



BRENT KUHLMAN

Friday, December 12, 2018



LEAGUE SECRETARY
Bowling League Reservation System

fsdb.kuhlaid.com

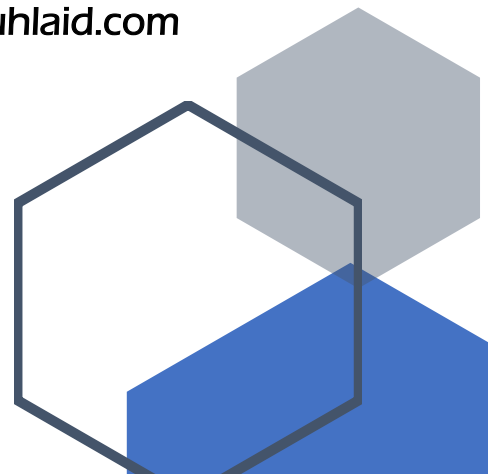


Table of Contents

Table Of Contents / Notes.....	1
About Me.....	2
Project Requirements	3
ER Diagram.....	4
Use Case Diagram.....	5
Custom Code Snippets.....	6
Bookings Controller	6
Locations UI Tile View / League Controller.....	7
Resume.....	8



About Me

For the last 7 years I've held various positions in the Customer Service and Warehouse industry with my most recent position being a Bowling Attendant at a local Main Event. While I enjoyed my time at these positions, I didn't feel like I wanted to do this for the rest of my life.

Growing up I played a variation of console video games and found an interest in PC gaming which requires decent hardware to be able to even run some of the simplest games. Working at a job where I made minimum wage, I saved up a lot of money and taught myself how to build computers utilizing online resources. I constructed the most intricate computer. After learning how to construct a computer and know the hardware portion, it opened a whole new world that is the software side. This fascinated me by how much you can do on the software side of computing, the results are limitless.

Attending Centriq has been one of the most rewarding experiences of my life. Coming into the track I had previous knowledge in programming from college (Programming fundamentals, Object oriented programming with C-Sharp, and courses pertaining to game development) but wanted to expand that knowledge. While I learned a lot through this course, I have the determination to learn more and keep learning; That is the lifestyle of this field. Learning more and more as technology advances and branches out. I have purchased a few online courses utilizing some of the languages we have already learned to build from, as well as a few new frameworks and languages to further my education.

I chose this project because It let me go off one of my hobbies, bowling. I can create a solution to solve real world problems and make it more convenient for others in this line of work as well as for the consumer.



Project Requirements

Out of the three briefs we were given, I decided to go with the Reservation System since it was the best fit for my concept. The Reservation System had certain required functionality, roles and accessibility limitations. We had to create at least three roles, including admin with full CRUD functionality.

Technical Requirements

- Data driven
- Login Capabilities (Identity)
- ASP.Net MVC
- CRUD functionality
- Responsive design (UI)

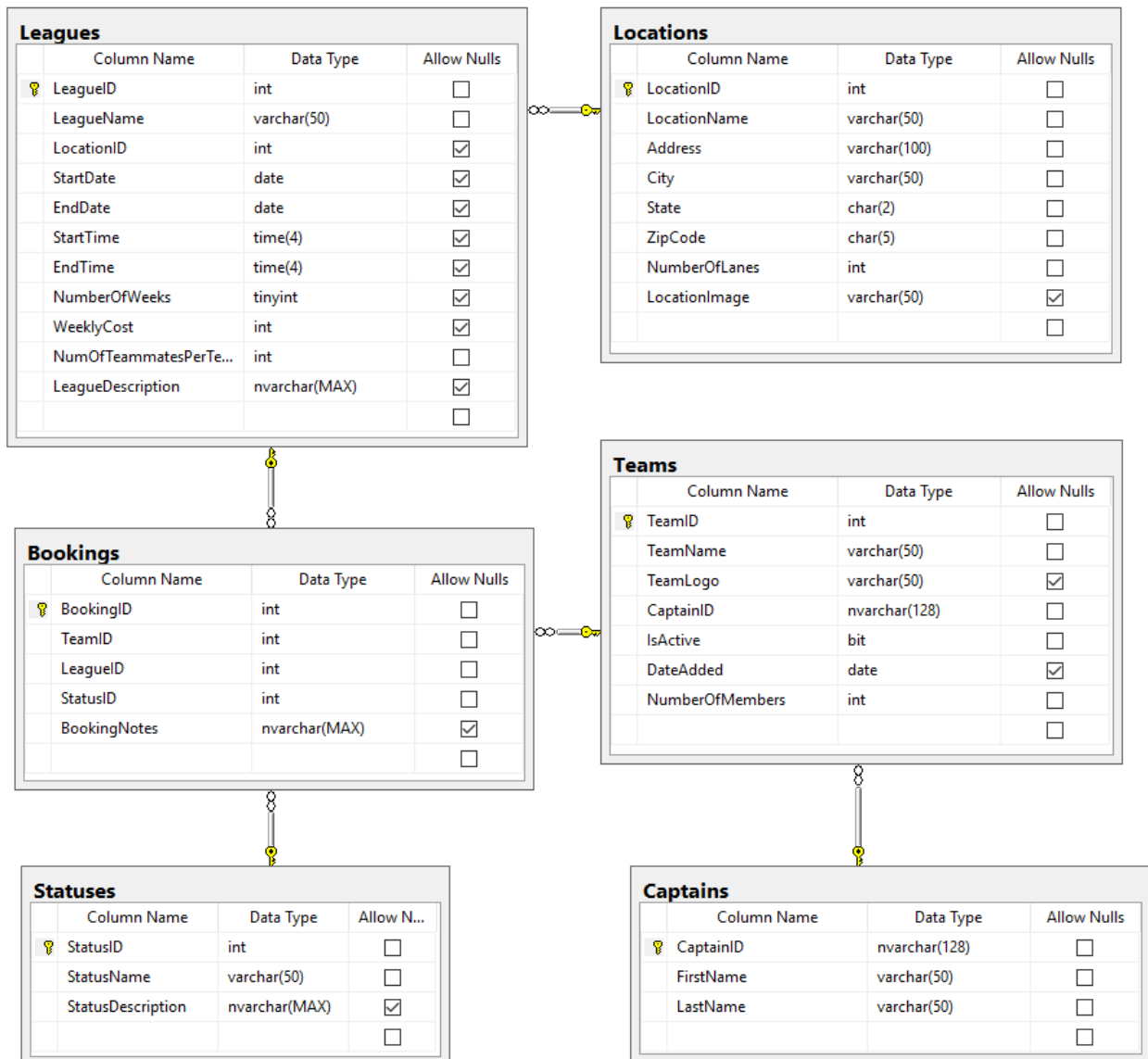
Programming Challenges

- Clients should be able to make a reservation with their assets at any location as long as the reservation limit for that location has not been met. The application is mobile responsive.
- Admins should have total CRUD functionality and be able to override the reservation limit when adding a reservation.
- Date added, and date submitted should be populated automatically.



ER Diagram

I customized my database as much as my project manager would let me to accomplish what I wanted. Ultimately, I wanted to branch this even further to store each users scores for each game as well, but unfortunately we were on a time crunch.



Use/Case Diagram

	Anonymous	Captain	Employee	Admin
Captain				
Index			✓	✓
Create			✓	✓
Edit			✓	✓
Delete			✓	✓
Booking				
Index		✓ Only theirs	✓	✓
Create		✓	✓	✓
Edit		✓ Only theirs	✓	✓
Delete		✓ Only theirs	✓	✓
Leagues				
Index	✓	✓	✓	✓
Create			✓	✓
Edit			✓	✓
Delete			✓	✓
Locations				
Index	✓	✓	✓	✓
Create			✓	✓
Edit			✓	✓
Delete			✓	✓
Statuses				
Index			✓	✓
Create			✓	✓
Edit			✓	✓
Delete			✓	✓
Teams				
Index		✓ Only theirs	✓	✓
Create		✓ Only theirs	✓	✓
Edit		✓ Only theirs	✓	✓
Delete		✓ Only theirs	✓	✓

In terms of CRUD functionality for each role, this table accurately reflects the functionality for each role in the application.

Custom Code

The below code snippet shows the Bookings Controller where it implements logic pertaining to the 'Create' action. If you look below, the admin can create the reservation and bypass the logic to apply the reservation limit to other users. If you are a user then it will deny you from creating a reservation for a league that is full/booked. The logic also checks to verify that the number of players on the team you are trying to register matches the number of players required for this league.

```
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "BookingID,TeamID,LeagueID,StatusID,BookingNotes")] Booking booking)
{
    ViewBag.Title = title;
    booking.StatusID = 1;
    //If the user is in the role 'Admin' Allow them to override the limit and add the booking.
    if (User.IsInRole("Admin"))
    {
        if (ModelState.IsValid)
        {
            uow.BookingRepository.Add(booking);
            uow.Save();
            return RedirectToAction("Index");
        }
    }
    else
    {
        //Create team obj
        Team t = uow.TeamRepository.Find(booking.TeamID);
        //Create League obj
        League l = uow.LeagueRepository.Find(booking.LeagueID);
        //Make sure the Team number of members fits the Leagues Number of members required for a team.
        if (t.NumberOfMembers == l.NumberOfTeamatesPerTeam)
        {
            int count = uow.BookingRepository.Get().Where(b => b.LeagueID == booking.LeagueID).Count();
            League loc = uow.LeagueRepository.Get().Where(x => x.LeagueID == booking.LeagueID).Single();
            int limit = loc.Location.NumberOfLanes;
            if (count < limit) //Check to make sure the reservation limit hasn't been reached.
            {
                if (ModelState.IsValid)
                {
                    uow.BookingRepository.Add(booking);
                    uow.Save();
                    return RedirectToAction("Index");
                }
            }
            else //If they dont, populate a viewbag error message to be sent back with the return view.
            {
                ViewBag.ReservationLimit = "This league is currently full, please consider a different option.";
            }
        }
        else //If they dont, populate a viewbag error message to be sent back with the return view.
        {
            ViewBag.MotValid = "The number of members on your team must match the number of players required for the league.";
        }
    }
    ViewBag.LeagueID = new SelectList(uow.LeagueRepository.Get(), "LeagueID", "LeagueName"); //Drop-down list for Leagues.
    ViewBag.TeamID = new SelectList(uow.TeamRepository.Get(), "TeamID", "TeamName"); //Drop-down list for Teams.
    ViewBag.StatusID = new SelectList(uow.StatusRepository.Get(), "StatusID", "StatusName"); //Drop-down list for Statuses.
    return View(booking);
}
```

I'm proud to show off this custom tile view, It has a link to return you back to the leagues table with a filtered list to that specific location. Alternatively, it also includes a 'Toggle View' button if you prefer the table view.

The below code snippet shows the create action in the league controller. Since the reservation is occurring weekly based on the day that it starts, this logic essentially is to auto calculate the number of weeks between the start date and the end date.

```
[Authorize(Roles = "Admin,Employee")]
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "LeagueID,LeagueName,LocationID,StartDate,EndDate,StartTime,EndTime,NumberOfWeeks,WeeklyCost,NumOfTeammatesPerTeam,LeagueDescription")] League league)
{
    ViewBag.Title = title;
    if (!ModelState.IsValid)
    {
        if (league.StartDate != null && league.EndDate != null)
        {
            //Make nullable timespan object 't' to establish the span of the two dates.
            TimeSpan? t = league.EndDate - league.StartDate;
            //Assign the timespan converted to days, then converted to a type 'byte', then store this value into Number of weeks.
            league.NumberOfWeeks = Convert.ToByte(Math.Floor(t.Value.TotalDays / 7));
        }
        uow.LeagueRepository.Add(league);
        uow.Save();
        return RedirectToAction("Index");
    }

    ViewBag.LocationID = new SelectList(uow.LocationRepository.Get(), "LocationID", "LocationName");
    return View(league);
}
```


BRENT KUHLMAN

Software/ Web Developer

(913) 972-2233

bekuhlman@outlook.com

linkedin.com/in/bkuhlman

www.kuhlaid.com

EDUCATION

2018 Full-Stack Web Developer Program

Centriq Training

- MVC Framework
- Troubleshooting & Debugging
- Source Control
- Agile/Scrum
- Website Deployment
- Pair Programming
- Code Review

2013 - 2015 College Credits Taken

Johnson County Community College

- Programming Fundamentals
- Game Development
- Concepts/Programming Algorithms with C#
- HTML & CSS

EXPERIENCE

Main Event Entertainment

Olathe, KS

- Communicating with supervisors and peers.
- Maintaining bowling lanes in a clean and professional manner.
- Assisting guests with problems, questions and service appropriately.

Bowl Attendant

Jan 2018
July 2018

United States Postal Service

Springhill, KS

- Operating right hand driven vehicles
- Arranging and delivering mail/packages in delivery sequence.
- Maintain accurate records of deliveries

Rural Carrier Associate

Aug 2016
Sept 2017

Cosentino's Price Chopper

Shawnee, KS

- Stock shelves with products, rotating stock as necessary.
- Deliver customer service in a professional manner.

Grocery & Dairy Clerk

June 2011
July 2016

QUALIFICATIONS

- Solid foundational knowledge of designing and developing full-stack web applications using .NET framework.
- Self-taught in how to build a computer and understand their inner functions.
- Ability to adapt to new situations, critically think, and problem solve.

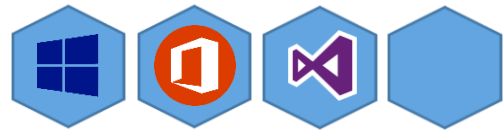


INDEPENDENT DEVELOPMENT PROJECTS

- **Personal Site:** www.kuhlaid.com
- **U Store:** Created a secure application for managing product data. Application is built to simulate an online store front with a shopping cart. Administrators have the ability to manage product, category and vendor data.
- **Final Project:** Created a secure data-driven ASP.NET MVC application from design through deployment for managing the tracking and organization of hardware and software within a company. Administrators have the ability to manage employee, department data and all details relating to assigned hardware and software.

TECHNICAL SKILLS

General



Windows 10 Office Visual Studios

Front End



HTML CSS JavaScript jQuery

Middle Tier



C Sharp .NET ASP.NET

Back End



SQL SQL Server ADO.NET