design





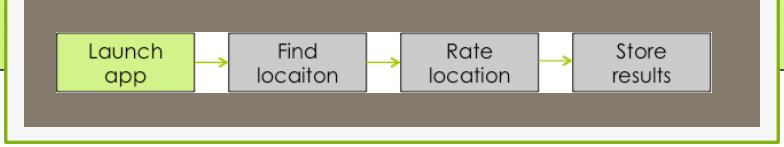
iAmbiance

- Problem
  - Many times before going to a restaurant, visitors want to know information such as how dark the restaurant is or how cold it can be, but generally, you can only gain relative information from friends or online reviews
- Solution
  - iAmbiance along with SensorDrone allow us to provide the users with measureable data as to how loud, bright, or cold the restaurant is on average



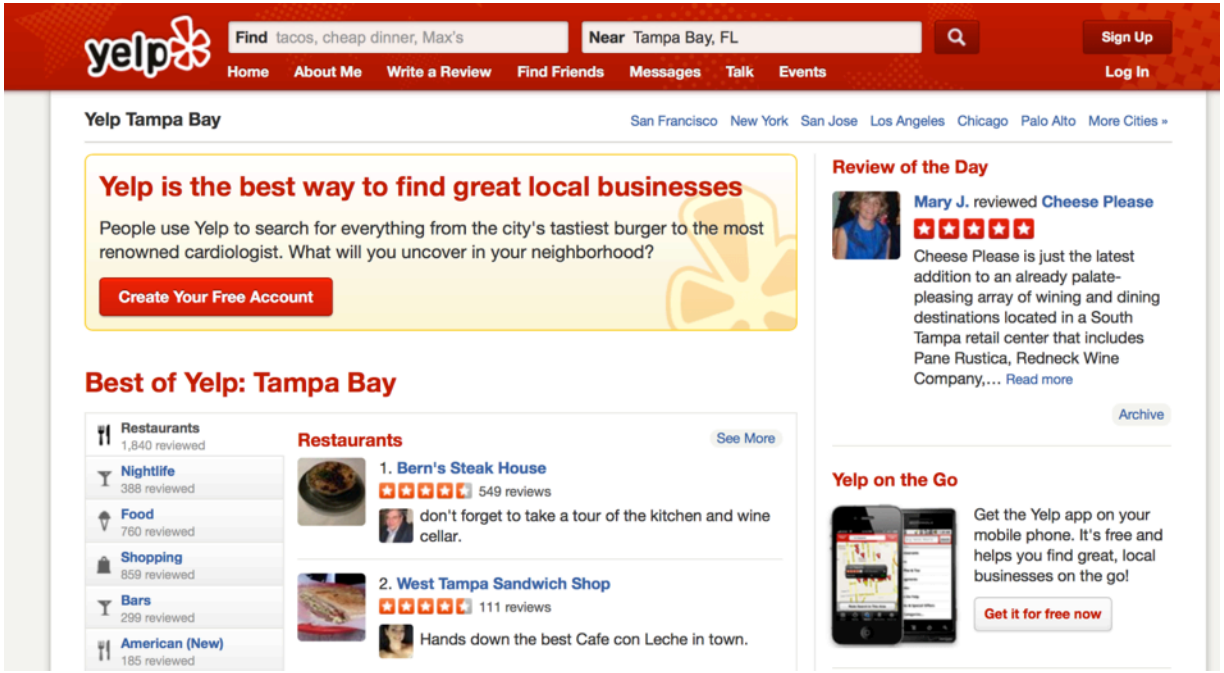
Key features

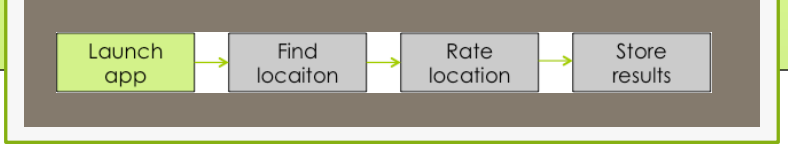
- GUI
- External sensor
- GPS
- Yelp
- Database



How It Works

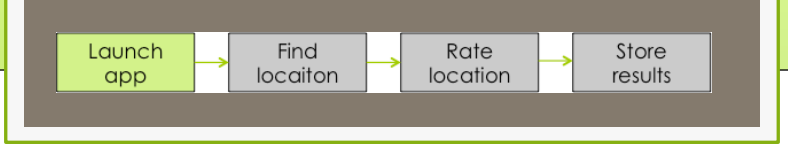
- SensorDrone
  - External sensor which reads data such as:
    1. Temperature
    2. Pressure
    3. Humidity
    4. Light Intensity
    5. Carbon monoxide (Air quality)
- Yelp
  - Use Yelp API to get a list of businesses around the user's location





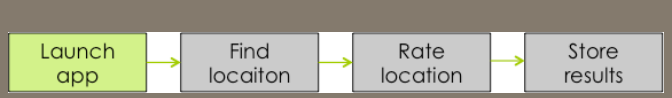
Yelp JSON Object


```
{
  "businesses": [
    {
      "categories": [
        [
          "Local Flavor",
          "localflavor"
        ],
        [
          "Mass Media",
          "massmedia"
        ]
      ],
      "display_phone": "+1-415-908-3801",
      "id": "yelp-san-francisco",
      "is_claimed": true,
      "is_closed": false,
      "image_url": "http://s3-media2.ak.yelpcdn.com/bphoto/7DIHu8a0AHhw-BffrDIxPA/ms.jpg",
      "location": {
        "address": [
          "140 New Montgomery St"
        ],
        "city": "San Francisco",
        "country_code": "US",
        "cross_streets": "3rd St & Opera Aly",
        "display_address": [
          "140 New Montgomery St",
          "(b/t Natoma St & Minna St)",
          "SOMA",
          "San Francisco, CA 94105"
        ],
        "neighborhoods": [
          "SOMA"
```



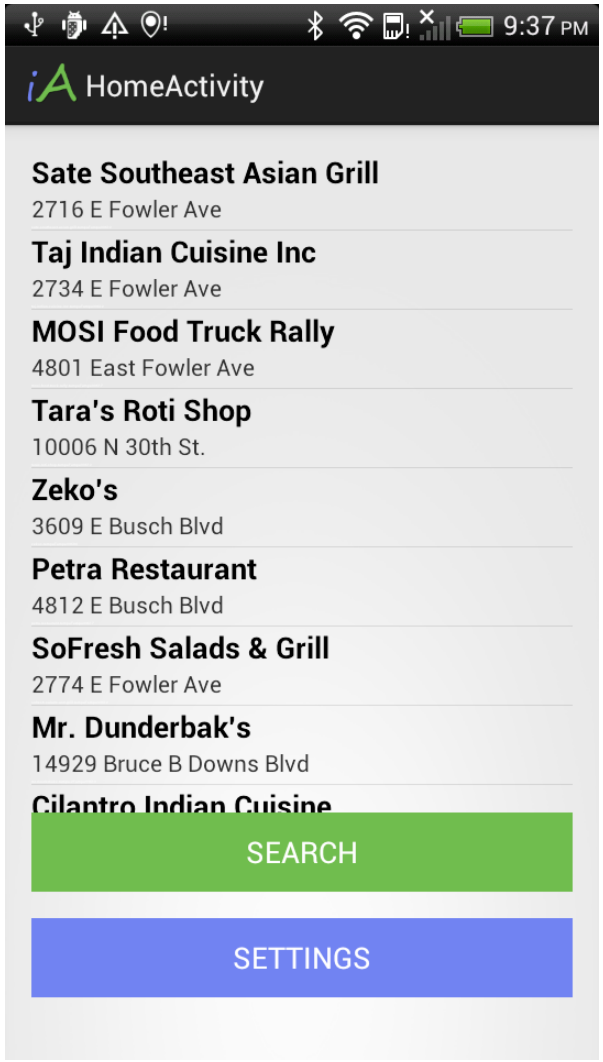
How It Works (cont'd)

- GPS
  - Uses Android's location services to gather the user's current location before any search is performed
- mySQL database
  - Stores rating data
  - Stores user account information



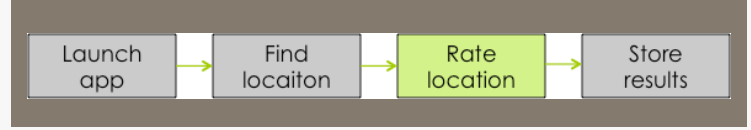
Splash Screen

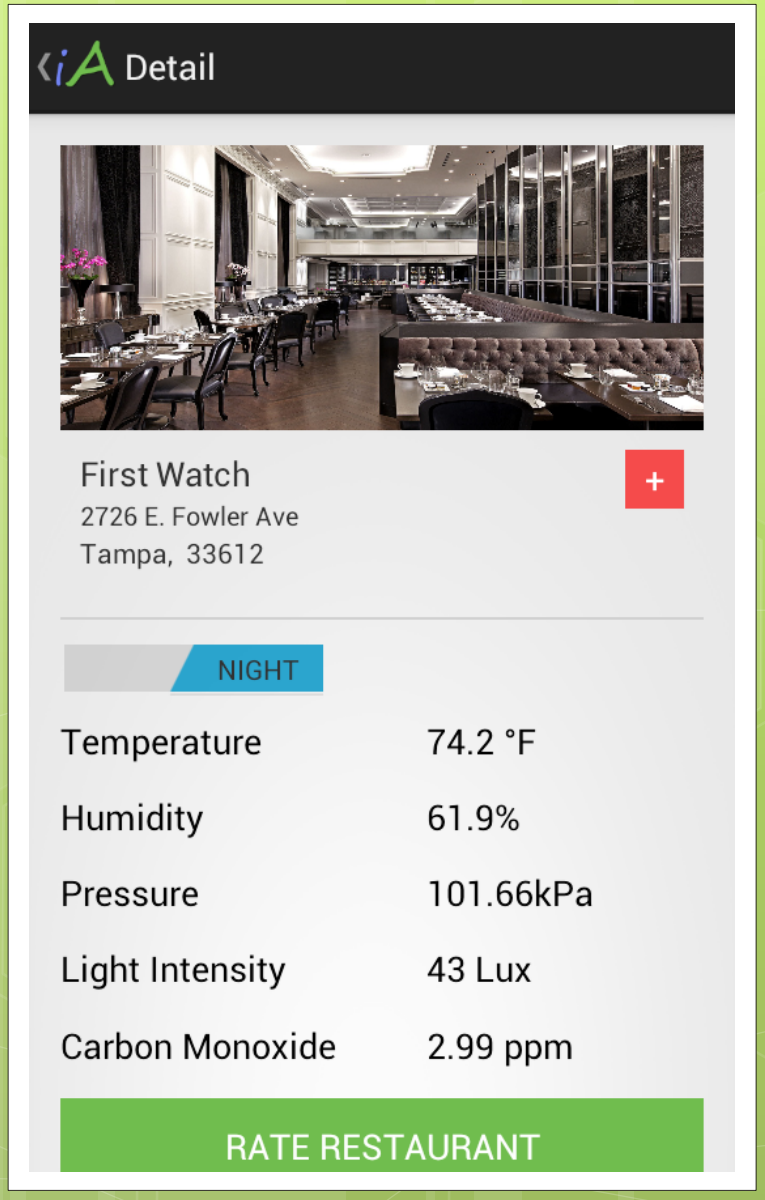
Used just to start any services (GPS locator)



Home Screen

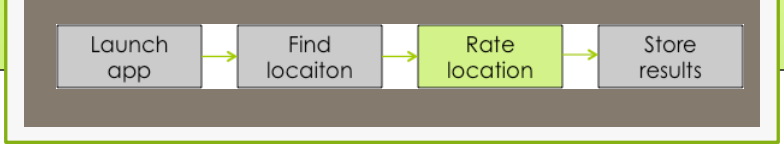
Makes a call to Yelp API with GPS coordinates to return a list of nearby restaurants. Clicking one of these will bring the user to a details page.





Restaurant Details

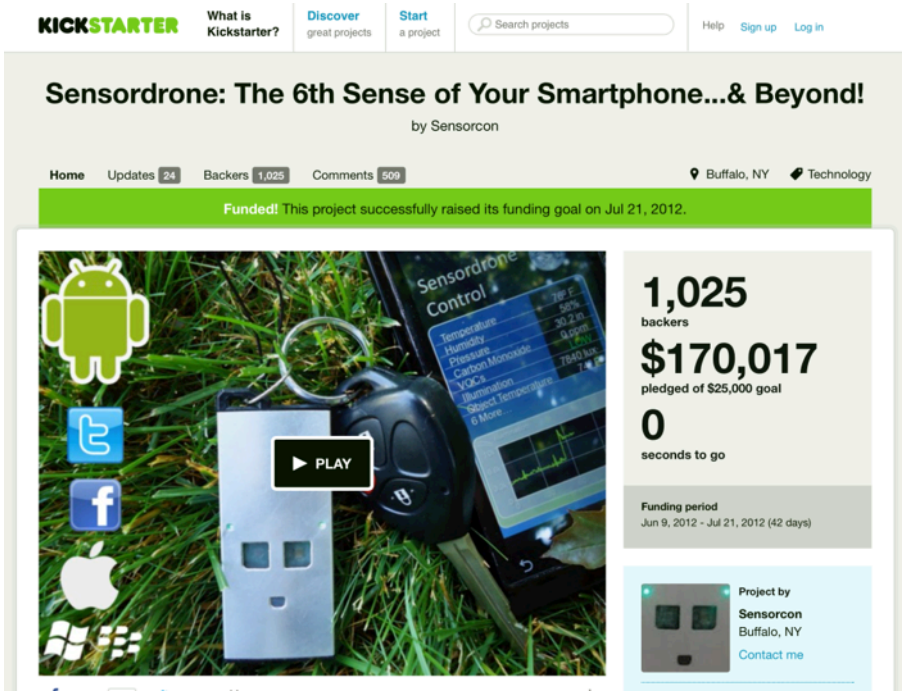
If the restaurant has been rated (in our external database), the rating will show in the Restaurant Details. The user will have the ability to switch between day and night ratings.



Sensorcon

- Startup company from Buffalo NY
- Huge success on Kickstarter!
- Sensorcon asked for $25K on Kickstarter and received $170k
- Dr. Labrador and our team are early adopters and have move the technologhy forward

http://www.kickstarter.com/projects/453951341/sensordrone-the-6th-sense-of-your-smartphoneand-be



Sensordrone Sensor

- Sensor working on Android and iOS
- Updated firmware
- Implemented Sensordrone API
- Established relationship with Sensorcon

State of the Art - Nothing Else Comes Close

Who Are We?

We are Sensorcon. We develop and make Sensors. Sensors are our passion. Sensors are our life. We are not sensor hobbyists, we are sensor fanatics!

SENSORCON

Sensorcon

Re: Sensordrone API questions

Mark Rudolph

Sent: Thursday, October 31, 2013 at 1:54 PM

To: Jason L Maynard

You replied to this message on 10/31/13, 3:25 PM. Show Reply

You replied to this message on 10/31/13, 6:08 PM. Show Reply

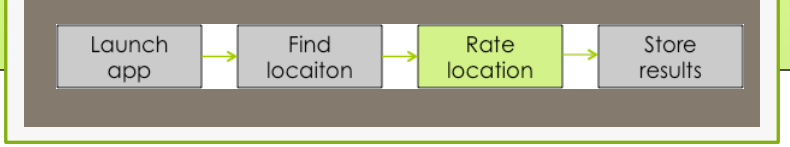
Hi Jason,

I tossed up an example app on our github page: 'Hello Sensordrone"

https://github.com/Sensorcon/Hello-Sensordrone

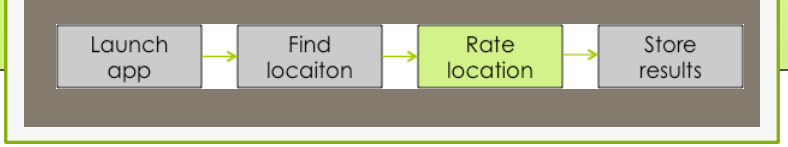
I tried to put as many comments in the code as I could; it also has a copy of the latest snapshot of the library as well. Let me know if anything is unclear, or if you have any other questions.

Best regards,
Mark Rudolph
Sensorcon, Inc.

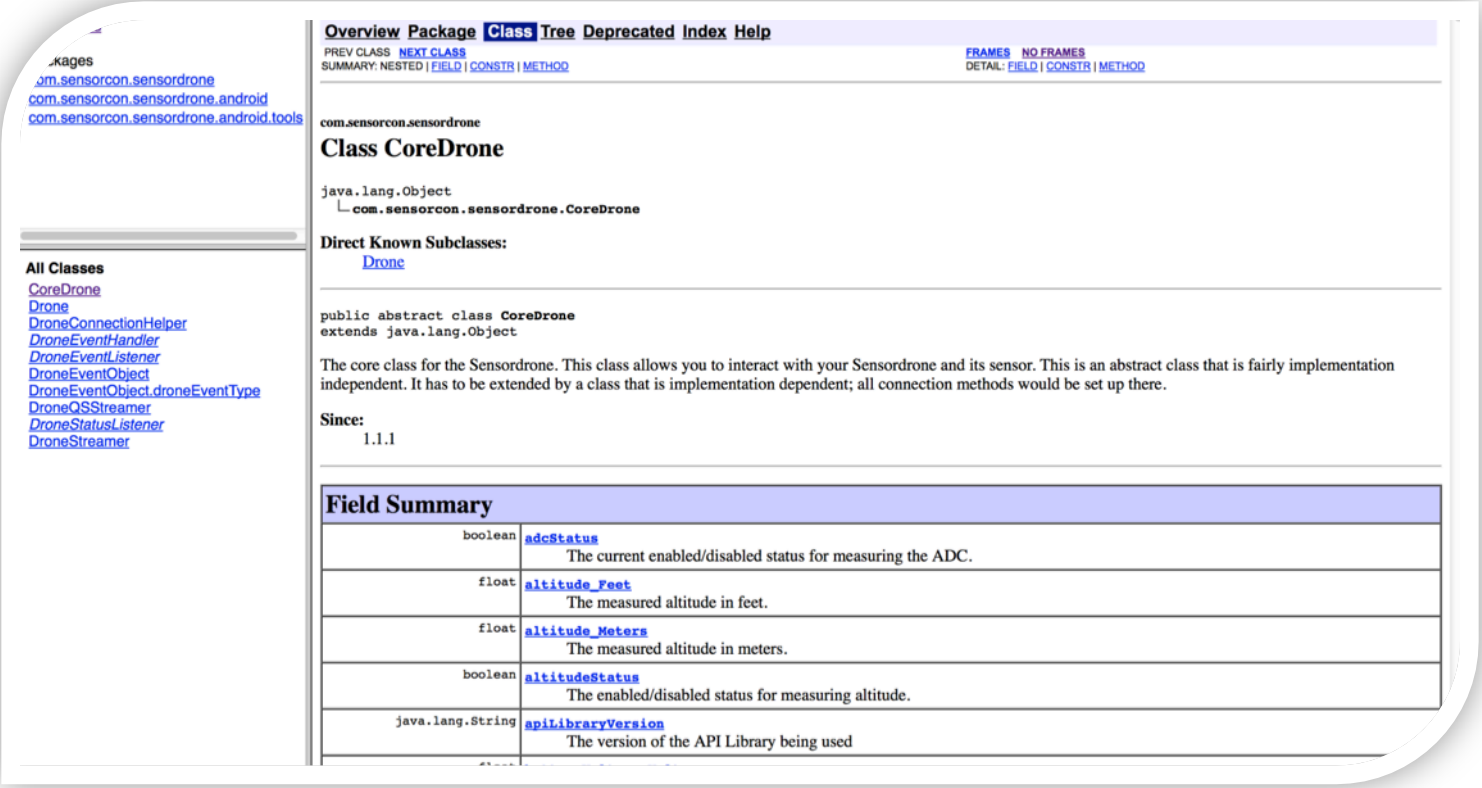


Diverse sensor package

Sensor Type	Technology	Existing And Possible Apps
Precision Gas Sensor	Electrochemical/Fuel Cell Sensor (Calibrated with CO)	Air Quality Breath Analysis Carbon Monoxide (CO) Monitoring Alcohol Testing
Reducing Gas Sensor	Heated Metal Oxide Semiconductor (MOS) Gas Sensor	Methane, Propane, Natural Gas leak detection
Oxidizing Gas Sensor	Heated Metal Oxide Semiconductor (MOS) Gas Sensor	Ozone sensing, Chlorine leaks
Non-Contact Thermometer	Infrared Thermopile Sensor	Non-contact thermometry, thermal leak detection, energy audits, engine diagnostics
Humidity Sensor	Capacitive polymeric Sensor	Weather, Incubators, Refrigerator Crispers, Heat Index, Comfort Guide, Storage
Temperature Sensor	Silicon Bandgap Sensor	A variety of ambient temperature monitoring applications
Light Sensor	Photodiode	Light Intensity, Solar monitoring, Indoor Lighting, Refrigerator Invader monitor!
Color Sensors	Filtered photodiodes for Red, Green & Blue	Automation projects, pattern recognition, color meter, color matching/color analyzer
Pressure	MEMS Pressure Sensor	barometer, altimeter, migrain warning device, weather, chamber pressure
Proximity	Capacitive Electrodes	touch sensing, non-contact sensing, automation, material capacitance analysis, water content
Expansion Connector	Digital (TTL UART & i2C) & Analog (0-3V) Interface: For Connecting other Hardware	Easy Interface for sensors like CO2, water pH, Dissolved Oxygen, EKG, Pulse Rate, OR OTHER HARDWARE Thermal Printers, RC Vehicles & More...



Sensordrone API



.kages
.om.sensorcon.sensordrone
com.sensorcon.sensordrone.android
com.sensorcon.sensordrone.android.tools

All Classes
CoreDrone
Drone
DroneConnectionHelper
DroneEventHandler
DroneEventListener
DroneEventObject
DroneEventObject.droneEventType
DroneQSStreamer
DroneStatusListener
DroneStreamer

Overview Package Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD

FRAMES NO FRAMES
DETAIL: FIELD | CONSTR | METHOD

com.sensorcon.sensordrone

Class CoreDrone

```
java.lang.Object
  └ com.sensorcon.sensordrone.CoreDrone
```

Direct Known Subclasses:
Drone

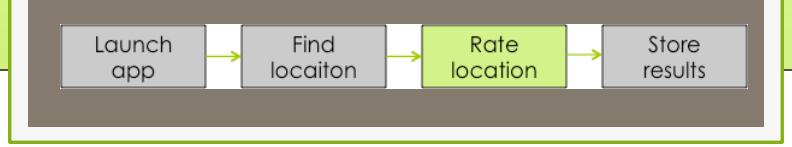
```
public abstract class CoreDrone
extends java.lang.Object
```

The core class for the Sensordrone. This class allows you to interact with your Sensordrone and its sensor. This is an abstract class that is fairly implementation independent. It has to be extended by a class that is implementation dependent; all connection methods would be set up there.

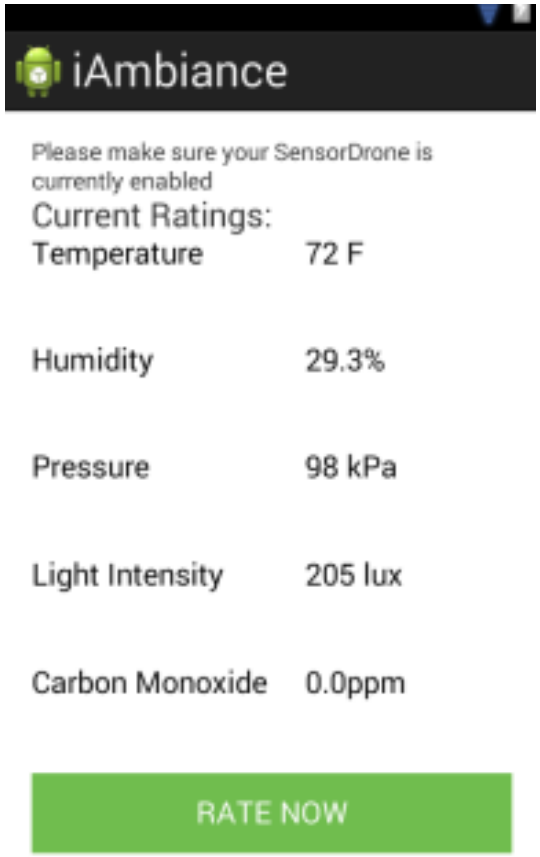
Since:
1.1.1

Field Summary

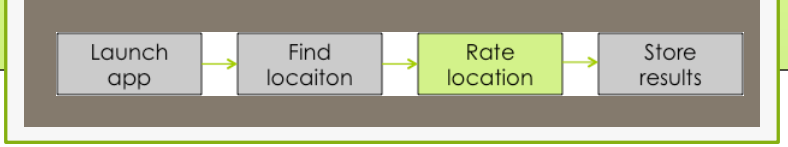
boolean	**adcStatus** The current enabled/disabled status for measuring the ADC.
float	**altitude_Feet** The measured altitude in feet.
float	**altitude_Meters** The measured altitude in meters.
boolean	**altitudeStatus** The enabled/disabled status for measuring altitude.
java.lang.String	**apiLibraryVersion** The version of the API Library being used



Sensor implementation




```
31      // Sensordrone Objects
32      Drone myDrone;
33      DroneEventHandler myDroneEventHandler;
34      DroneConnectionHelper myHelper;
35
36          @Override
37          protected void onCreate(Bundle savedInstanceState) {
38                  super.onCreate(savedInstanceState);
39                  setContentView(R.layout.activity_rate_now);
40
41                  /******** ADDED *********/
42                  // Set up our Sensordrone object
43          myDrone = new Drone();
44
45          // Set up our DroneConnectionHelper
46          myHelper = new DroneConnectionHelper();
47
48          // Target text view for temperature
49          tvTemperature = (TextView)findViewById(R.id.tv_temperature);
50          tvStatus = (TextView)findViewById(R.id.sensor_status);
51          tvHumidity = (TextView)findViewById(R.id.tv_humidity);
52          tvPressure = (TextView)findViewById(R.id.tv_pressure);
53          tvCarbon = (TextView)findViewById(R.id.tv_carbon);
54
55          // Connect
56          if (myDrone.isConnected) {
57              // Don't try to connect again if we are already connected!
58              genericDialog("Whoa!","You are already connected to a Sensordrone.");
59          }
60          else {
61              // Show a list of paired drones that can be selected to connect to.
62              // If there are none, then a message will be displayed about how
63              // to pair one.
64              // You can check out the source code (mentioned above) to see how we do it,
65              // if you need/want to implement your own style.
66              myHelper.connectFromPairedDevices(myDrone, RateNowActivity.this);
67          }
```

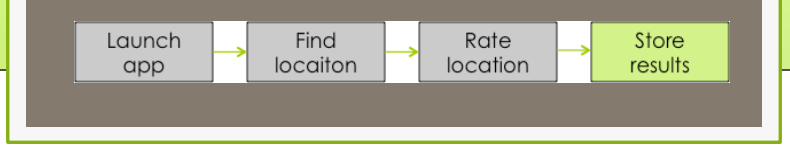


Sensor Implementation

Once an event is triggered…

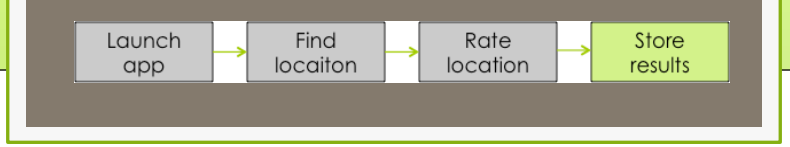
```
else if (droneEventObject.matches(DroneEventObject.droneEventType.TEMPERATURE_ENABLED)) {
    myDrone.measureTemperature();
}
else if (droneEventObject.matches(DroneEventObject.droneEventType.TEMPERATURE_MEASURED)) {
    temp = String.format("%.1f",myDrone.temperature_Fahrenheit) + " \u00B0F";
    updateTextViewFromUI(tvTemperature, temp);
}
```

An event handler parses the event and calls the matching event. The sensor is first enabled, then the reading is measured.



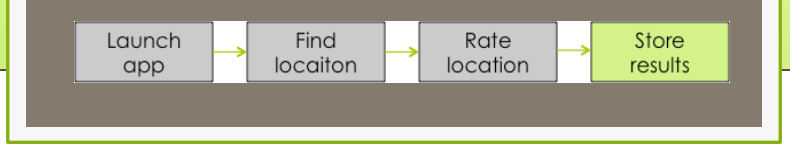
Database

- Users Table
  - ID
  - Email
  - Password (hashed/salted)
- Ratings Table
  - Unique Business ID (FK -> Yelp ID)
  - Raw cumulated ratings and rating counts for each metric
- Audit Table
  - User ID (FK -> Users.ID)
  - Business ID (FK -> Yelp ID)
  - Ratings for each metric
  - Timestamp



Web Service

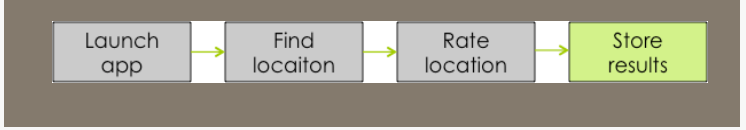
- Written in PHP
- Intermediary between the database and application
- Supported Procedures
  - Submit a Rating (submitRating.php)
    - Email
    - Business ID
    - Metrics
  - Get a Rating (getRating.php)
    - Business ID
  - Create a user (registerUser.php)
    - Email
    - Password
  - Authenticate a user (authenticateUser.php)
    - Email
    - Password

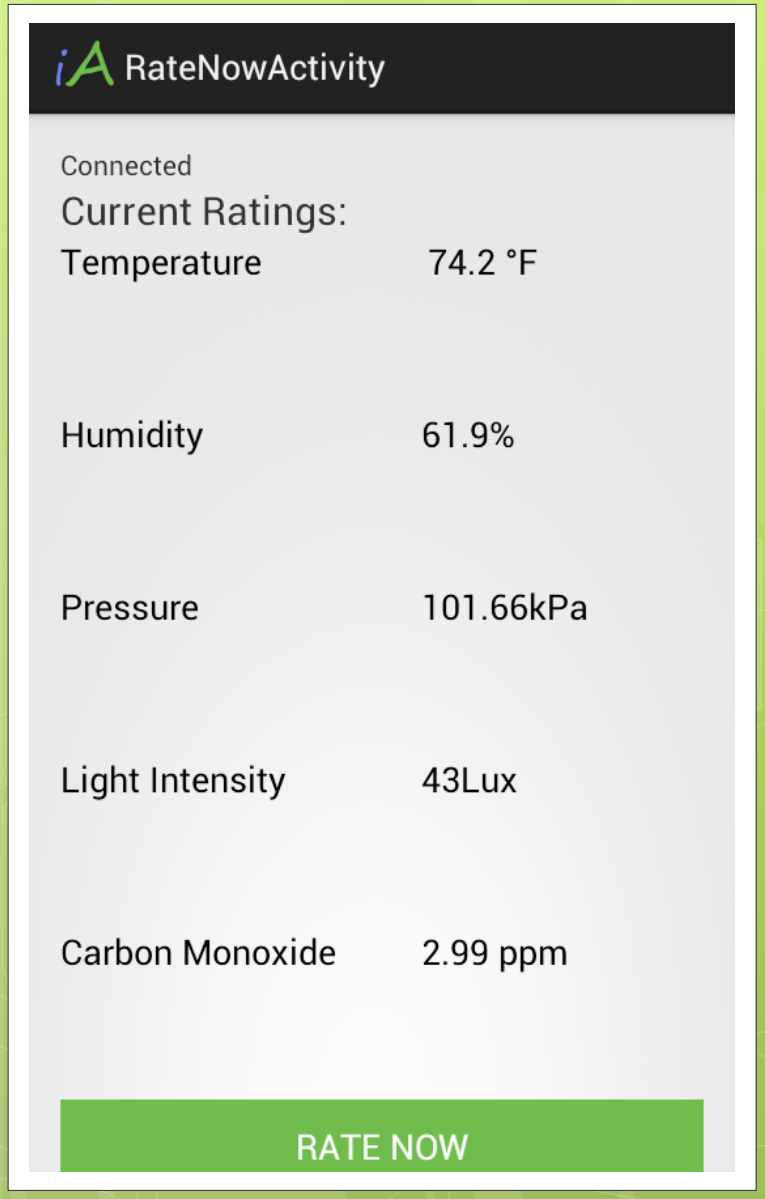


User Security

- Main priority throughout development
- Procedure
  - Unique salt is appended to password
  - The new appended string goes through 1000 iterations of SHA-256 hashing

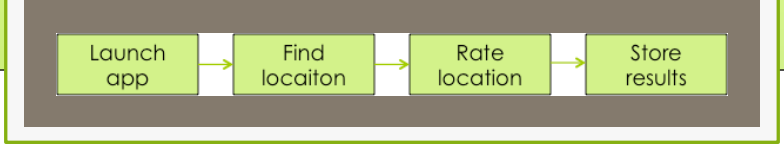
id	email	password
18	a@a.a	sha256:1000:jCC3+AWScozkcNVpniWhxHoFCPWH8lJu:VsEtkYf7oWtG6eT4P4wCm1ULOF2+K0m2
31	notTheNSA@gmail	sha256:1000:U5IjDEcavN541L1WaFFO66e6Fsxstat0:+QHUDwiTtTv1u4aBfabZSUeDtbu4ST5B
29	girlnextdoor@mail.usf.edu	sha256:1000:DH07Hj479X44+BbbtVkhOg6rBbdp0vFx:FaINuN9pleHBjIm0Y7hNnEqZ6JrbISyI
30	freecandy@gmail	sha256:1000:o0jdyHt6wJUW/njBt7lxyS/2BO4CzbNO:9f38/mTUXyjMwHRBpaxiSdD1M/1L+oiJ
19	shawn@gmail.com	sha256:1000:YT8U+lHJbRf4hplQ5VYt4VNlcP7ZRByp:9RTat5V/dBSqZQArN5aO/4KXeGRUGucY
22	btormey@mail.usf.edu	sha256:1000:1B9oXH+1uc34E49FQdLEta9Wrw/Ye7WV:63Kek80b5+sYqLAAn0E9dVXsAXq/eNlZ
28	samueljackson@mail.usf.edu	sha256:1000:yppguNIQu+EGZJ2jYHhLokgExL7elke6:zQaSGOtlP1OVLXY+BV2dMiHkpZFFoE2w
26	jmaynard@mail.usf.edu	sha256:1000:sqP9qfQjEP+szm+udoRULROXBCFi2/Na:ICweAEXr+ILYVRVkCaqd3oVwz/OMrr7e
27	jasonbrown@mail.usf.edu	sha256:1000:9/PJP9k1G46VWZxYhuy7Ag9nSVF+PE6P:gm8M7hULT6QqEBvfEgQMXUXBQXWrt+4q



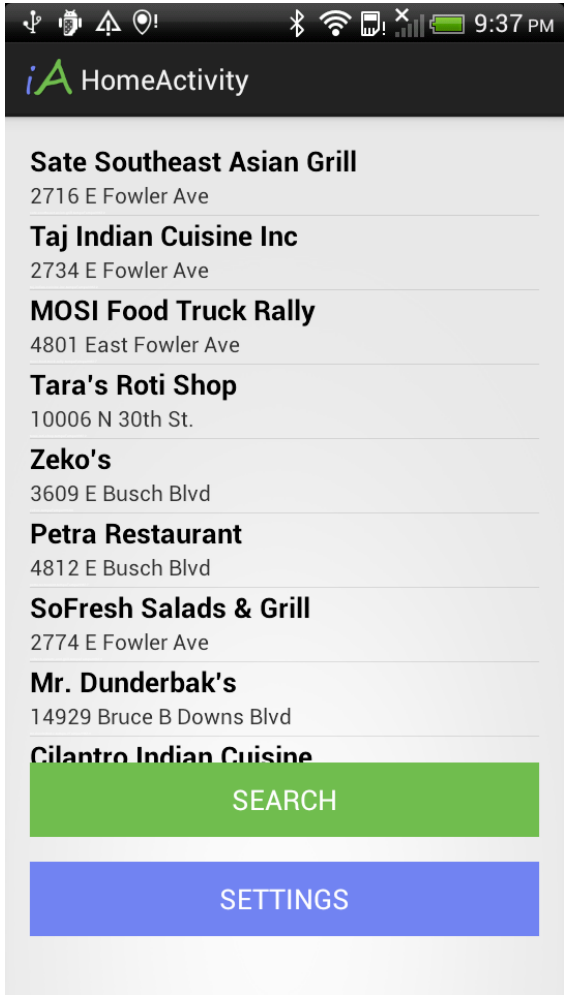


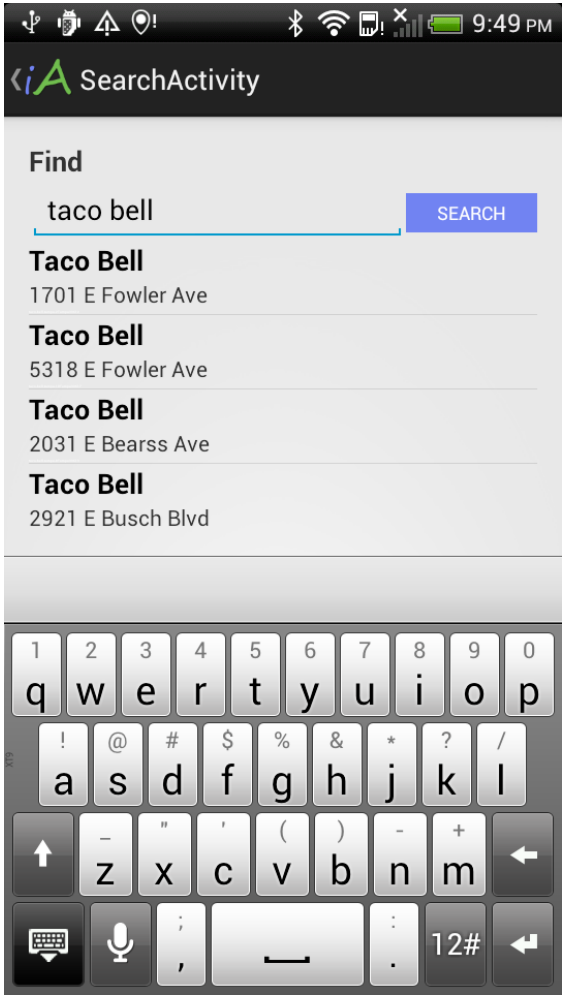
Rate Now

Checks SensorDrone is enabled and gets current ratings. Once rate now is clicked, the current ratings are submitted to external database.


Launch app
Find locaiton
Rate location
Store results

Live demo


9:37 PM
HomeActivity
Sate Southeast Asian Grill
2716 E Fowler Ave
Taj Indian Cuisine Inc
2734 E Fowler Ave
MOSI Food Truck Rally
4801 East Fowler Ave
Tara's Roti Shop
10006 N 30th St.
Zeko's
3609 E Busch Blvd
Petra Restaurant
4812 E Busch Blvd
SoFresh Salads & Grill
2774 E Fowler Ave
Mr. Dunderbak's
14929 Bruce B Downs Blvd
Cilantro Indian Cuisine
SEARCH
SETTINGS


9:49 PM
SearchActivity
Find
taco bell
SEARCH
Taco Bell
1701 E Fowler Ave
Taco Bell
5318 E Fowler Ave
Taco Bell
2031 E Bearss Ave
Taco Bell
2921 E Busch Blvd

Take Away

- We were able to integrate with brand new SensorDrone technology
- We used the phone's GPS sensor to get nearby businesses from Yelp
- Built a robust external database to centrally store ratings for all businesses
- Created a simple, intuitive, user interface

- This is only the beginning - the possibilities are endless.