Lab 1 – Traffic Tamer Descriptive Paper

Emily Reyna

CS 411W

Professor T. Kennedy

07 October 2024

Version 1

Table of Contents

Listing of Figures

Listing of Tables

1 Introduction

Traffic Tamer is a web application designed to address the complexities of traffic laws by offering an intuitive platform for users to search, understand, and navigate traffic laws across different states. The application aims to provide up-to-date, simplified information on traffic regulations, making it accessible for various user groups including general drivers, traffic violators, driver improvement class students, and law students. By leveraging AI and real-time updates, Traffic Tamer will ensure users have the most current and relevant information.

1.1 Societal Problems

Navigating traffic laws across different jurisdictions can be overwhelming due to the varying regulations and legal jargon. Approximately half of the cases in U.S. state courts pertain to traffic violations, and many drivers find it difficult to comprehend the intricacies of the laws they encounter. Traffic Tamer aims to resolve this by simplifying traffic law information.

1.2 Current Traffic Avoidance Methods

Traditional traffic law information sources, such as websites or legal databases, are not easily accessible or understandable to the average user. Traffic Tamer's real-time law lookup feature, supported by AI and machine learning, will enhance access to and understanding of traffic laws, providing detailed explanations in layman's terms.

1.3 Statistics

In 2024, traffic violations accounted for nearly 50% of state court cases, highlighting the need for a solution like Traffic Tamer. Law students and professionals alike could benefit from a resource that simplifies traffic regulations and updates information in real-time.

1.4 Solution Overview

Traffic Tamer addresses these challenges by providing a user-friendly interface that simplifies complex laws, making them accessible to users from different backgrounds. The application automatically looks up traffic laws and generates simplified explanations for users based on their location and violation codes.

2 Traffic Tamer Product Description

Traffic Tamer will be a web-based application designed to simplify the understanding of complex traffic laws, making them more accessible to a wide range of users, including general drivers, traffic violators, driver improvement class students, and law students. The product will leverage AI and machine learning algorithms to provide up-to-date and accurate traffic law information tailored to specific jurisdictions. Key features of Traffic Tamer will include simplified explanations of legal terms, an automated law lookup tool that allows users to quickly find relevant regulations, and categorization of laws by state, county, and violation codes.

The system will be composed of both hardware and software components that work together to deliver real-time updates. The software will be built using React or Angular for the frontend, and Node.js for server-side programming, while the backend will utilize MySQL and SQLite databases. The application will run on any internet-capable device, making it highly accessible. Apache will serve as the web server, and Docker will be employed to ensure seamless deployment across various environments. Traffic Tamer's architecture will enable a smooth user experience, ensuring that users can effortlessly navigate the complex landscape of traffic regulations.

2.1 Key Product Features and Capabilities

The software will analyze current traffic laws, ensuring that it remains updated with any new regulations as they are enacted. It will categorize these laws by state, allowing users to easily find relevant information based on their location. To enhance user understanding, the software will offer simplified explanations of traffic laws, making the legal language more accessible. It will provide the capability to output any requested traffic law, ensuring that users have quick and easy access to the information they need. To better understand how the Traffic Tamer prototype compares to existing solutions,

2.2 Major Components (Hardware/ Software)

The Traffic Tamer system is composed of various hardware and software components that work in harmony to deliver real-time updates and seamless performance. Figure 1 illustrates the MFCD (Multi-Functional Component Diagram), which outlines the key modules and their interactions within the system, including the user interface, server-side processing, and database management.

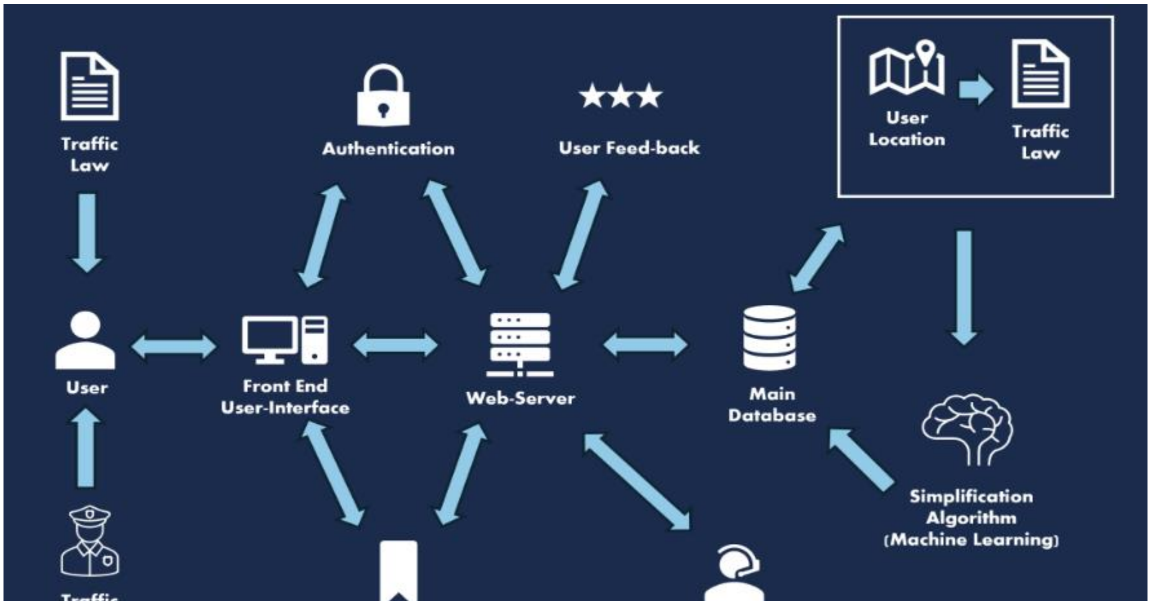
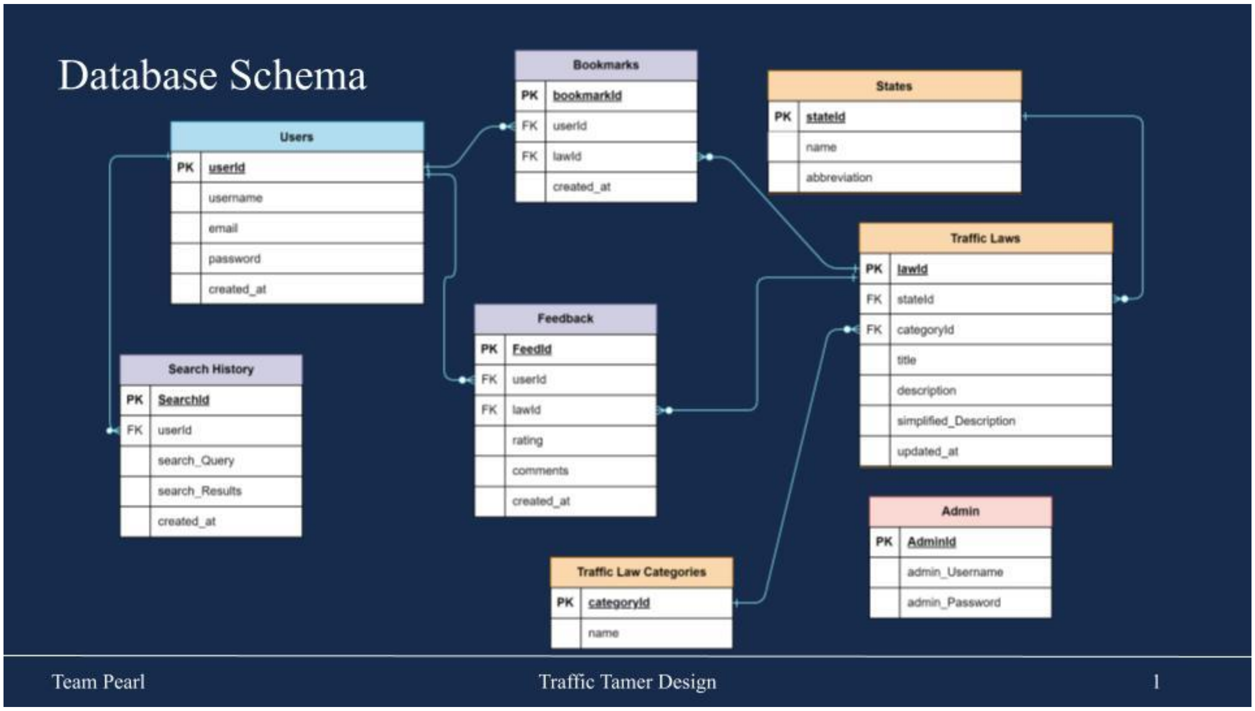


Figure 1: MCFD Diagram

The software will run on a Linux system and use Apache as the web server. It will handle data with MySQL and SQLite databases. Node.js will be used for server-side programming, while HTML, CSS, and JavaScript will be used for the client-side. The front-end will be built with either React or Angular frameworks. Docker will manage the deployment, ensuring the software runs smoothly in different environments. This setup will make the software efficient, easy to maintain, and able to provide up-to-date traffic law information to users.

Docker will be employed to manage deployment, guaranteeing that the application operates smoothly in different environments. This combination of robust software components will provide users with reliable and up-to-date traffic law information.

Figure 2 illustrates the database schema, which plays a pivotal role in the backend of Traffic Tamer. The schema is designed to manage traffic law information, user interactions, and administrative oversight, ensuring seamless retrieval and efficient handling of data. It consists of several interconnected tables. The Users table stores user information such as username, email, and password. The Search History table logs users' search queries and results. The Bookmarks table allows users to save specific laws, with each bookmark linked to a user and a law. The Feedback table records user ratings and comments on traffic laws. The locations table holds information about various locations including their names, abbreviations, & types. The Traffic Laws table contains detailed information about each law, including the associated location and category, along with a simplified description and last update timestamp. The Traffic Law Categories table categorizes laws into distinct types. Lastly, the Admin table manages administrator credentials. The schema is designed to facilitate easy retrieval and management of traffic law information, user interactions, and administrative oversight.



Database Schema

	Users
PK	userId
	username
	email
	password
	created_at

	Bookmarks
PK	bookmarkId
FK	userId
FK	lawId
	created_at

	States
PK	stateId
	name
	abbreviation

	Search History
PK	SearchId
FK	userId
	search_Query
	search_Results
	created_at

	Feedback
PK	FeedId
FK	userId
FK	lawId
	rating
	comments
	created_at

	Traffic Laws
PK	lawId
FK	stateId
FK	categoryId
	title
	description
	simplified_Description
	updated_at

	Traffic Law Categories
PK	categoryId
	name

	Admin
PK	AdminId
	admin_Username
	admin_Password

Team Pearl Traffic Tamer Design 1

Figure 2: Database Schema

3 Product Prototype Description

The Traffic Tamer prototype is designed to simulate the essential functions of the final application. Its primary goal is to validate key features of the product while focusing on essential architectural components. The prototype will serve as a proof of concept to test the core features, functionality, and usability of the Traffic Tamer system before full-scale development.

3.1 Prototype Functional Goals and Objectives

The main objective of the Traffic Tamer prototype is to demonstrate the ability to simplify traffic laws and provide users with accurate, up-to-date legal information. This prototype will focus on validating the core features, including real-time law lookup, location-based law retrieval, and simplified explanations of traffic regulations.

3.2 Prototype Architecture

The prototype architecture consists of several interconnected modules, including a web-based frontend, backend services, and a database infrastructure. The frontend will be developed using React or Angular, while the backend will be powered by Node.js and MySQL or SQLite databases. Docker will be used to manage deployment, ensuring that the prototype functions in different environments smoothly.

Key components of the prototype include:

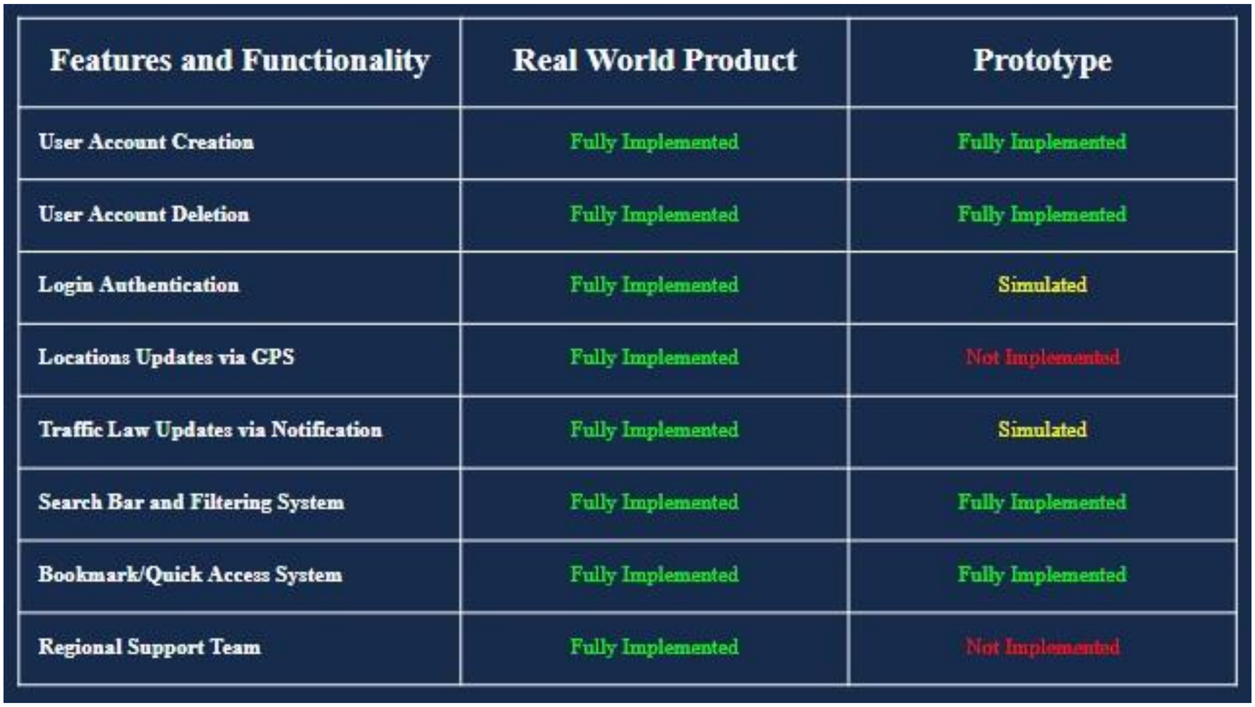
- Web Interface: The prototype will feature a simplified user interface that allows users to easily search for traffic laws based on their location.
- Database: The backend will utilize a relational database to store and categorize traffic laws, user interactions, and search history.
- AI Integration: Machine learning models will help categorize and present simplified versions of traffic laws.
- Law Lookup System: The system will support real-time law lookup using external legal resources.

3.3 Prototype Features and Capabilities

Table 1 offers a comparative analysis between existing real-world product and the Traffic Tamer prototype, showcasing the unique features and benefits that Traffic Tamer introduces to the field of traffic law navigation.

Traffic Tamer prototype will showcase the following features:

- Simplified Law Lookup: Users will be able to search for traffic laws relevant to their location and receive simplified explanations.
- Real-Time Law Updates: The system will showcase its ability to fetch up-to-date legal information from external sources.
- Categorization by Jurisdiction: Traffic laws will be categorized by state, county, and violation type, allowing users to quickly find the information they need.
- Easy-to-understand Explanations: The prototype will simplify legal jargon, making it easier for non-experts to understand traffic laws.



Features and Functionality	Real World Product	Prototype
User Account Creation	Fully Implemented	Fully Implemented
User Account Deletion	Fully Implemented	Fully Implemented
Login Authentication	Fully Implemented	Simulated
Locations Updates via GPS	Fully Implemented	Not Implemented
Traffic Law Updates via Notification	Fully Implemented	Simulated
Search Bar and Filtering System	Fully Implemented	Fully Implemented
Bookmark/Quick Access System	Fully Implemented	Fully Implemented
Regional Support Team	Fully Implemented	Not Implemented

Table 1: Real World Product and Prototype Comparison

By focusing on these key aspects, Traffic Tamer aims to provide a more easily understood approach to navigating traffic laws, addressing gaps left by existing solutions.

4 Glossary

Apache: An open-source web server software that is widely used to serve web content over the internet.

CSS (Cascading Style Sheets): A style sheet language used for describing the presentation of a document written in HTML or XML, defining the look and layout of a web page.

Docker: A platform for developing, shipping, and running applications in containers. Containers are lightweight, portable, and self-sufficient environments that include all the necessary components to run a piece of software.

HTML (HyperText Markup Language): The standard markup language used to create web pages. HTML elements are the building blocks of web pages.

JavaScript: A programming language that enables interactive web pages and is an essential part of web applications. Along with HTML and CSS, it is one of the core technologies of the web.

Linux: An open-source operating system based on UNIX. It is used to run servers, desktops, and mobile devices.

Machine Learning: A branch of artificial intelligence that focuses on building systems that can learn from and make decisions based on data.

MySQL: An open-source relational database management system (RDBMS) that uses SQL (Structured Query Language) to manage and manipulate databases.

Node.js: A JavaScript runtime built on Chrome's V8 JavaScript engine, allowing developers to use JavaScript to write server-side code.

React: A JavaScript library for building user interfaces, particularly single-page applications where data changes over time.

SQLite: A C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine.

State Court: A court that has authority over disputes with some connection to a U.S. state.

Traffic Law: Rules and regulations that govern how vehicles operate on the roads and how road users must behave to ensure safety and order.

Web Application: An application software that runs on a web server and can be accessed through a web browser.

5 References

Aghayari, H., Kalankesh, L. R., Sadeghi-Bazargani, H., & Feizi-Derakhshi, M.-R. (2021, August 2). Mobile applications for Road Traffic Health and safety in the mirror of the Haddon's Matrix - BMC Medical Informatics and decision making. BioMed Central. Retrieved June 11, 2024, from https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-021-01578-8

Online legal services and legal advice. LegalShield. (2024, June 6). Retrieved June 11, 2024, from https://www.legalshield.com/

Traffic attorneys in North Carolina. iTicket.law - Powered by Hatley Law Office. (n.d.). Retrieved June 11, 2024, from https://www.iticket.law/

Traffic Caseload Highlights. (2024, April). National Center for State Courts. Retrieved June 11, 2024, from https://www.courtstatistics.org/__data/assets/pdf_file/0029/99920/Traffic-CLHL.pdf

Web Application Security: Risks, Technologies & Best Practices." CyCognito, Retrieved June 26, 2024, from www.cycognito.com/learn/application-security/web-application-security.php

Odnoletkov, Pavel. "Top 8 Web Application Security Threats and How to Mitigate Them." MBC Managed IT Services, 9 Aug. 2023, www.mbccs.com/web-application-security-threats/

“7 Common Web Security Threats for an Enterprise.” Fortinet, Retrieved June 26, 2024, from www.fortinet.com/resources/cyberglossary/web-security-threats