### Error Handling and Debugging

- Every module will contain exception handling for input validation, database errors and hardware malfunctions (ticket printer down, etc.)
- Logging mechanisms will be used to record system errors for debugging.
- All debugging tools (breakpoints, log tracing, etc.) will be utilized throughout development and testing.

### Collaboration and Integration of coding

- Version Control and collaboration will consist of different branches on GitHub for feature development and a separate branch for bugs and issues.
- We'll maintain code parity using pull requests and code reviews allowing us to maintain high quality of code, and consistency on features while minimizing conflicts.
- We will have weekly integration checkpoints to stay aligned on system parity between front-end, back-end, and database tiers.

### Performance Optimization Plan

- We will take advantage of asynchronous JavaScript (AJAX) with server-side caching to reduce the load on the database for updating things in real-time.
- The initial backend design will utilize efficient queries and connection pooling to help with high traffic.
- Load testing will validate response time as the number of vehicles increases.

### Support and Maintenance Strategy

- The system will include an administrator dashboard to provide an overview of system health and usage data.
- Logs can be monitored on a regular basis to find and address persistent issues as they arise.
- Operator guidance, support documentation, and troubleshooting guides will be prepared.
- End-user feedback will be configured into the system to allow users to report bugs and/or make suggestions.