Programming III .NET Coursework

DanSearch, Daniel Carnovale – K1336511

Introduction

We were set the task of constructing an Event Management system of sorts for a theoretical research group. We were introduced to the Microsoft Visual Studio application, convenient for us due to the amount of code it generates specifically for these types of projects (and many more). It was required that we modify this generated code to allow our application to manage Customers, Events etc, provide a basic level of Admin functionality and some other small customisations like the Contact and About pages.

I have produced the best version of this specification I could manage that introduces the concept of managing customers which would essentially be duplicated for the other tables / functions. I also included a logo, provided basic Contact and About information and tested the register / login scenario. Little tweaks like linking the Customers link to the actual Customers page and so on were also implemented.

In addition I have delivered the database in the following file:

DanSearch\App_Data\aspnet-DanSearch-20170328073235.mdf

To run my application, all that is necessary is extraction of the zip file I submitted, the loading of the project file included in the extracted folder in Microsoft Visual Studio and clicking the green “play button” in the top toolbar. It should start building the application and then open a new browser tab connecting to the localhost address. You can log into the admin account using:

Username: daniel.carnovale@gmail.com

Password: 93Sniper!

Requirements Analysis

DanSearch is an activity management company which deals with multiple clients and arranges multiple events. On any given day staff need information about existing customers, some customers have never attended any events yet but are still known to the organisation. Other customers have a history of attendance to previous events and have booked to attend future events.

Events can be scheduled well into the future and have no customers until registrations are received. Staff need to be able to introduce new events and accept registration requests, which enrol a known customer to a known event. On some occasions a planned event is cancelled and needs to be removed, also some customers are one-time-only customers, and need to be removed from the customer list provided they did not attend any events.

The requirement is to build a public-facing website that reports events for activities customers are participating in, and allows anyone to browse information about these events. The same website should support a login feature which allows internal staff to perform the administration functions described above.

Database Design

The primary entities implied by the requirements are:

- Customer
- Event

In addition the registration of a customer at an event needs to be handled by an intersection table named “Attendance”. Cardinality rules are as follows:

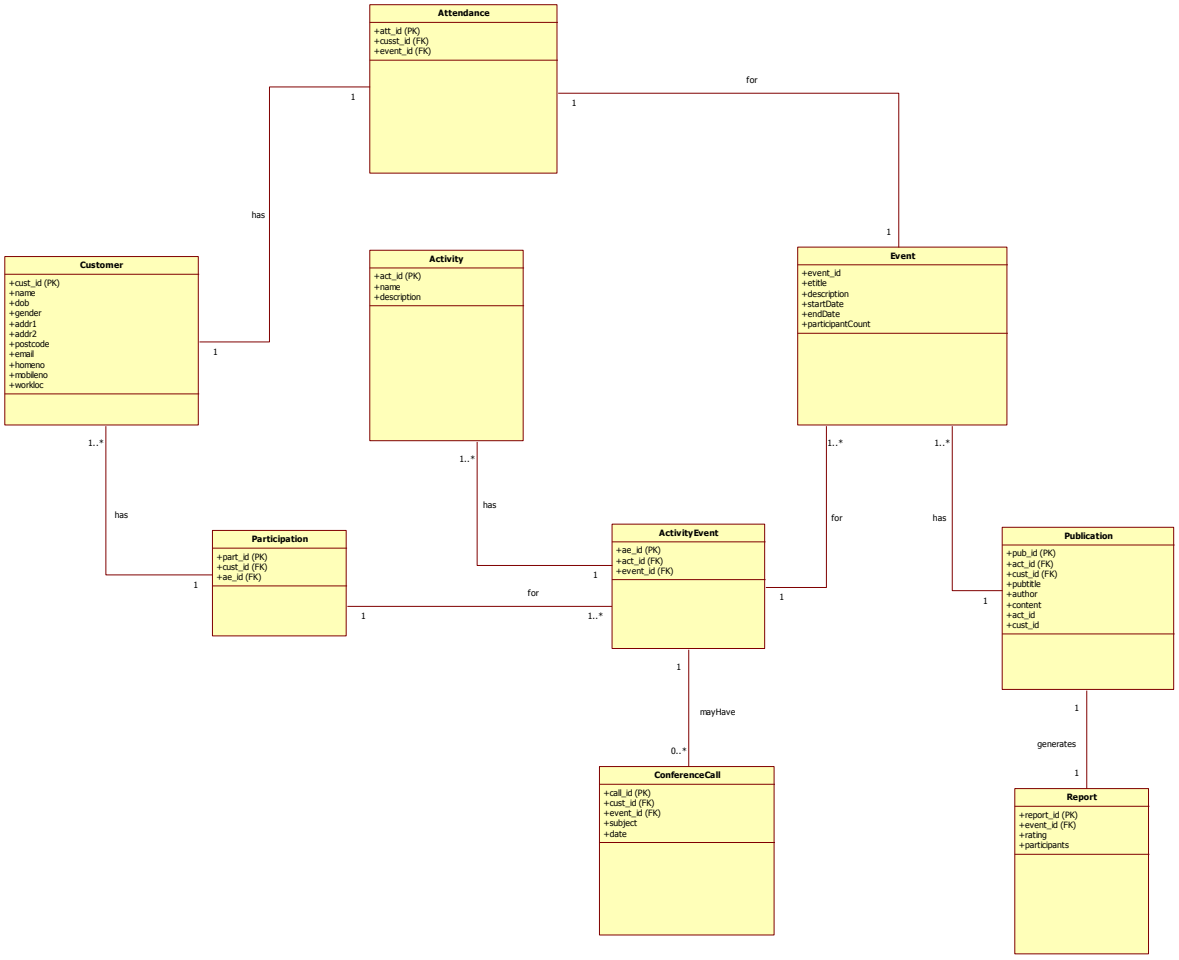
- A Customer may have zero or more attendances
- An Event may have zero or more customers
- An attendance record must have one and only one Customer
- An attendance record must have one and only one Event

To support the development of the application, these three tables will be created using the standalone version of SQL Server that is tied with Microsoft Visual Studio. The web application must support these integrity rules but they should also be enforced via the referential integrity rules present within the database.

Additional tables to support more comprehensive functionality include:

- Activity
- Call
- Report
- Publication
- Attendance

I produced the following UML diagram to support an initial database design:



Web Design

I envisioned my application to start off with a simple Home page, showing activities or events available for potential customers to participate in, alongside register / log in links. Clicking an event or the register link will guide them to creating an account, through which the authentication of would bring them back to the Home page and links to participate in these new events or view events they are currently participating in would be shown. Currently a non-logged in user will only see the "Blardy McBlah" entry introduced through the code, simply as a test of previewing data on this Home page and a demonstration of the styling to be used.

If the customer is an admin however, they are not interested in their own event participation status but rather that of the others, the creation of new events etc. Currently for anyone logged in (though the goal was to be solely for admins) the "Customers" link becomes visible, providing the admin with Edit, Details, Delete and Create functionalities for the customers. Similarly the "Events" and "Announcements" links are visible. Their functionality is missing though I aimed to have them link to modified versions of that of Customers.

The default header, footer, about and contact pages have been modified to be suitable for this project instead. The contact page lists basic contact information with stub links to Chair and Co-chair sites. A logo was also created by myself in Photoshop, uploaded to Amazon S3 (Simple Storage Service), converted to a short url on tinyurl.com and then linked to this within image tags in the index.cshtml file in the header section. This ensured it would be displayed on any page, as it is an absolute link rather than a relative one.

All coding changes, the UML Diagram and this report as they progressed were managed with GitHub, being committed regularly to a separate repository as a backup and source-control strategy.

URL: https://github.com/darkquake93/VisualStudio/

Summary

Issues

Unfortunately I had serious difficulty regards binding in-memory objects to the persistent database (the data seemed to only be dynamic; it vanished on restart of the application) however I found later on in the available time that this was due to an inconsistency in the low-level web.config file which is not normally visible to the standard Visual Studio user. My project and Database connection names being DanSearch and DanDB instead of its initial values, hence issues such as these slowed down production and in the end I could not complete all functionality.

Another issue is that as is often the case with database design, during coding I became aware of improvements that could be made to the data model. I can see from the MVC design structure that it would have been possible to implement these data design changes without major disruption to the web application.

Because of these issues the code I have finally delivered is an earlier version which does not attempt the full database design.

Conclusion

I've noticed that in the Visual Studio .NET MVC environment the approach to including different languages of code within another (for example HTML code within C-Sharp) is quite different to that of embedding the code in other languages such as PHP. I can see that these changes encourage the use of better MVC techniques.

Although I was not able to live up to the final application providing full functionality, I was pleased with the small insight I acquired in creating a new .NET application and attempting to link the different aspects together. I'm pleased however with the stage I left it at and have no doubt a future developer could pick up from where I left off as most of the code is slightly modified generated code, plus a few model, view etc additions yet these don't clutter existing code or confuse the coder. I hope to refer to this application myself in the future as it is already a great head-start in a way to making a decent web application using the MVC environment with Visual Studio.