CIS174 – Advanced C# Final Project Brainstorming

Chris Bodirsky

**Brainstormed Ideas**

I explored several ideas for this project. First, an ERP or accounting system came to mind—especially one tailored for inventory management in a corner store setting, with modules for tracking sales, stock levels, and basic reporting. Another idea was a photography portfolio and gallery site, where photographers could showcase their best work on a public splash page while offering clients secure access to their own private galleries. I also considered a social media platform similar to Tumblr or Instagram, allowing users to share posts, view content feeds, and manage access restrictions between users and administrators. Ultimately, I decided to pursue the most ambitious and potentially rewarding concept: a logistics dispatch suite.

**Project Details**

This dispatch suite would serve small to medium-sized logistics companies, helping them manage vehicles, delivery jobs, routes, and personnel. The system would log each delivery step, flag unassigned jobs, and use color-coded status indicators to highlight issues. Mobile and API integration would support package scanning and real-time updates. A map interface—powered by Google Maps API or similar—could display delivery locations and route paths. Routing could begin with regional grouping, but a more advanced implementation would include a custom A\* algorithm to suggest optimized delivery sequences for multiple delivery locations and return destination. Drivers would retain the ability to override suggested routes, ensuring flexibility in the field.

**Why Use This App And For Who**

This kind of dispatch software has wide-reaching utility. Courier services can use it to streamline deliveries and reduce missed or delayed jobs. Service industries like handyman or technician dispatch can benefit from route optimization and proximity-based scheduling, allowing workers to prioritize urgent or nearby requests. A payment system could be considered through this software as well, making it an easy way to handle transactions on the spot through mobile device transactions. The system could also be adapted for emergency dispatch or field support not related to deliveries. By creating a modular design for managing logistics, this project can be scaled for businesses that rely on efficient routing and clear communication between dispatchers and drivers but also can optionally lock away or hide features that will not be needed or used for a given client.

**Technical Course Requirements**

Technically, the project satisfies all course requirements. It will feature a multipage layout with navigation, including a dashboard and pages for drivers, vehicles, routes, and deliveries. CRUD operations will manage drivers, vehicles, delivery logs, and route assignments. Razor views will use tag helpers and partials for modular UI components. Session state will track active routes, driver status, and user preferences. Input validation will ensure required data is entered correctly, such as delivery signatures and timestamps. Dependency injection will support modular service layers like route, driver, and delivery services. Unit testing will cover business logic such as route optimization and delivery status updates. Authentication and authorization will distinguish roles (admin, dispatcher, and driver) while the Web API will enable mobile access, package scanning, and external logistics integration.