



# Turner Madick

[turnermadick.com](http://turnermadick.com)

[linkedin.com/in/turner-madick/](https://www.linkedin.com/in/turner-madick/)

[tcmadick@gmail.com](mailto:tcmadick@gmail.com)

## Internal Job Board for Dunder Mifflin

[fsdp.turnermadick.com](http://fsdp.turnermadick.com)

Presentation Date: July 12, 2019

## About Me

My name is Turner Madick. I have grown up using computers my entire life. My dad got me into PC gaming at 3 years old with the original Diablo, though my taste in gaming moved more to strategy games. I became pretty tech savvy throughout my childhood by trouble shooting my own PC problems. The “Googling Skills” I picked up at a young age came in very handy.

I decided I wanted a career in the IT field after high school and attended MCC’s Network Admin & Security program for a few semesters before taking my first coding course, C++. I enjoyed coding so much that I transitioned into Centriqs .NET Full-Stack Web Development program. Centriq furthered my knowledge of languages, such as HTML5, CSS3, JavaScript, C#, and SQL.

Having some college experience outside the IT field, coding felt so much different than anything I had learned. It’s much less about memorizing facts, and more about breaking down problems to find solutions. Puzzles and strategy games have always been forte, so coding just felt natural. As my learning experience here at Centriq quickly comes to an end, I look forward to furthering my knowledge and applying my skills in my newly found passion.

# Project Brief

The objective is to create an Internal Job Board that allows Dunder Mifflin to share job openings from all their branches across the United States. User's should have the following functionality based on their roles:

## Admin/Corporate Capabilities:

- CRUD functionality for User Accounts, Locations, Positions, Open Positions and Applications

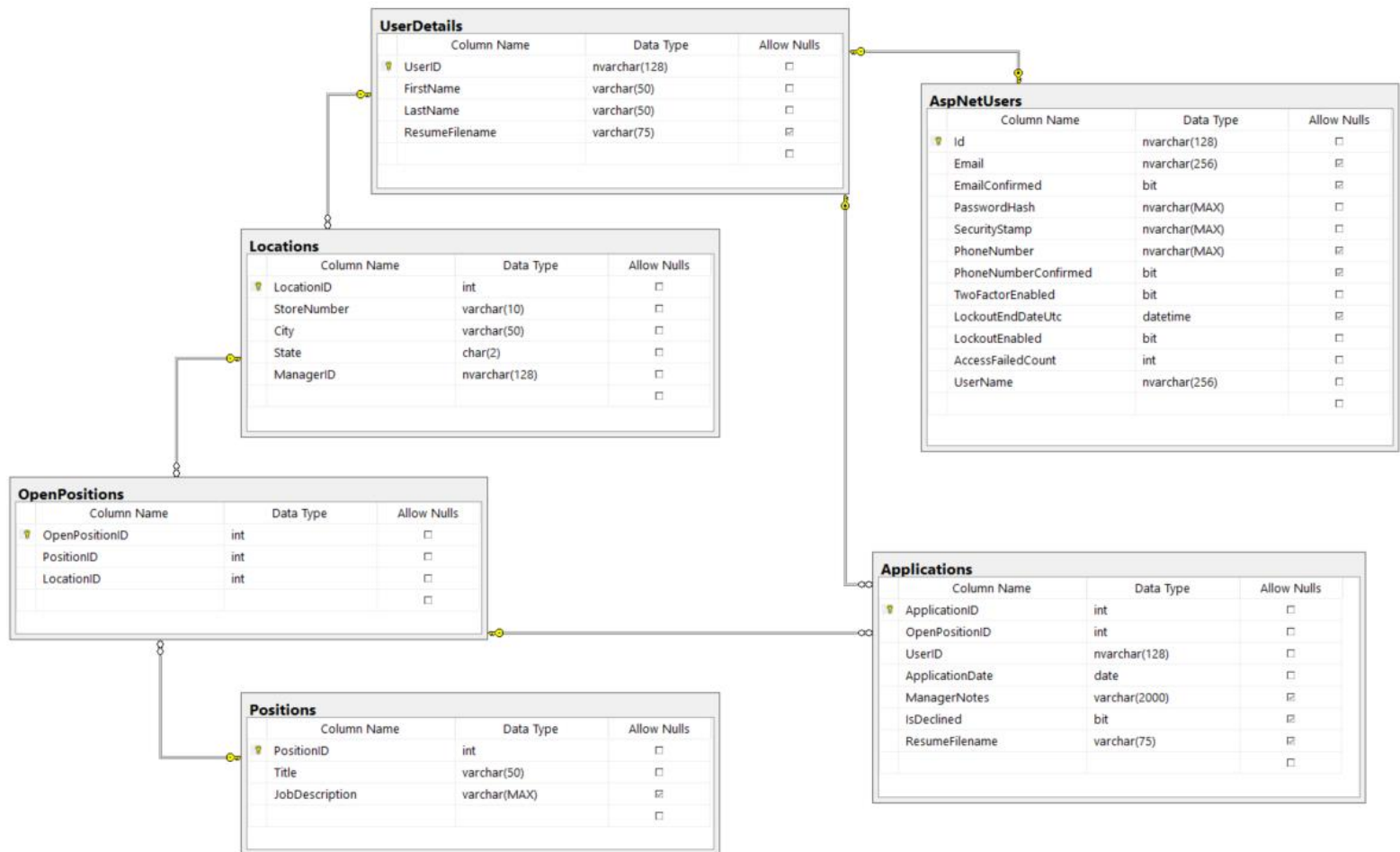
## Manager Capabilities:

- CRUD functionality for positions at their location
- View and manage applications for positions at their location

## Employee Capabilities:

- CRUD functionality for their account
- Upload a resume to their account
- Apply for positions with one click and view the application status

# Database Diagram



# One-Click Apply

```
[HttpPost]
[Authorize(Roles = "User, Manager")]
public ActionResult Apply(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    OpenPosition openPosition = db.OpenPositions.Find(id);
    if (openPosition == null)
    {
        return HttpNotFound();
    }
    var user = User.Identity.GetUserId();

    //Delete old app
    Application oldApp = db.Applications.Where(c => c.OpenPositionID == id && c.UserID == user).SingleOrDefault();
    if (oldApp != null)
    {
        db.Applications.Remove(oldApp);
        db.SaveChanges();
    }

    Application app = new Application()
    {
        OpenPositionID = openPosition.OpenPositionID,
        UserID = user,
        ApplicationDate = DateTime.Now,
        IsDeclined = null,
        ResumeFilename = db.UserDetails.Where(c => c.UserID == user).Select(c => c.ResumeFilename).Single(),
        ManagerNotes = null
    };

    db.Applications.Add(app);
    db.SaveChanges();
    return RedirectToAction("Index", "Applications");
}
```

---

# Identity Example

```
<tbody>
@foreach (var item in Model)
{
    if (item.UserID == User.Identity.GetUserId() || User.IsInRole("Admin") || (User.IsInRole("Manager") &&
        item.OpenPosition.Location.ManagerID == User.Identity.GetUserId()))
    {
        if (item.IsDeclined == true)
        {
            <tr>
            <td>
                @Html.DisplayFor(modelItem => item.OpenPosition.Position.Title)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.OpenPosition.Location.City),
                @Html.DisplayFor(modelItem => item.OpenPosition.Location.State)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.UserDetail.FullName)
            </td>
            @if (User.IsInRole("Admin") || User.IsInRole("Manager"))
            {
                <td>
                    @Html.ActionLink("Declined", "Edit", new { id = item.ApplicationID })
                </td>
            }
            </tr>
        }
    }
}
```

---

# UserID Filter via Role

```
public async Task<ActionResult> Create()
{
    List<ApplicationUser> managersAndAdmins = await UserManager.Users.ToListAsync();
    var userRole = db.UserDetails.ToList();

    for (int i = managersAndAdmins.Count - 1; i >= 0; i--)
    {
        for (int x = managersAndAdmins[i].Roles.Count - 1; x >= 0; x--)
        {
            //Only show users with roles "Admin" or "Manager"
            if (managersAndAdmins[i].Roles.ToList()[x].RoleId == "e245ff5d-e4df-4606-b056-d3e50641d309" ||
                managersAndAdmins[i].Roles.ToList()[x].RoleId == "bd7b11e4-df24-4fd2-acd5-60ec5eed8ca")
            {
                break;
            }
            else if (managersAndAdmins[i].Roles.ToList()[x] == managersAndAdmins[i].Roles.ToList().Last())
            {
                userRole.Remove(userRole.Where(m => m.UserID == managersAndAdmins[i].Id).Single());
            }
        }
    }
    ViewBag.ManagerID = new SelectList(userRole, "UserID", "FullName");
    return View();
}
```

---

# Account Register

```
string image = "NoResume.pdf";

if (resume != null)
{
    image = resume.FileName;

    string ext = image.Substring(image.LastIndexOf("."));

    string[] goodExts = new string[] { ".jpg", ".jpeg", ".png", ".gif", ".pdf", ".doc", ".docx" };

    if (goodExts.Contains(ext.ToLower()))
    {
        image = Guid.NewGuid() + ext;

        resume.SaveAs(Server.MapPath("~/Content/images/Resumes/" + image));
    }
    else
    {
        image = "NoResume.pdf";
    }
}

UserDetail info = new UserDetail()
{
    UserID = user.Id,
    FirstName = model.FirstName,
    LastName = model.LastName,
    ResumeFilename = image
};

FinalEntities ctx = new FinalEntities();
ctx.UserDetails.Add(info);
ctx.SaveChanges();
```

---

# Login Partial

```
@using JobBoard.DATA.EF
@using Microsoft.AspNet.Identity
@if (Request.IsAuthenticated) {
    using (Html.BeginForm("LogOff", "Account", FormMethod.Post, new { id = "logoutForm" })) {
        @Html.AntiForgeryToken()

        FinalEntities db = new FinalEntities();
        string userID = User.Identity.GetUserId();
        UserDetail currentUser = db.UserDetails.Where(c => c.UserID == userID).SingleOrDefault();

        <li>
            @Html.ActionLink(currentUser.FullName, "Edit", "UserDetails", new { id = User.Identity.GetUserId() }, null)
        </li>
        <li><a href="javascript:document.getElementById('logoutForm').submit()">Log Off</a></li>
    }
}
else {
    <li>@Html.ActionLink("Register", "Register", "Account", routeValues: null, htmlAttributes: new { id = "registerLink" })</li>
    <li>@Html.ActionLink("Log in", "Login", "Account", routeValues: null, htmlAttributes: new { id = "loginLink" })</li>
}
```

## Agile Process

In order to simulate an Agile development environment, we performed daily standups in class during our sprint weeks. During stand up we went over what we had recently finished and what we planned to do next.

## Post Project Reflection

Going into the project my biggest worry was the front-end styling. I felt more comfortable working in middle-tier languages. Working on this project helped a ton with my understanding of writing JavaScript / jQuery as I worked with the animations of the template.

One thing I didn't expect to go so in-depth with was customizing Identities. Pulling data from IdentitySamples proved to be more difficult than expected when the tables aren't added / linked to the .edmx file. After completing v1 of this project, I feel confident and comfortable navigating through the scaffolded controllers / views and calling to specific data fields to use branching logic.



# Turner Madick

FULL-STACK DEVELOPER

816-550-0138  
tcmadick@gmail.com  
[www.turnermadick.com](http://www.turnermadick.com)



## Qualifications

---

- Solid foundational knowledge of designing and developing full-stack web applications using .NET framework.
- Complex problem-solving skills, time management, and attention to detail honed through experience in a customer service and coaching role.

## Technical Skills

---

**Front End:** HTML5, JavaScript, jQuery, jQueryUI, CSS3, Responsive/Mobile Web Development, Bootstrap

**Middle Tier:** Visual Studio, C#.NET, C++, ASP.NET, LINQ, MVC, EF

**Back End:** ADO.NET, SQL, SQL Server, SSMSE jQuery

## Independent Development Projects

---

- **Personal Site:** [www.turnermadick.com](http://www.turnermadick.com)
- **U Store:** Created an application for managing product data. Application is built to simulate an online store with a shopping cart
- **Dungeon App:** Created a console window application using C#. Application simulates a combat-based decision tree.

## Technical Training

---

**CENTRIQ TRAINING | KANSAS CITY, MO**  
FULL-STACK WEB DEVELOPER PROGRAM

**MARCH 2019 – PRESENT**

### *Core Competencies:*

- MVC Framework
- Trouble Shooting & Debugging
- Source Control
- Team Project
- Website Deployment
- Pair Programming
- Code Review
- Professionalism, Teamwork, Problem Solving & Effective Communication

**MCCKC | INDEPENDENCE, MO**  
COURSES COMPLETED IN NETWORK ADMINISTRATION

**AUGUST 2017 – MAY 2018**

### *Core Competencies:*

- Basic Computer Concepts
- Networking Concepts

- Hardware
- Operating Systems
- Safety

## Professional Experience

---

### **HARVEY'S | KANSAS CITY, MO**

**JUNE 2017 – PRESENT**

SERVER, BARTENDER, EVENT WORKER

- Developed attention to detail taking and keeping track of many different customer orders at one time.
- Memorized all options and sub options of menu selections.
- Became fluent in navigating and utilizing POS systems.

### **INDEPENDENCE SCHOOL DISTRICT | INDEPENDENCE, MO**

**JUNE 2015 – FEB 2019**

ASSISTANT WRESTLING COACH

- Utilized active learning skills while determining and using the best learning methods for each kid on the team.
- Assisted the head coach with planning practices and deciding which areas need improvement on the most.
- Resolved conflicts kids ran into, such as accepting and learning from defeat.