**KIOSK (Kinetic Informational and Orientation Search KIOSK)\***

**James Raynor, Joseph Lyon, Schulyer Goodman**

KIOSK is a web application went to be deployed on touch screen kiosks throughout a municipality or retail store. These Kiosks would providing information about local government buildings, places of interest, and local businesses or, in the case or retailer, they would provide information about the location and quantity of store inventory. In our specific case we targeted deployment of Kiosks in Provo Downtown Area. This application would help pedestrian traffic in Downtown Provo discover:

* List Government buildings and their respective departments/functions
* Places of interest
* Local participating businesses in downtown area
* Include hours of operation, brief blurbs about each node, and contact information for that entity
* Allow admins to easily add information, modify information, etc. without our intervention
* Future: add in advertising for local business

The application is built on a MongoDB running on a Node server and a Jquery front. Eventually we want to transition most to AngularJS so that we can implement the admin section easier.

**Database Schema**:

We used mongoose to bind schemas to our database and allow easy entry and removal of data from the MongoDB. Mongo is a document based database. So the Schemas we used for Place, and the other objects that inherit from it is:

'use strict';

/\*\*

\* Module dependencies

\*/

var mongoose = require('mongoose'),

Schema = mongoose.Schema;

var util = require('util');

/\*\*

\* Place Schema

\*/

function AbstractPlaceSchema() {

Schema.apply(this, arguments);

this.add({

name: {

type: String,

trim: true,

default: ''

},

logo: {

data: Buffer,

contentType: String

},

address: {

type: String,

trim: true,

default: ''

},

hours: String,

website: {

type: String,

trim: true,

default: ''

},

coordinates:

{

longitude: {

type: Number,

default: -111.658492

},

latitude: {

type: Number,

default: 40.233622

}

}

});

}

util.inherits(AbstractPlaceSchema, Schema);

var PlaceSchema = new AbstractPlaceSchema();

var GovBSchema = new AbstractPlaceSchema({

govBDept: String,

govBServices: {

type: [String],

index: true

}

});

var RestauranteSchema = new AbstractPlaceSchema({

menu: {

data: Buffer,

contentType: String

},

type: String, //TODO change to enumeration

price: Number,

yelpReview: String

});

var StoreSchema = new AbstractPlaceSchema({

type: String, //TODO change to enumeration

inventory: [{

item: String,

quantity: {

type: Number,

default: 0

}

}]

});

var LocationSchema = new AbstractPlaceSchema({

type: String, //TODO change to enumeration

activities: String

});

var Place = mongoose.model('Place', PlaceSchema);

var GovB = mongoose.model('GovB', GovBSchema);

var Restaurant = mongoose.model('Restaurant', RestauranteSchema);

var Store = mongoose.model('Store', StoreSchema);

var Location = mongoose.model('Location', LocationSchema);

Producing JSON like:

"Places": {

"GovB": [

{

"name": "Utah County Planning & Zoning",

"logo": "/imgs/placeholder-logo.png",

"address": "51 S University Ave, Provo, UT 84601, United States",

"hours": "9-5",

"website": "utahcounty.gov",

"govBDept": "Planning & Zoning",

"govBServices": [

"Planning",

"Zoning"

],

"coordinates": {

"longitude": -111.657964,

"latitude": 40.232845

}

}

],

"Restaurant": [

{

"name": "Los Hermanos",

"logo": "/imgs/placeholder-logo.png",

"address": "71 E Center St. Provo, UT, 84601, United States",

"hours": "12-10",

"website": "loshermanosutah.com",

"menu": "loshermanosutah.com/menu.html",

"type": "Mexican",

"price": 2,

"yelpReview": "yelp.com",

"coordinates": {

"longitude": -111.657249,

"latitude": 40.233932

}

}

],

"Store": [

{

"name": "Hookah Collection",

"logo": "/imgs/placeholder-logo.png",

"address": "198 Web Center Street Provo, UT, 84601, United States",

"hours": "9-5",

"website": "none",

"type": "GiftShop",

"inventory": [

{

"item": "hookah1",

"quantity": 2

},

{

"item": "hookah2",

"quantity": 1

}

],

"coordinates": {

"longitude": -111.662065,

"latitude": 40.233960

}

},

{

"name": "Lund Optical",

"logo": "/imgs/placeholder-logo.png",

"address": "20 North University Avenue Provo, UT 84601",

"hours": "10-6",

"website": "none",

"type": "Optical",

"inventory": [],

"coordinates": {

"longitude": -111.658428,

"latitude": 40.234154

}

}

],

"Location": [

{

"name": "Acaydia School of Aesthetics",

"logo": "/imgs/placeholder-logo.png",

"address": "86 N University Ave, Provo, UT 84601, United States",

"hours": "9-5",

"website": "acaydia.com",

"type": "School",

"activities": "Learning, appearance",

"coordinates": {

"longitude": -111.658045,

"latitude": 40.234761

}

}

]

}

UserSchema = new Schema({

firstName: {

type: String,

trim: true,

default: '',

validate: [validateLocalStrategyProperty, 'Please fill in your first name']

},

lastName: {

type: String,

trim: true,

default: '',

validate: [validateLocalStrategyProperty, 'Please fill in your last name']

},

displayName: {

type: String,

trim: true

},

email: {

type: String,

trim: true,

default: '',

validate: [validateLocalStrategyProperty, 'Please fill in your email'],

match: [/.+\@.+\..+/, 'Please fill a valid email address']

},

username: {

type: String,

unique: 'Username already exists',

required: 'Please fill in a username',

trim: true

},

password: {

type: String,

default: '',

validate: [validateLocalStrategyPassword, 'Password should be longer']

},

salt: {

type: String

},

provider: {

type: String,

required: 'Provider is required'

},

providerData: {},

additionalProvidersData: {},

roles: {

type: [{

type: String,

enum: ['user', 'admin']

}],

default: ['user']

},

updated: {

type: Date

},

created: {

type: Date,

default: Date.now

},

/\* For reset password \*/

resetPasswordToken: {

type: String

},

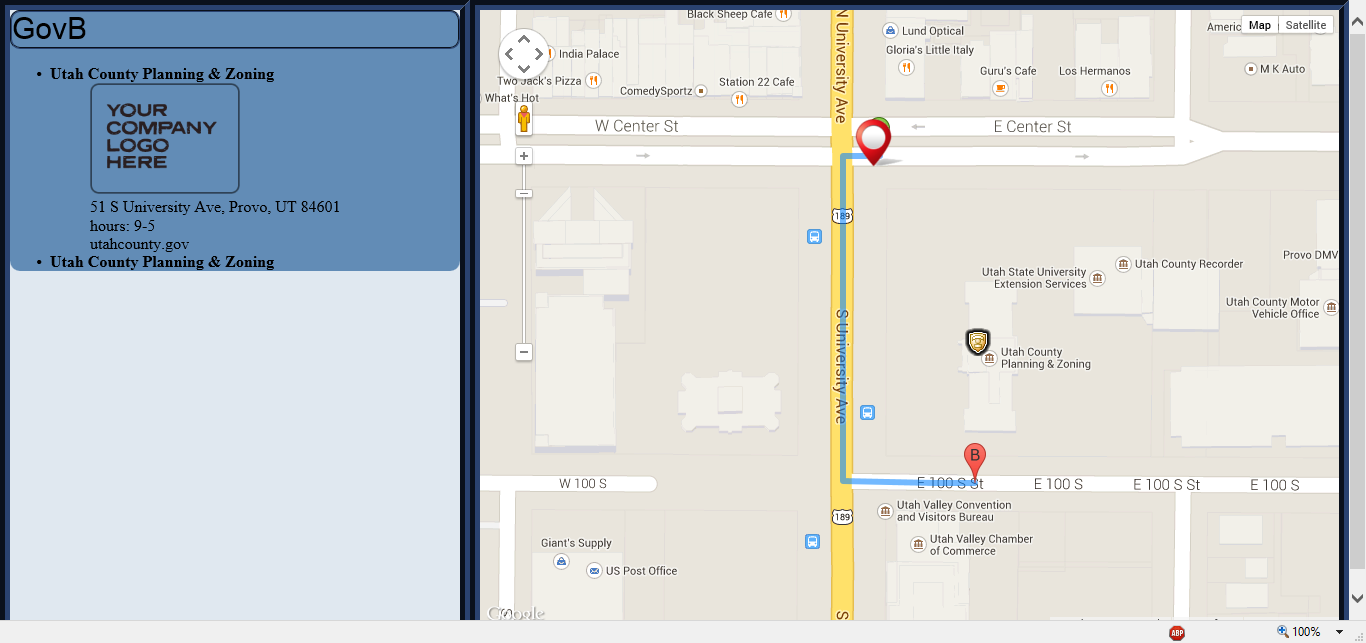
resetPasswordExpires: {

type: Date

}

**How it Works**

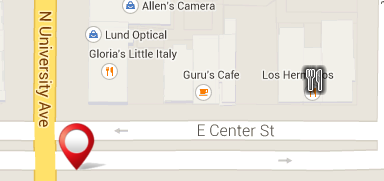
The user is presented with a splash screen prompting them to “Touch the Screen to Begin”. The user is then taken to a screen with a few categories corresponding to the number of Types of Places in our database. The Right hand side of the screen is occupied by a google map. Upon entering this view, different markers for each type of location. When a user clicks on a type of location (restaurant, government building, etc), the other types minimize and a list of specific locations of that type are displayed. Also only markers for that type of location are left on the map drill down for the user.



Clicking on a specific Location gives the route from the kiosk to the location and,



Also on the left menu area expands to show more details about the location



Other markers are available but don’t expand the additional information available in the database

**Future Work**: Describe a roadmap for future development, with additional features you could add or changes to the interface.

We need to spruce up the left hand side interface so that it looks cleaner and simpler. We plan on finishing up the admin editor in the web interface, for now we send REST commands to our API to add in and edit places. We added in authorized users but didn’t finish implementing the front end for it. In the future we would also like to add a search bar for the user and a category search. Also we planned on having iframes for yelp reviews for restaurants or government websites. Parking meter integration is also a future option.

**Hosted URL:**

<http://104.236.124.22:5000/>