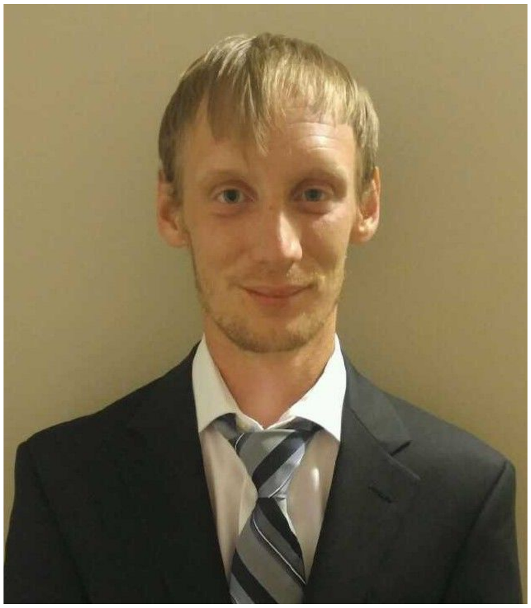
Mike Sherman
Restaurant Technologies
Service Ticket Tracker

Tst.suremandevelopment.com



www.suremandevelopment.com

https://www.linkedin.com/in/mike-sherman

8/12/2016

About Me

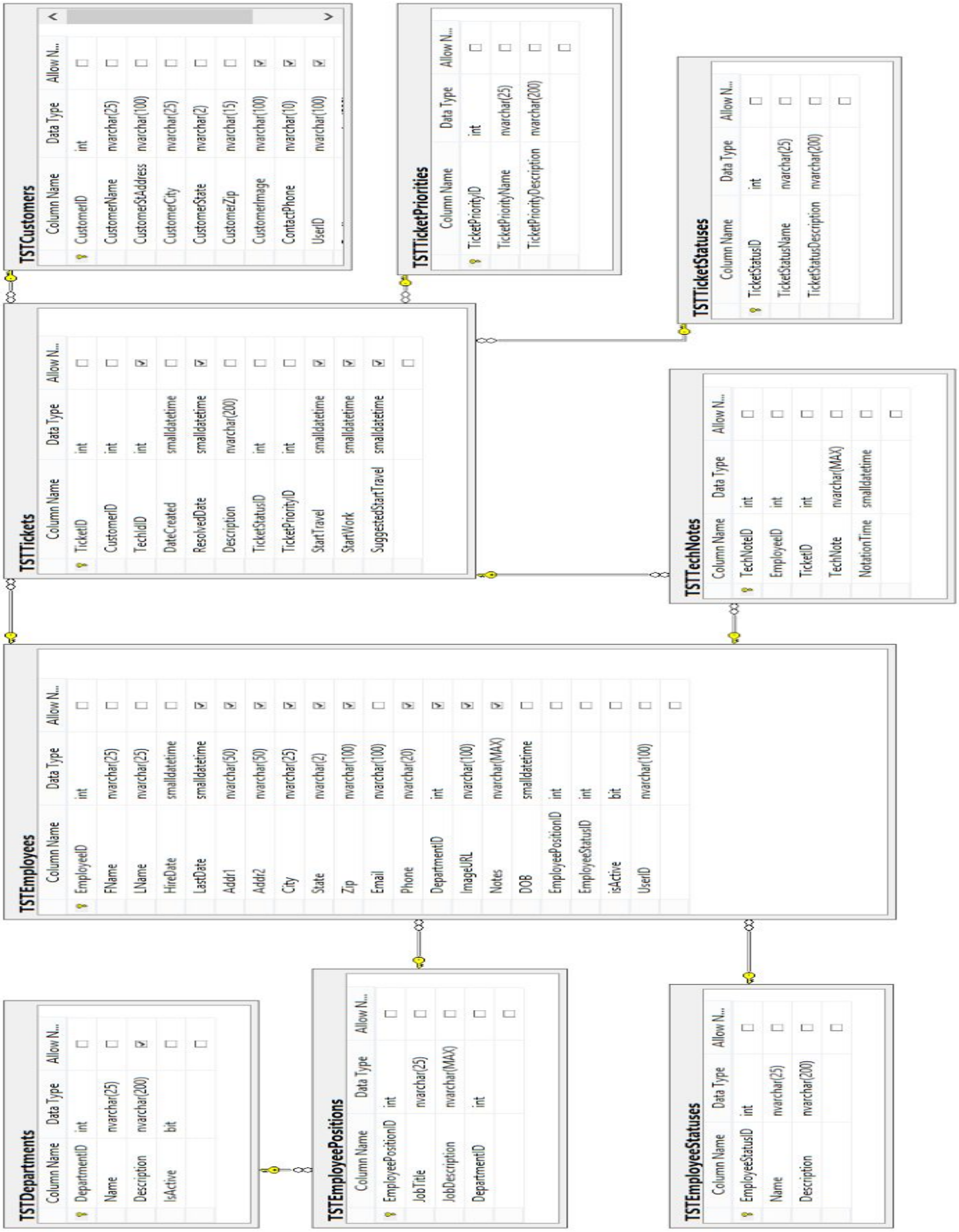
I’m Mike Sherman. I’ve spent most of my adult life working in the customer service industry. I started working for a company called Restaurant Technologies just after my 21st birthday, and stayed put for the next 5 years. I was a service technician, so my responsibilities consisted of completing service requests in a timely manner on a day to day basis. I had no experience at all coming into into it, but I’m a pretty quick learner, and before long I was the youngest master technician the company ever had.

Restaurant Technologies is a cooking oil management company for customers, and the main selling point on their oil system was the technological advances they had made to help user’s extend their oil life. They started rolling out equipment that allowed managers/owners to have direct access to whether their employees were filtering the cooking oil when they were supposed to be, if they were filtering as long as suggested, and every pound of oil the customer used/discarded without having to be there personally to verify the process was being handled efficiently. When I started digging into the system I got to see how everything worked on the IT side of things, and I fell in love. I realized that the mental stimulation I was craving would be found making applications to serve a purpose for the end user. I just didn’t know how I’d get there.

Now, fast forward a couple years. I’m just about done at Centriq, and I finally feel like I’ve found it. Everything I’ve wanted in a career is literally right in front of me, and I’ve never been more excited. I’ll admit, I was skeptical at first. It took me a while to actually pull the string. I had no coding experience at all coming into this, and was truly terrified. Luckily, it all really clicked for me. The troubleshooting, and problem solving skills paid off a little more than I felt they would. The logic of the code is seriously cool. The feeling of making an idea come to life is truly one of a kind.

As an employee, I feel I have a lot to offer. Anyone that’s ever been around me will tell you that I’m a team player. I’ve always been extremely dedicated to the success of not only myself, but the people surrounding me as well. It’s like all the years being in/around baseball has literally pounded into my head “You’re only as strong as your weakest link”. People will also tell you that I’m extremely dedicated and goal oriented. Once I set my mind to getting something done, it usually takes someone literally prying me away from it. I take a lot of pride in everything I do, and I’ve been known to take my personal goals a little too seriously sometimes. That’s enough about me, let’s get to the project.

The base requirements for this project were to have tables to manage the following: Departments, Employees, Tickets, Ticket Statuses , Technician Notes, Notes



TSTCustomers

Column Name	Data Type	Allow N...
CustomerID	int	☐
CustomerName	nvarchar(25)	☐
CustomerStAddress	nvarchar(100)	☐
CustomerCity	nvarchar(25)	☐
CustomerState	nvarchar(2)	☐
CustomerZip	nvarchar(15)	☐
CustomerImage	nvarchar(100)	☑
ContactPhone	nvarchar(10)	☑
UserID	nvarchar(100)	☑

TSTTicketPriorities

Column Name	Data Type	Allow N...
TicketPriorityID	int	☐
TicketPriorityName	nvarchar(25)	☐
TicketPriorityDescription	nvarchar(200)	☐
		☐

TSTTicketStatuses

Column Name	Data Type	Allow N...
TicketStatusID	int	☐
TicketStatusName	nvarchar(25)	☐
TicketStatusDescription	nvarchar(200)	☐
		☐

TSTTickets

Column Name	Data Type	Allow N...
TicketID	int	☐
CustomerID	int	☐
TechID	int	☑
DateCreated	smalldatetime	☐
ResolvedDate	smalldatetime	☑
Description	nvarchar(200)	☐
TicketStatusID	int	☐
TicketPriorityID	int	☐
StartTravel	smalldatetime	☑
StartWork	smalldatetime	☑
SuggestedStartTravel	smalldatetime	☑
		☐

TSTTechNotes

Column Name	Data Type	Allow N...
TechNoteID	int	☐
EmployeeID	int	☐
TicketID	int	☐
TechNote	nvarchar(MAX)	☐
NotationTime	smalldatetime	☐
		☐

TSTEmployees

Column Name	Data Type	Allow N...
EmployeeID	int	☐
FName	nvarchar(25)	☐
LName	nvarchar(25)	☐
HireDate	smalldatetime	☐
LastDate	smalldatetime	☑
Addr1	nvarchar(50)	☑
Addr2	nvarchar(50)	☑
City	nvarchar(25)	☑
State	nvarchar(2)	☑
Zip	nvarchar(100)	☑
Email	nvarchar(100)	☐
Phone	nvarchar(20)	☑
DepartmentID	int	☑
ImageURL	nvarchar(100)	☑
Notes	nvarchar(MAX)	☑
DOB	smalldatetime	☐
EmployeePositionID	int	☐
EmployeeStatusID	int	☐
isActive	bit	☐
UserID	nvarchar(100)	☐
		☐

TSTDepartments

Column Name	Data Type	Allow N...
DepartmentID	int	☐
Name	nvarchar(25)	☐
Description	nvarchar(200)	☑
IsActive	bit	☐
		☐

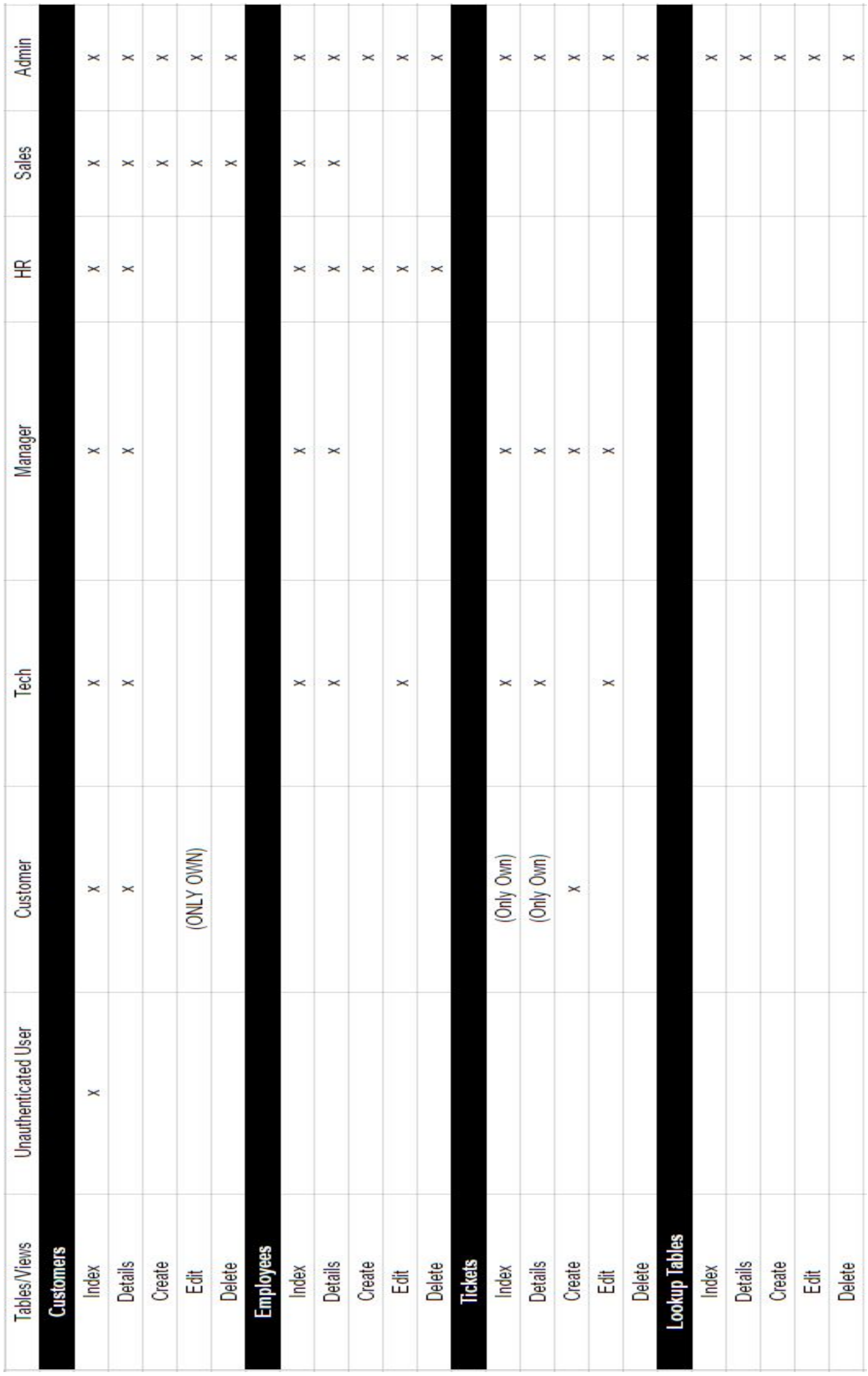
TSTEmployeePositions

Column Name	Data Type	Allow N...
EmployeePositionID	int	☐
JobTitle	nvarchar(25)	☐
JobDescription	nvarchar(MAX)	☐
DepartmentID	int	☐
		☐

TSTEmployeeStatuses

Column Name	Data Type	Allow N...
EmployeeStatusID	int	☐
Name	nvarchar(25)	☐
Description	nvarchar(200)	☐
		☐

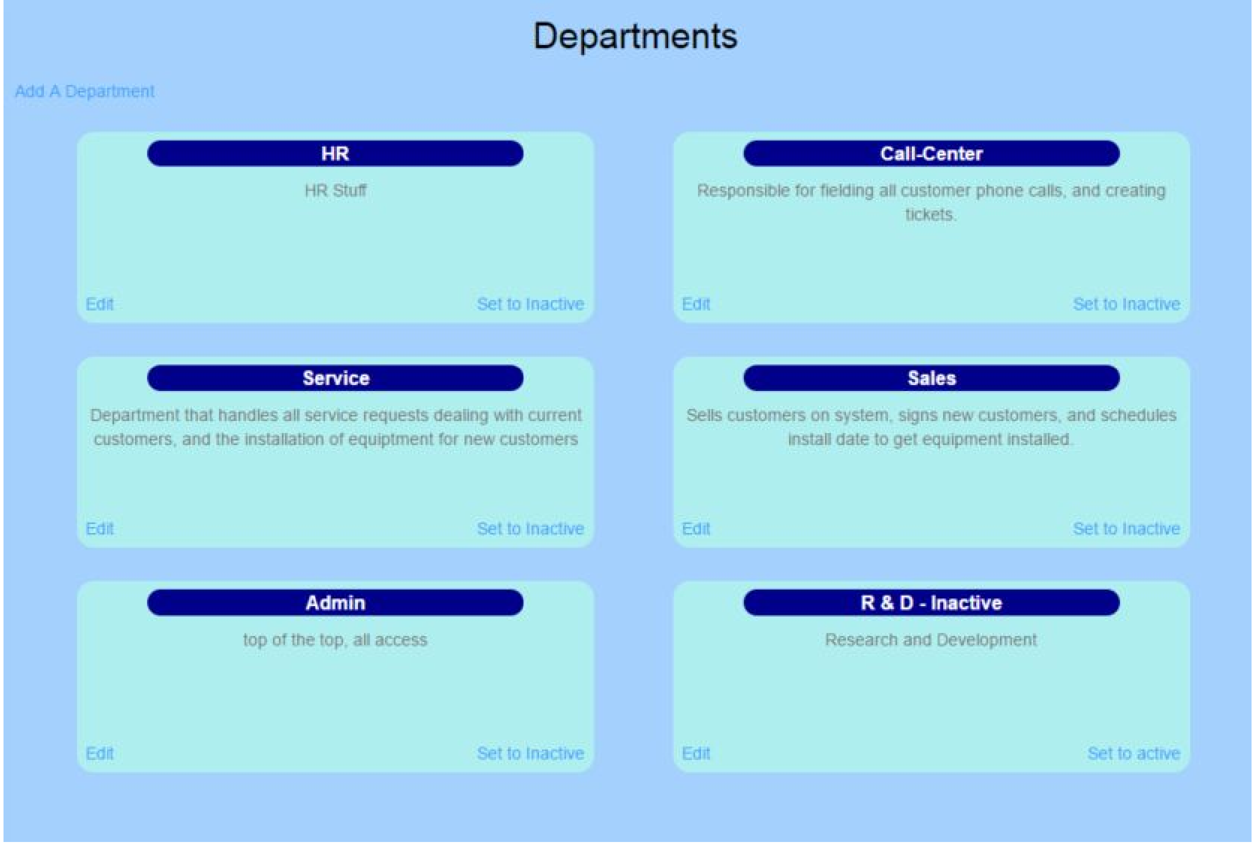
As you can probably tell from the database diagram, I took it a little bit further than that. My goal with this project was for an external customer to be able to submit a service request, which would then be handled internally, assigned by a manager, and “worked” by a technician. I also added a employee status table on-top of the isActive bit, with the intent of eventually tracking/manufacturing an employee’s timesheet. I didn’t quite get to that within the timeframe unfortunately.



Tables/Views	Unauthenticated User	Customer	Tech	Manager	HR	Sales	Admin
Customers							
Index	x	x	x	x	x	x	x
Details		x	x	x	x	x	x
Create						x	x
Edit		(ONLY OWN)				x	x
Delete						x	x
Employees							
Index			x	x	x	x	x
Details			x	x	x	x	x
Create					x		x
Edit			x		x		x
Delete					x		x
Tickets							
Index		(Only Own)	x	x			x
Details		(Only Own)	x	x			x
Create		x		x			x
Edit			x	x			x
Delete							x
Lookup Tables							
Index							x
Details							x
Create							x
Edit							x
Delete							x

Original Use Case Diagram (changed a bit along the way)

Now with the database created, and a rough idea of the roles I wanted, it was time to get to work. After scaffolding everything out, I started by deleting the details views for all of the lookup tables, as they were pretty self explanatory. I <div>'d out all the lookup tables the same way, looking to just get them out of the way to start working on bigger and more interesting things.



Departments

Add A Department

HR

HR Stuff

Edit Set to Inactive

Call-Center

Responsible for fielding all customer phone calls, and creating tickets.

Edit Set to Inactive

Service

Department that handles all service requests dealing with current customers, and the installation of equiptment for new customers

Edit Set to Inactive

Sales

Sells customers on system, signs new customers, and schedules install date to get equipment installed.

Edit Set to Inactive

Admin

top of the top, all access

Edit Set to Inactive

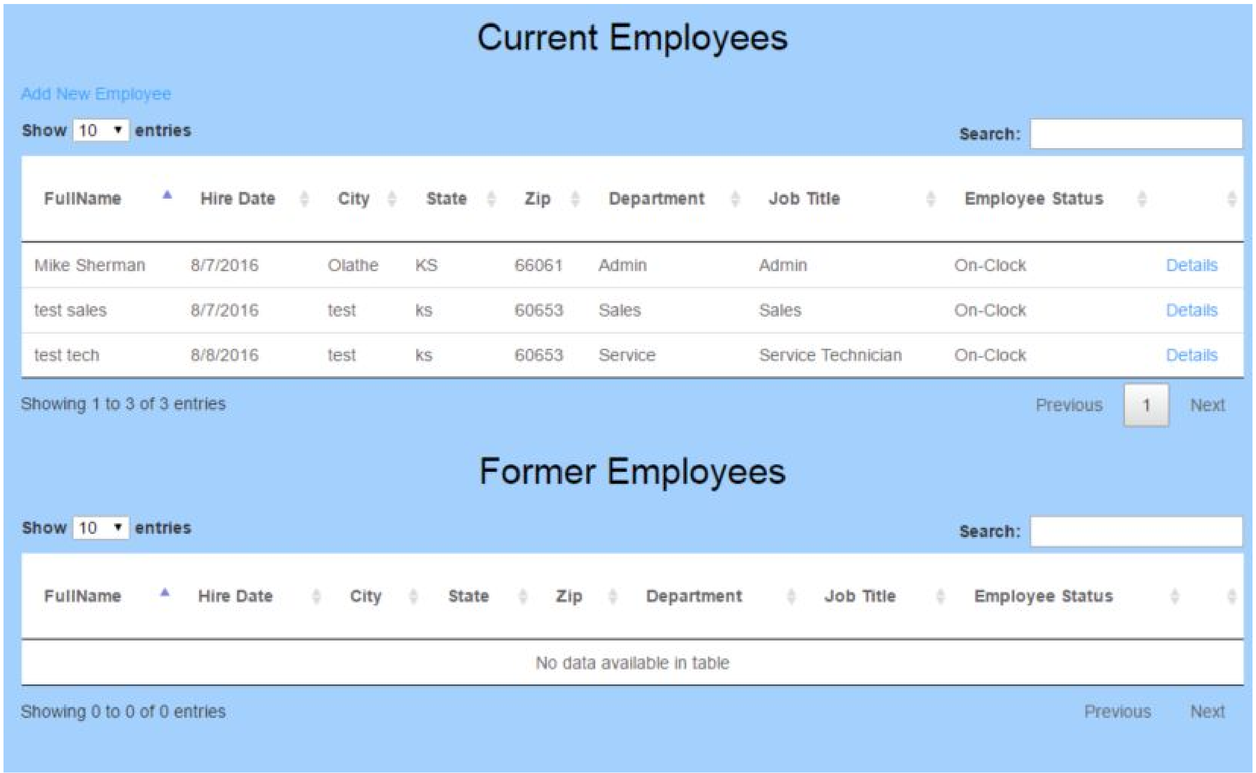
R & D - Inactive

Research and Development

Edit Set to active

This is the index view for all of my lookup tables. This is from an admin role, where they have access to the edit, and deactivate/reactivate button (soft delete). In this case, any user with access to the departments index but not access to crud functionality, would not see the inactive R&D department.

Index and Details view from the “Admin” role



Current Employees

Add New Employee

Show 10 entries

Search:

FullName	Hire Date	City	State	Zip	Department	Job Title	Employee Status	
Mike Sherman	8/7/2016	Olathe	KS	66061	Admin	Admin	On-Clock	Details
test sales	8/7/2016	test	ks	60653	Sales	Sales	On-Clock	Details
test tech	8/8/2016	test	ks	60653	Service	Service Technician	On-Clock	Details

Showing 1 to 3 of 3 entries

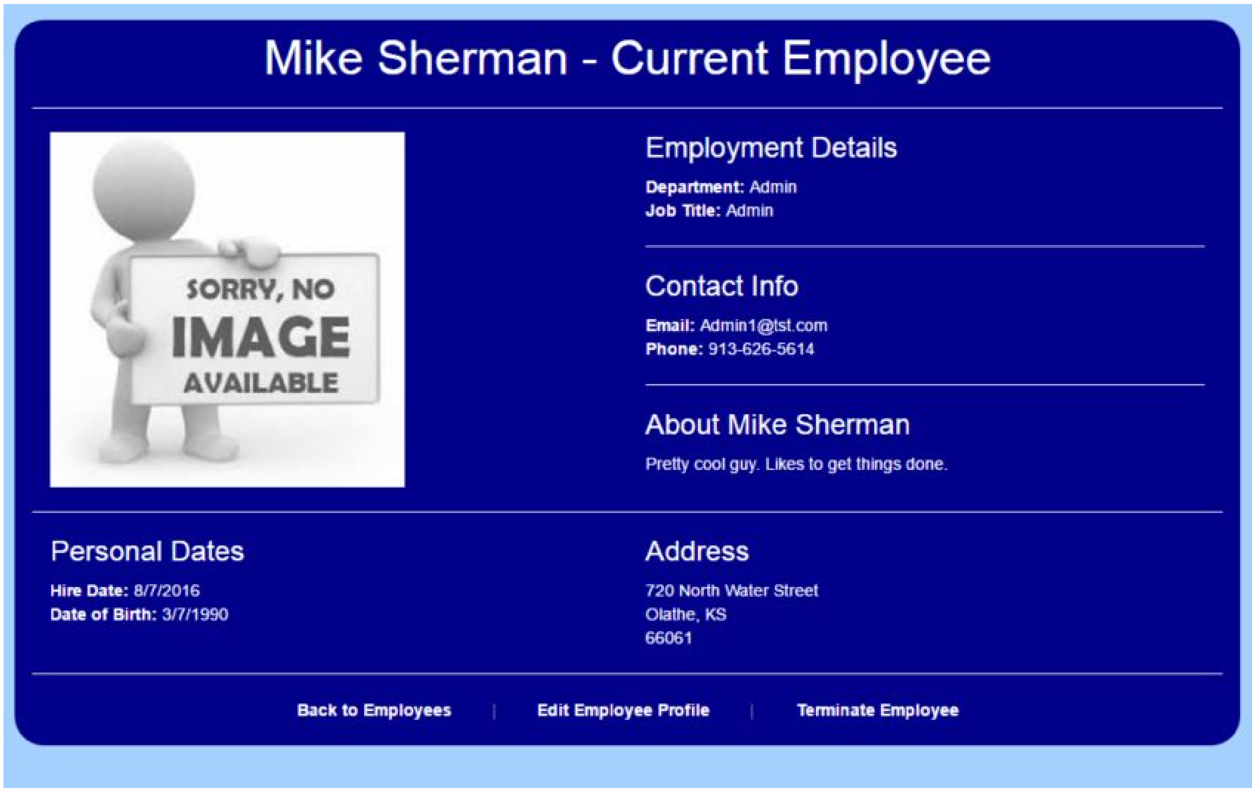
Former Employees

Show 10 entries

Search:

FullName	Hire Date	City	State	Zip	Department	Job Title	Employee Status	
No data available in table								

Showing 0 to 0 of 0 entries



Mike Sherman - Current Employee

SORRY, NO
IMAGE
AVAILABLE

Employment Details

Department: Admin
Job Title: Admin

Contact Info

Email: Admin1@tst.com
Phone: 913-626-5614

About Mike Sherman

Pretty cool guy. Likes to get things done.

Personal Dates

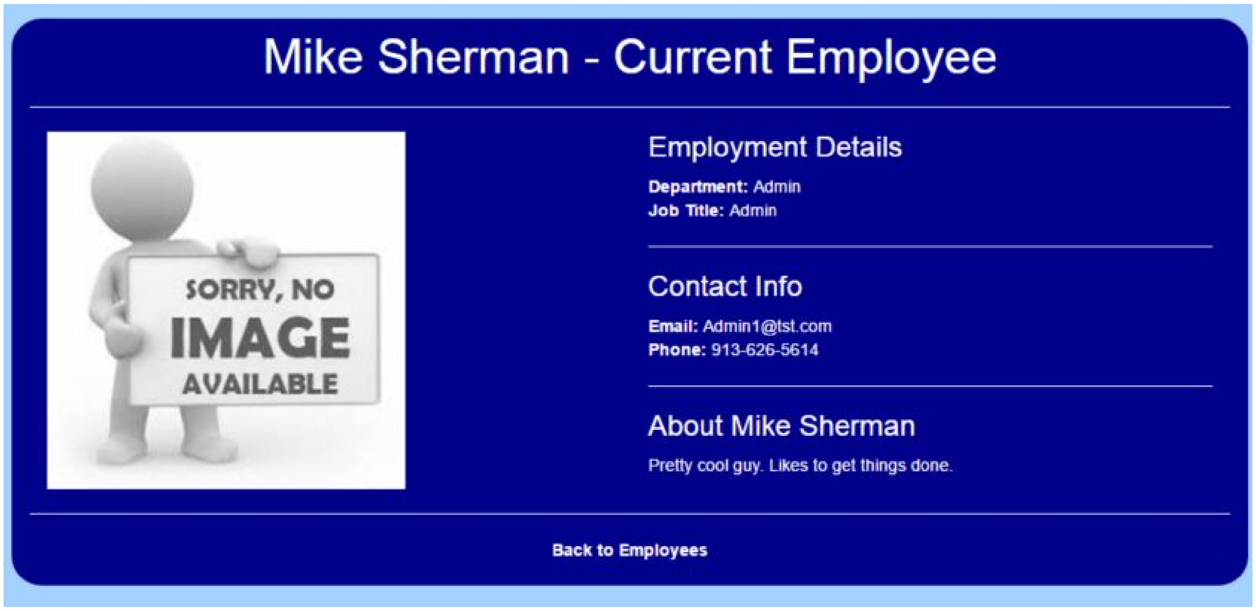
Hire Date: 8/7/2016
Date of Birth: 3/7/1990

Address

720 North Water Street
Olathe, KS
66061

Back to Employees | Edit Employee Profile | Terminate Employee

Same views from the “Tech” role



Mike Sherman - Current Employee

Employment Details

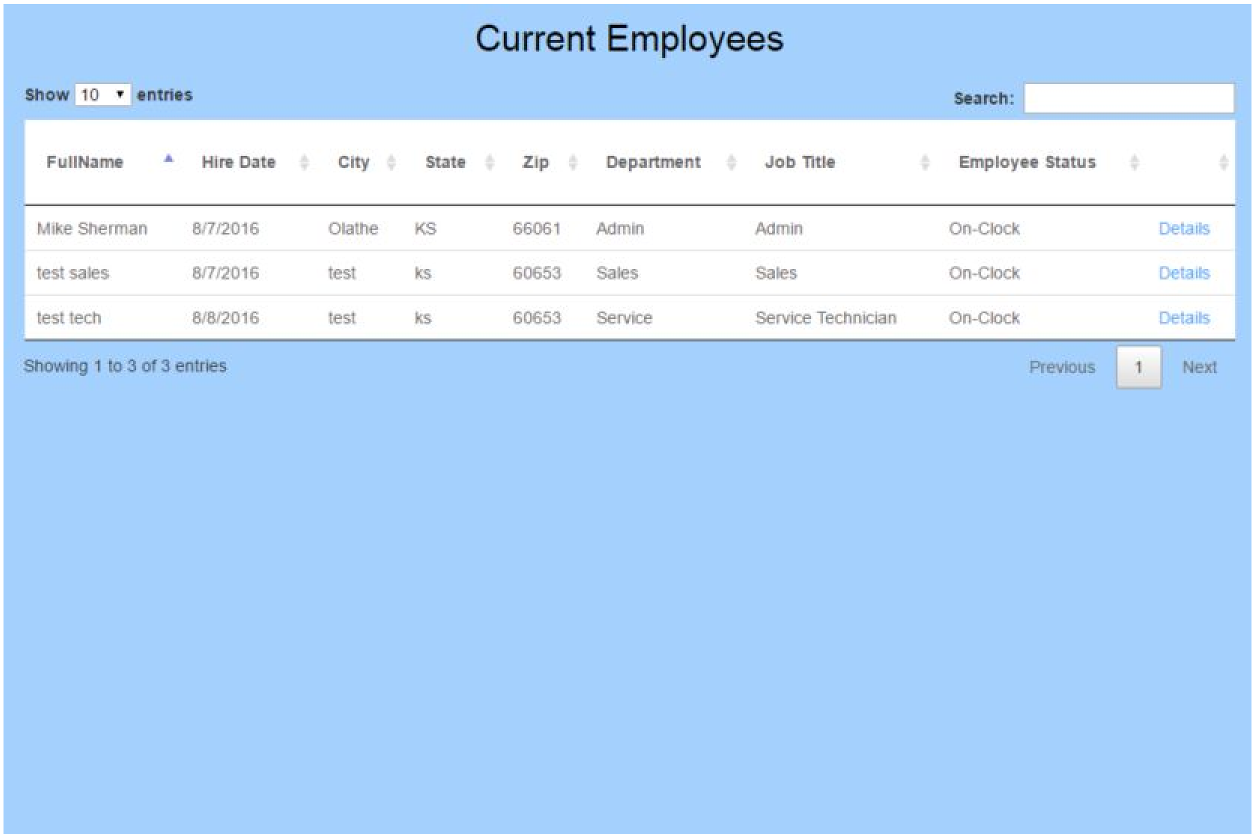
Department: Admin
Job Title: Admin

Contact Info

Email: Admin1@tst.com
Phone: 913-626-5614

About Mike Sherman

Pretty cool guy. Likes to get things done.



Current Employees

FullName	Hire Date	City	State	Zip	Department	Job Title	Employee Status	
Mike Sherman	8/7/2016	Olathe	KS	66061	Admin	Admin	On-Clock	Details
test sales	8/7/2016	test	ks	60653	Sales	Sales	On-Clock	Details
test tech	8/8/2016	test	ks	60653	Service	Service Technician	On-Clock	Details

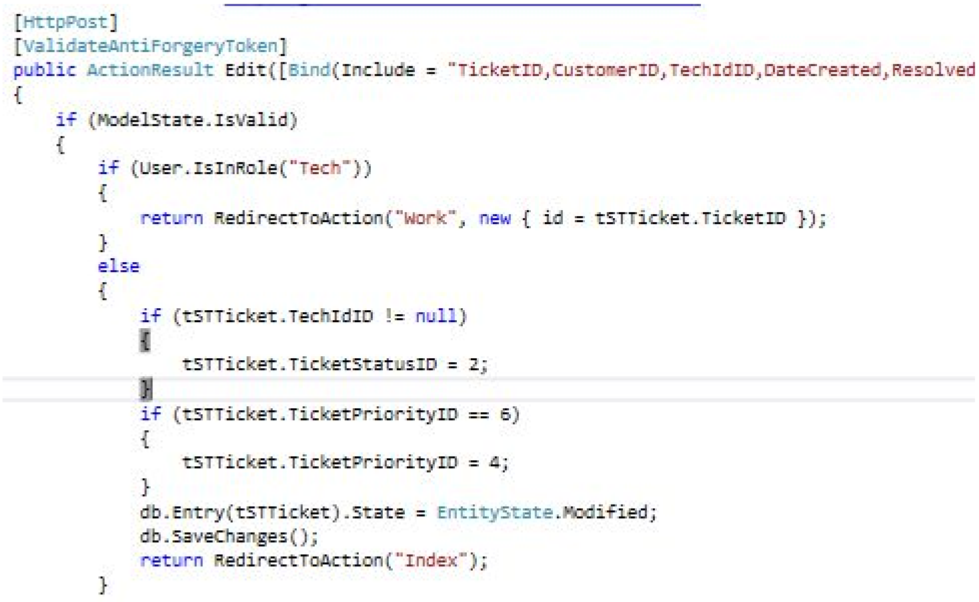
Showing 1 to 3 of 3 entries

Notice that the former employees section, and all the links to crud functionality aren't visible to the user not in the role HR or Admin. The customer's index and details act the same way, with the role able to access their crud functionality being sales, instead of Admin. Every authenticated user also has the capability to edit their own profile. In their details view, they'll have the link to edit visible.

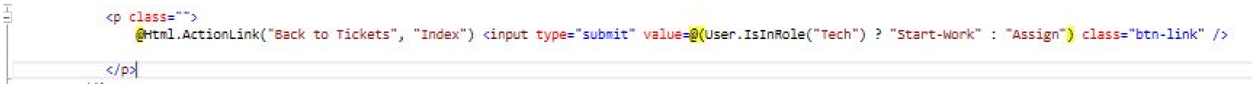
Now this is where things got interesting for me. The main focus of the project for me was all encompassing the tickets themselves. The only people with the capability to create a ticket are customers, service manager, and the admin. Customers when they sign in see an extra link in the nav bar that reads "Request Service". This link takes them to the tickets create, where they describe their issue. All required fields are automatically populated in the controller based on which customer is actually signed in.

All of the roles with access to the index see it a different way. Customers only see tickets they've submitted. Techs see 2 separate tables with the first containing their "Calls for the day" and the second containing anything that had yet to be assigned. The Manager/Admin see 3 tables. The first with unassigned tickets. The second with tickets that have already been assigned but yet to be resolved. The last table contains all resolved tickets.

I changed the link for the edit view to show "Assign" when in the roles that would be in charge of assigning the calls, and "Start Travel" when in the tech role. The view only shows the values needing to be assigned in the ticket for the admin/manager, and an assign button to confirm the changes. For the tech it shows no editable fields, but does timestamp the start travel time and change the status of the call from "assigned" to "In-route". The tech's new link inside the edit view reads "start work". Clicking this link timestamps the ticket again for the arrival time, and changes the status to "on-site". The tech is returned to the same view, but the link now reads “Service Complete”. Clicking service complete closes the ticket, changing the status to resolved, and again timestamping the ticket.



```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include = "TicketID,CustomerID,TechIdID,DateCreated,Resolved
{
    if (ModelState.IsValid)
    {
        if (User.IsInRole("Tech"))
        {
            return RedirectToAction("Work", new { id = tSTTicket.TicketID });
        }
        else
        {
            if (tSTTicket.TechIdID != null)
            {
                tSTTicket.TicketStatusID = 2;
            }
            if (tSTTicket.TicketPriorityID == 6)
            {
                tSTTicket.TicketPriorityID = 4;
            }
            db.Entry(tSTTicket).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
```

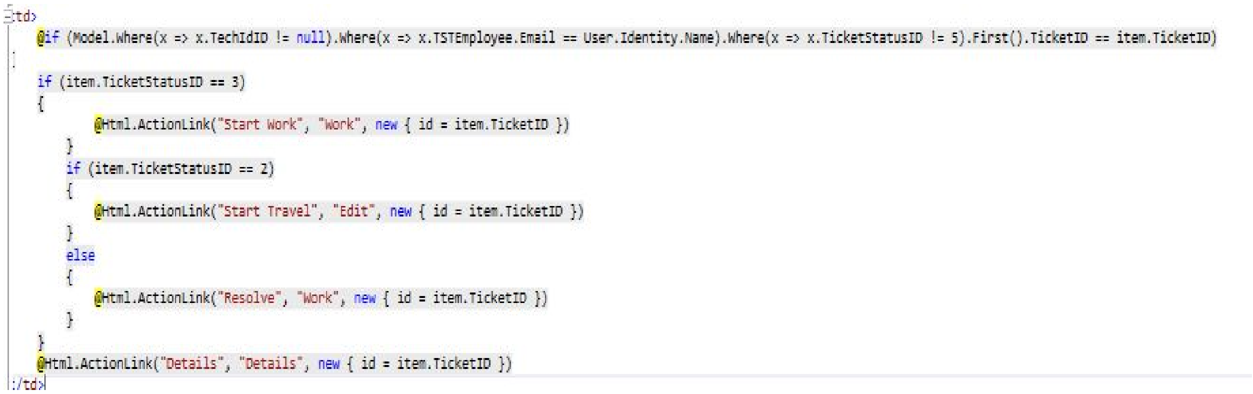


```
<p class="">
    @Html.ActionLink("Back to Tickets", "Index") <input type="submit" value=@(User.IsInRole("Tech") ? "Start-Work" : "Assign") class="btn-link" />

</p>
```

Then I realized that the technicians had the ability to push start travel on multiple tickets at the same time, which at first pretty much crushed my hopes and dreams for the feature. After a ridiculous amount of time, I got it to where the link changed based on the status of the ticket, and that the edit view was only available on the first ticket shown in the view.

The code that wound up getting me there.



```
<td>
    @if (Model.Where(x => x.TechIdID != null).Where(x => x.TSTEmployee.Email == User.Identity.Name).Where(x => x.TicketStatusID != 5).First().TicketID == item.TicketID)

    if (item.TicketStatusID == 3)
    {
            @Html.ActionLink("Start Work", "Work", new { id = item.TicketID })
        }
        if (item.TicketStatusID == 2)
        {
            @Html.ActionLink("Start Travel", "Edit", new { id = item.TicketID })
        }
        else
        {
            @Html.ActionLink("Resolve", "Work", new { id = item.TicketID })
        }
    }
    @Html.ActionLink("Details", "Details", new { id = item.TicketID })
</td>
```

(I swear that other curly brace is there, it just ended up falling off the snip)

All in all, I learned a ridiculous amount about coding in general, just from this project itself. Some of my views remained unstyled, because some of the custom controller work needed to complete to get the functionality I wanted working was a bit more important to me. The planning process seemed overrated to me, and I suffered the first few days of the project just trying to figure out where to look next. Starting to mentally freak out a bit, watching the time slip away, I made a checklist, and followed it step by step. I definitely learned my lesson.

If you'd like to test the functionality of the project, or run through the "ticket tracking" process yourself, I've created a few test accounts you can use on the live site. If you have any feedback I'd love to hear it. Email me at

mike_sherman11@outlook.com

testtech@tst.com = tech role
testcustomer@tst.com = customer role
testmanager@tst.com = manager role

Their passwords are P@ssw0rd.
If you actually read all of this, I appreciate it. Thanks for your time.