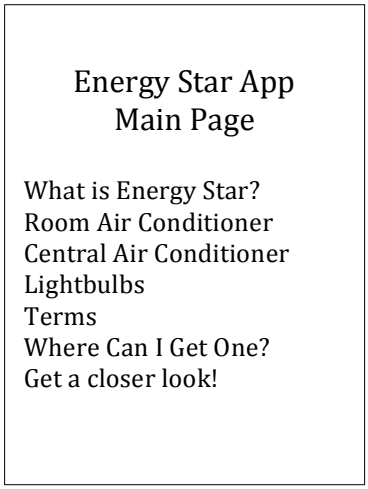
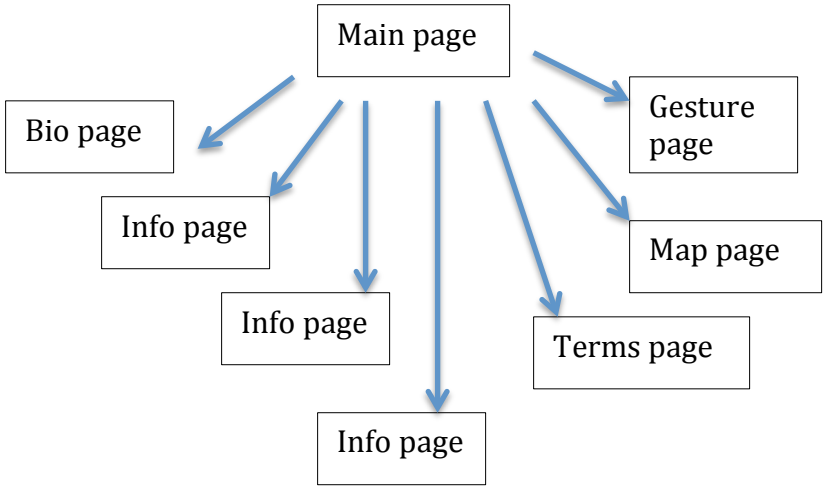
Sarah Egener
Project plan 2

I want to continue my idea from project 1 into project 2. For my first app I incorporated the idea of energy efficiency. Now I want to continue that idea but expand the organization, aesthetics, and functions. So far my app holds a bunch of information but it is squished on the single-page design. Ideally I would figure out the multi-page interface because then I can add way more features without running out of space. I noticed from one of the MAD folders on Edmodo that this week we are starting multi-view controllers in class and this is exactly the type of information I am looking for. In the textbook there is a whole chapter dedicated to mutliview applications (chapter 6), so I have a strong foundation there to explore. I have previewed the slides that start this week, referring to this aspect. They say that view controllers act as the glue for connecting the view and model. Because a single controller connects with a single view, I'm going to need multiple view controllers. As of my project plan it seems like I am going to need 7 view controllers. The tab bar controller, used to present different types of information, could be a useful design for my app. My main page displaying all my other views, whether it be an info view, gesture view, map view, etc. is the root controller. From the slides, notes, and textbook I think I will be able to tackle to multi-view app. I don't want to include code from those resources in my project plan because I learn much better by being taught in class before I approach something new.

I'll have a main page (root controller view) that links to the information I had for project 1, as well as more calculations I put together about other appliances. The buttons that showed definitions when tapped will have their own page. It's important to include these because more often than not people don't have a good grasp of the way we describe energy. I don't want to have a functional app but users can't even understand the jargon used. The Mapkit will have its own page too. I want the map to display annotations depending on the user's current location. So at Atlas, the lab annotations in Boulder will pop up. But if the user is in Denver, different annotations will show of places to buy energy star appliances in Denver. If I can really get this multi-page aspect to work I can expand my app even more with lots of information.

This is my main page when the user enters the app initially. So far I have crunched the numbers for the room AC (that was project 1) but as of now I have for central AC systems and lightbulbs. This is a hierarchical depiction of what I mean.





On the information views, I'll still have images and segmented controls that are connected. When the conventional side to the control is selected, a descriptor image pops up. Then, when each of the segmented controls switch to energy star, the energy star label pops up next to the initial image.

Along with the image, information will continue to pop up in labels explaining the numerical data regarding whichever segment the user chooses. For the air conditioners, here is the average information I have gathered with Denver's electricity rate of $0.113/kwh:
This was the information I included in project 1:

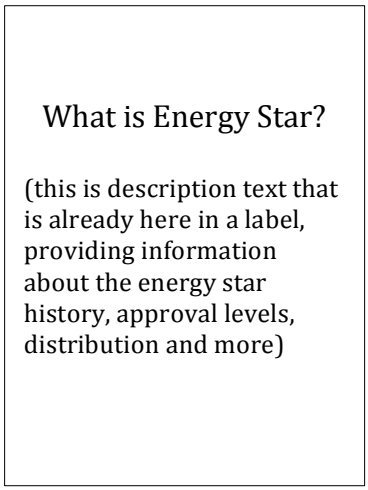
Room AC	Conventional	Energy Star
Upfront cost	$170	$220
Energy efficiency ratio (EER)	9.8	10.8
Energy cost/*lifetime*	$537	$488
Lifecycle energy saved	n/a	534 kWh
Air pollution reduction	n/a	822 lb/CO2

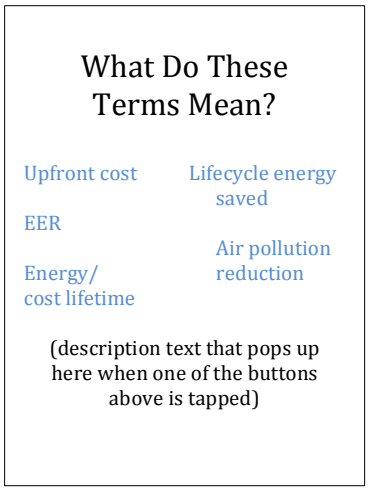
This is the information I'm adding to project 2:

Central AC	Conventional	Energy Star
Upfront cost	$170	$220
Energy efficiency ratio (EER)	9.8	10.8
Energy cost/*lifetime*	$537	$488
Lifecycle energy saved	n/a	534 kWh
Air pollution reduction	n/a	822 lb/CO2

Lightbulbs	Conventional	Energy Star
Upfront cost	$170	$220
Energy efficiency ratio (EER)	9.8	10.8
Energy cost/*lifetime*	$537	$488
Lifecycle energy saved	n/a	534 kWh
Air pollution reduction	n/a	822 lb/CO2

So the images, labels, and segmented controls are for the informational pages and I have lots of experience in those. The bio page is to explain to the user what this app is all about and a little history of the energy star approval label. The button page has the same idea of the buttons I used for project 1. For example when you tap "upfront cost," text pops up in a label with a description of that term. I want a view dedicated to those terms, now. These views are pretty basic, looking like this:





My map view comes from work I've done in project 1 and lab 5. I have all of the code needed to find current location with a marker that tracks user location with a blue line. I also know how to make annotation pins to alert the users where in the Boulder/Denver area they can then find energy star appliances. I want this map to take up the entire view, as oppose to project 1 where I had to fit it in a tiny section of the single-view. In project 1 I also included a longitude, latitude, time, and speed label from the Mapkit data and I want to include this as well. I want to include annotations from Boulder, Westminister, and Denver and about 3 for each city. For lab 5 I already found the annotations for Boulder. I want labels to pop up providing the information for each city's annotations including phone number and address. Using Google I can find the addresses and then using stevemorse.org/jcal/latlon.php I find the coordinates.

Boulder:

Best Buy: 1740 30th st. 303-938-2889…lat: 40.0167 lon: -105.2532
Home Depot: 1600 29th st. 303-449-4221…lat: 40.0155 lon: -105.25602
Contract Appliance: 1661 28th st. 720-245-2323…lat: 40.0159 lon: -105.25875

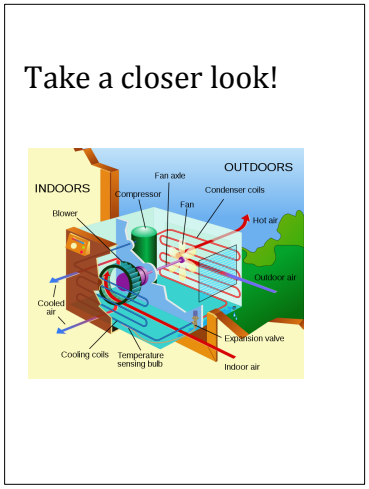
Westminister:

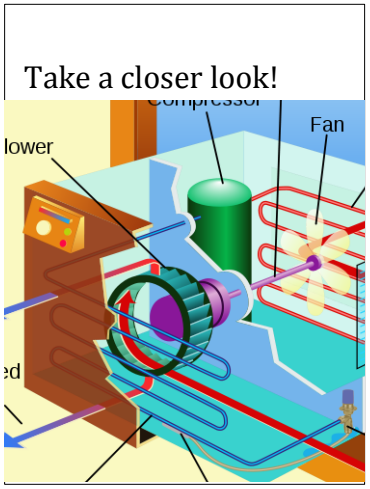
Appliance Factory: 5880 W 88th Ave. 303-657-0199
Best Choice Appliance: 7850 N Federal Blvd. 303-650-2225
Best Buy: 9369 N Sheridan Blvd. 303-426-4434

Denver:

Appliance Factory: 2816 Colorado Blvd. 303-388-0823
Contract Appliance: 1045 Zuni st. 303-996-1002
Sears: 701 Osage st. 303-892-8649

Finally, I want to include a gesture page in order to give users a better look at the inside technicalities of efficient appliances. If I can find HD images of this physicality's, I can then use the rotate and scale gestures to include some more user/app interaction. The whole point to the app is proving the technological advancements we have nowadays so I think a view to check this out further would be prudent.





Segmented control code:

```
@IBAction func applianceSwitch(sender: UISegmentedControl) {
        if infoSwitch.selectedSegmentIndex == 0{
            imageSwitch.image=UIImage(named:"conventional")
            efficiencyAnswer.text = "$170"
            upfrontAnswer.text = "9.8"
            costAnswer.text = "$537"
            savingsAnswer.text = "none"
            pollutionAnswer.text = "none"
            }
        else if infoSwitch.selectedSegmentIndex == 1{
            imageSwitch.image=UIImage(named:"star")
            efficiencyAnswer.text = "$220"
            upfrontAnswer.text = "10.8"
            costAnswer.text = "$488"
            savingsAnswer.text = "534 kwh"
            pollutionAnswer.text = "822 lb. CO2"
        }
    }
```

Button/terms code:

```
@IBAction func buttonExplanations(sender: UIButton) {
        if sender.tag == 1{
            explanationLabel.text = "how efficient it is"
        }
        else if sender.tag==2{
            explanationLabel.text = "first cost paid right when appliance received"
```

```
        }
        else if sender.tag==3{
            explanationLabel.text = "accumulated costs over its lifetime"
        }
        else if sender.tag==4{
            explanationLabel.text = "accumulated energy saved due to efficiency"
        }
        else if sender.tag==5{
            explanationLabel.text = "accumulated pollution avoided due to efficiency"
        }
    }
```

Mapkit code:

```
@IBOutlet weak var theMap: MKMapView!
    @IBOutlet weak var theLabel: UILabel!

    var manager: CLLocationManager!
    var myLocations: [CLLocation] = []


    func locationManager(manager:CLLocationManager, didUpdateLocations
locations:[AnyObject]) {
        theLabel.text = "\(locations[0])"
        myLocations.append(locations[0] as! CLLocation)

        let spanX = 0.007
        let spanY = 0.007
        var newRegion = MKCoordinateRegion(center: theMap.userLocation.coordinate, span:
MKCoordinateSpanMake(spanX, spanY))
        theMap.setRegion(newRegion, animated: true)

        if (myLocations.count > 1){
            var sourceIndex = myLocations.count - 1
            var destinationIndex = myLocations.count - 2
            let c1 = myLocations[sourceIndex].coordinate
            let c2 = myLocations[destinationIndex].coordinate
            var a = [c1, c2]
            var polyline = MKPolyline(coordinates: &a, count: a.count)
            theMap.addOverlay(polyline)
        }
    }

    func mapView(mapView: MKMapView!, rendererForOverlay overlay: MKOverlay!) ->
MKOverlayRenderer! {

        if overlay is MKPolyline {
            var polylineRenderer = MKPolylineRenderer(overlay: overlay)
            polylineRenderer.strokeColor = UIColor.blueColor()
            polylineRenderer.lineWidth = 4
            return polylineRenderer
        }
        return nil
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        manager = CLLocationManager()
        manager.delegate = self
        manager.desiredAccuracy = kCLLocationAccuracyBest
        manager.requestAlwaysAuthorization()
        manager.startUpdatingLocation()
        theMap.delegate = self
        theMap.mapType = MKMapType.Satellite
        theMap.showsUserLocation = true
       let bestBuy = CLLocationCoordinate2D(latitude:40.0167, longitude: -105.25326)
        let span2 = MKCoordinateSpanMake(0.05, 0.05)
        let region2 = MKCoordinateRegionMake(bestBuy,span2)
        theMap.setRegion(region2, animated: true)
```

```
        let annotation2 = MKPointAnnotation()
        annotation2.coordinate = bestBuy
        annotation2.title="Best Buy"
        annotation2.subtitle="1740 30th St. Boulder"
        theMap.addAnnotation(annotation2)

        let homeDepot = CLLocationCoordinate2D(latitude:40.01556, longitude: -105.256022)
        let span3 = MKCoordinateSpanMake(0.05, 0.05)
        let region3 = MKCoordinateRegionMake(homeDepot,span3)
        theMap.setRegion(region3, animated: true)
        let annotation3 = MKPointAnnotation()
        annotation3.coordinate = homeDepot
        annotation3.title="Home Depot"
        annotation3.subtitle="1600 29th St. Boulder"
        theMap.addAnnotation(annotation3)

        let contractApp = CLLocationCoordinate2D(latitude:40.01596, longitude: -
105.258756)
        let span4 = MKCoordinateSpanMake(0.05, 0.05)
        let region4 = MKCoordinateRegionMake(contractApp,span3)
        theMap.setRegion(region4, animated: true)
        let annotation4 = MKPointAnnotation()
        annotation4.coordinate = contractApp
        annotation4.title="Contract Appliance"
        annotation4.subtitle="1661 28th St. Boulder"
        theMap.addAnnotation(annotation4)
}
```

Gesture code:

```
   //SCALE
    @IBAction func handlePinch(sender: UIPinchGestureRecognizer) {
        sender.view!.transform = CGAffineTransformScale(sender.view!.transform,
sender.scale, sender.scale)
        sender.scale=1
    }

    //ROTATE
    @IBAction func handleRotate(sender: UIRotationGestureRecognizer) {
        sender.view!.transform = CGAffineTransformRotate(sender.view!.transform,
sender.rotation)
        sender.rotation=0
    }

    //GESTURES AT SAME TIME
    func gestureRecognizer(gestureRecognizer: UIGestureRecognizer,
shouldRecognizeSimultaneouslyWithGestureRecognizer otherGestureRecognizer:
UIGestureRecognizer) -> Bool {
        return true
    }
```