Blake Matthews  
Capstone Project Proposal

The application I have planned is a ticketing system for a hypothetical computer repair company with multiple physical locations. Through work, I am familiar with tech support ticketing software and the business rules thereof. The other reason I chose this as a project is because making a full-stack CRUD application (without the hypothetical business this is all the software actually is) seemed interesting to me and I thought I would learn a lot.

The business will offer different tech support services as listed in the table below:

A screenshot of a white screen

Description automatically generated

The application will be open enough to facilitate all kinds of devices being serviced from personal computers to a business’s wifi router. Due to inherent limitations, remote consultations could not be performed for physical or hardware repairs, and the effectiveness of remote appointments for software issues would depend on what the issue is e.g. if someone has a corrupted operating system on their computer, no installation media, and is not good at maintaining computers, they would likely benefit from on-site or in-shop work.

\* \* \*

As stated above, the application is CRUD: there is a database where all the data lives, the program logic layer with an API for the front-end, a web front-end for the technicians employed by the company, and a web front-end for customers (typically I would use the word clients but given the technical usage of the word I do not want to confuse).

I am writing the program in C#. For the database I am using SQLite. I intend to use entity framework but there is a chance I may change that when I actually start working on the project. I am picking EF mostly because we talked about it in class and staying within the .net environment will make things go smoothly. I plan to use Blazor for front-end development, again because it keeps my project within the .net environment and we talked about it in class. It took a lot of deliberation to decide this as I researched Blazor and started to seriously wonder whether investing time into learning it would be worth it, but my research indicated that Blazor would be a fine fit for a CRUD app, especially one using all .net otherwise.

A diagram of a diagram

Description automatically generated with medium confidence

*Application Map*

A diagram of a diagram

Description automatically generated with medium confidence

*DB Map*

A screenshot of a computer screen

Description automatically generated

*DB Schema*