Brian Baecher
CS-499
9/7/2025

Self-Introduction

How long have you been in the Computer Science program?

I think approaching 3 years at this point. Truth be told I wish I never signed up.

What have you learned while in the program? List three of the most important concepts or skills.

Most of my meaningful progress has come from self-directed study rather than coursework. I'm pretty dissatisfied with this program, nearing the end and still not feeling as though I've learned much that I couldn't elsewhere (for far less money, or even for free). To answer the question, I think the three most important things I've done are:

Studied C# programming patterns

I've spent a good amount of time trying to solidify my understanding of things like LINQ, delegate types, partial classes, event system, reflection (which seems best avoided if possible), generics, and so on. I really enjoy using the language, and the tools can be used to build any number of things.

Priority-based behavioral system design

Probably the most complicated programming I've ever done, designed game-AI (not machine learning) system with priorities, and contextual decision-making, including custom priority queues and various bits of state management. It's unfortunately still incomplete, but the base structure exists, and should be straightforward to extend.

Fullstack web application development with .NET

I've become familiar Blazor frontends, ASP.NET APIs, and both SQL and NoSQL database integration. I'm hoping to develop using this approach here in this term's project.

While the program itself provided some brief and superficial exposure to some areas related to what I've learned, the depth and challenges came exclusively from my own initiatives, really makes me wonder what it is I paid for.

Discuss the specific skills you aim to demonstrate through your enhancements to reach each of the course outcomes.

I would like to recreate and improve CS-465's fullstack web application assignment to demonstrate my knowledge of .NET, software design, algorithms, and database management. Using Blazor WebAssembly, ASP.NET APIs, and MongoDB, I think I'd be able to create a more sophisticated version of the project.

How do the specific skills you will demonstrate align with your career plans?

I've largely given up on the idea of working in this field, but when I was still laboring under that delusion I was hoping to get into fullstack development, maybe with a particular focus on the backend side of things. That's the reason I'm hoping to revisit CS-465, which I think was a half-decent introduction to the idea of fullstack development, but ultimately I found it lacking in depth.

screenshot of github page.

Enhancement Plan

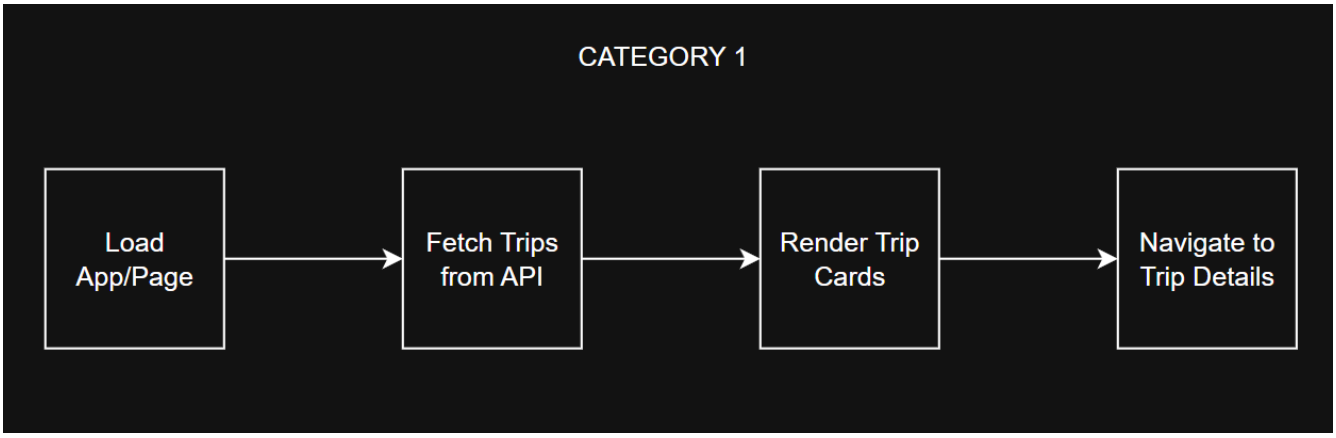
My plan coming into this class was to revisit the fullstack web application we created in CS-465. I've been eager for the opportunity to use my preferred language and environment at some point in this program, and I'm hoping that this is the opportunity – but I suppose you'll let me know.

What I was hoping to do, is to re-create, and improve upon/expand the features in the "Travlr Getaways" web application. The original assignment used the MEAN stack, but I am hoping to recreate it using the tools provided by .NET, with the site itself running Blazor WebAssembly, the backend being controlled via an ASP API. I have come to prefer using NoSQL databses like Mongo, so I could keep that aspect of the original assignment. You can let me know if this is an appropriate choice of project, as instead of choosing one or two 'artifacts' to attempt to improve upon I'd rather recreate the project in its entirety. Ultimately the final product we created in CS-465 was one exceedingly simple Angular page, and a set of html pages that were already provided for us in the initial project files, so not exactly a challenge...

For all 'artifacts' their origin would be from within the "Travlr Getaways" CS-465 course project. It would all be getting rewritten in C#. There may be opportunities to re-use existing JS/TS from the project via Blazor's interoperability capabilities, though.

Category One: Software Engineering and Design

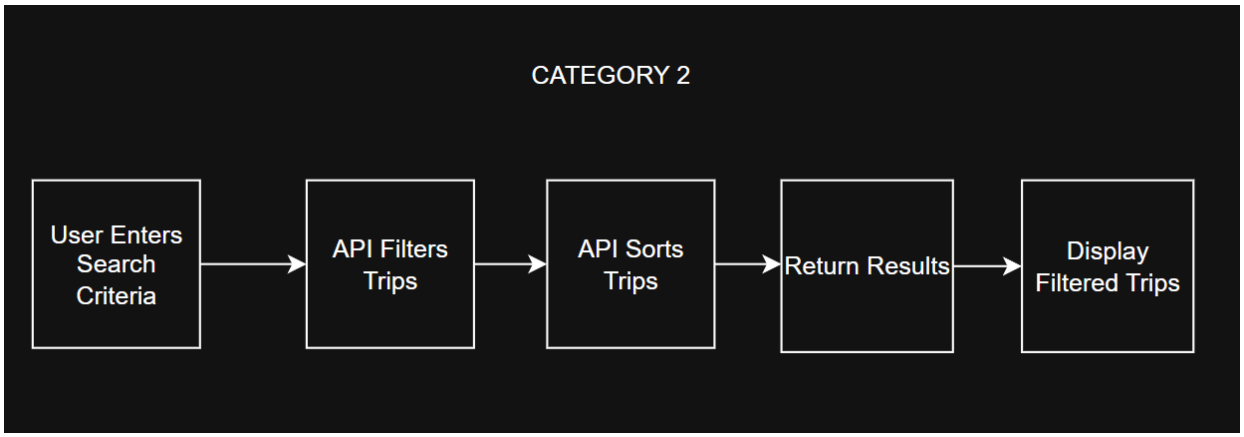
> Redesign the site using Blazor WebAssembly with improved component structure and functionality. Translate the reusable UI Angular components for trips and navigation into Blazor components. Leverage .NET specific frameworks/packages like Radzen to allow for data visualization and ease of development.



> *Relevant outcome:* Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals.

Category Two: Algorithms and Data Structures

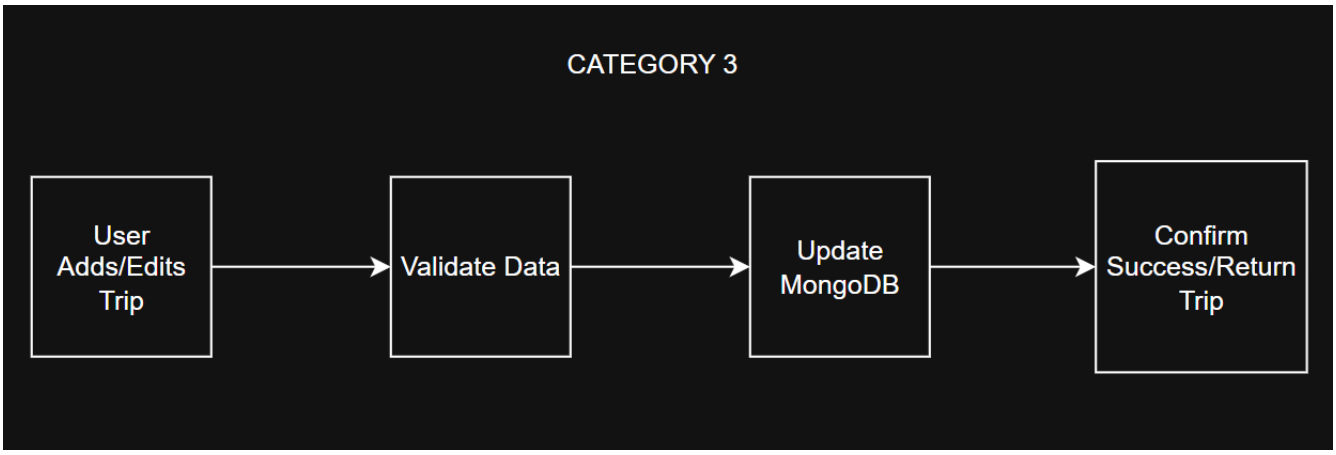
Expand and enhance the searching functionality for browsing through the travel agency's available packages. Originally the search functionality was incredibly barebones, with the only available search criteria being the trip package's identifier string. Expanding the search capabilities would require adding appropriate methods to the API 'trips' controller. Ideas for searching, filtering, and sorting options would be informed by the stored Trip objects, such as the price, distance from user-specified location, popularity, rating, etc.



Relevant outcome: Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices

Category Three: Databases

The redesigned project will continue to use MongoDB for storing necessary information, but some additions and alterations will be made to accommodate new functionalities and ensure the data's integrity. For instance, in the original project, the 'code' field of a trip record did not have its uniqueness enforced. This 'code' string served as the company's internal identifier for trip packages.



Relevant Outcome: Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources

ePortfolio Overall Skill Sets

Skills and outcomes planned to be illustrated in the code review:
Implementation of reusable components, search/filter functionality, I'm sure I'll use LINQ in the API, async/await, separation of concerns.

Skills and outcomes planned to be illustrated in the narratives:
Design decisions, problem-solving, application of software engineering principles.

Skills and outcomes planned to be illustrated in the professional self-assessment:
Examining growth in programming, self-directed learning, how useless this program's been, and ability to apply technical skills/concepts.