

System setup and configuration

A database back has been supplied with the filename Spatial2016.bak. Once this database has been restored in SQL Server, the application's connection strings can be changed to allow a connection to the database.

In the Web.config file you will find a connection string property. Use Find and Replace to replace the following:

data source=JAKE\SPATIALD with **data source=*your MS SQL Server Name***

user id=SpatialAdmin with **user id=*the username of a SQL server account that has admin rights***

password=SYSAD with **password=*your SQL Server account password***

Once this has been done the application should be able to access the database. Upon first starting up the app you will be met with a Login screen. You can use either of the following accounts to log in:

Username: SpatialAdmin

Password: bluecatz

Username: SpatialGuest

Password: guest

The SpatialGuest account does not have full access rights to all areas of the application and is merely intended for you to test the user privileges.

System features

In line with the exercise requirements, I opted to create an ASP.NET MVC application in the Visual Studio framework. The following features have been implemented:

User login and authentication:

Users can log in using an account that is matched against a record in the database. Not all users have the same access privileges and only higher level users can access pages that allow them to modify employee data.

Users cannot access the rest of the app without logging in and attempting to log in without an active session will result in the user being returned to the Login page. No registration feature is provided as all registration is handled in the database.

List of all users:

After logging in the user can access a page that lists all active employees who have records in the database.

CRUD operations:

Pages exist for editing, deleting or adding records in the database. These pages are only accessible to users with the right access privileges.

Employee search feature:

A rudimentary search feature has been implemented that allows the user to search for employees based on their name. I wanted to implement multiple search options, one to search by name, one to search by ID and one to search for all employees with a specific job title. Unfortunately time constraints prevented me from achieving this goal, but I would like to elaborate on how I intended to implement this feature. I intended to make use of a Html Radiobox which would pass its selected value to the appropriate ActionResult function in the relevant controller and use if statements to determine which database entries to return to the view in the return statement.

Relationships between employees:

From the employee details page there is an ActionLink that takes you to a page that shows a specific employee's subordinates.

Proposed/intended future features:

The ability to search for employees based on more criteria.

The ability to see the name and employee details of an employee's superior. I unfortunately realised too late that I should have used a self-referencing table for this and did not have time to correct it.

A logout feature. While this is a simple case of running an ActionResult function that clears the session[] variables and returns the user to either a logged out page or the login page, I did not consider it a very high priority and did not get to it in the end.