Semester 1 Capstone Report: Blind Spot Management System (BSMS)
Name: Rohan Kakarlapudi | **UIN:** 678620927
Email: skaka4@uic.edu | **College:** College of Engineering | **Major:** Computer Science
Capstone Title: Blind Spot Management System
Anticipated Completion Date: November 4, 2025 | **Hours Worked per Week:** 5

1. Project Overview

The Blind Spot Management System (BSMS) is an innovative safety and monitoring solution developed to improve parking security and reduce the incidence of accidents caused by limited visibility and blind spots in parking structures. The system integrates ultrasonic sensors, LEDs, buzzers, microcontrollers, and a serial communication protocol to provide real-time feedback to drivers. This feedback mechanism helps ensure safer parking maneuvers by alerting the lot controller to nearby objects and providing parking status updates. The system is modular and scalable, making it a promising prototype for modern urban environments and commercial deployment.

2. Objectives

The primary goal of this semester's work was to develop a working prototype of the BSMS using Arduino-based microcontrollers and ultrasonic sensors. The objectives included enabling reliable serial communication between individual Parking Spot (PS) modules and the BSMS central module, creating a real-time front-end interface to visualize sensor feedback, and designing a system that operates independently using battery power. A major milestone included establishing a communication protocol between devices and integrating the sensor data into a visually accessible and user-friendly dashboard built with React.js and styled using CSS.

3. Technical Implementation

The BSMS hardware includes Arduino Uno R3 microcontrollers programmed in C++ using the Arduino IDE. Ultrasonic distance sensors (HC-SR04) are mounted to detect proximity of vehicles in parking spots, with feedback relayed to the driver through LED indicators (Red for danger, Yellow for booked, Green for clear) and passive buzzers. The modules communicate via the SoftwareSerial library using defined RX and TX pins. A key highlight of the system is the serial data packet exchanged every five seconds, carrying information about each spot's status and the remaining number of free spots. The central BSMS unit parses and processes this data using a line-by-line reader to ensure robustness and synchronization.

Complementing the embedded system is a fully functional front-end interface developed in React.js. The dashboard connects to the serial port through the browser’s Web Serial API and visualizes the real-time parking data in a well-structured layout. CSS was used extensively to enhance the look and feel, using card-style components for each module update, a dynamic safety alert area, and a stylized spatial map with individually labeled parking spots. Spot status is color-coded, and booked spots (like PS3) are highlighted with yellow and annotated with license plate information for added clarity.

4. Serial Communication Protocol

The communication between parking modules and the central unit follows a structured and reliable protocol. Each data packet begins with the keyword "PARKING SPOT UPDATE" and includes information on the number of available spots, followed by individual statuses such as "SPOT 1 STATUS: TOO CLOSE" or "SPOT 3 STATUS: BOOKED." The end of each transmission is marked with an "END" delimiter. This structure allows the BSMS to process real-time updates seamlessly and trigger alerts or display statuses accordingly.

5. Challenges and Solutions

Simultaneous sensor triggers caused serial lag. Adding timeouts to pulseIn() and switching to non-blocking code resolved this. The buzzer initially activated during routine updates; isolating alert conditions from update logic fixed the issue.

6. Testing and Validation

Lab tests verified sensor accuracy and serial communication. Real-world testing in a UIC garage confirmed performance under varied lighting and motion conditions. The system also operated reliably on battery power.

7. Front-End Integration

The React-based dashboard dynamically reflects incoming data and presents spot statuses, alerts, and parking layouts. CSS was used for modular card styling, spot highlighting, color transitions, and spatial map visuals. Booked or dangerous zones appear with distinct visuals, aiding immediate recognition.

8. Business Potential and Future Work

BSMS presents strong potential for commercialization, especially for parking lot operators and city planners. With further refinement, the system can serve as a cost-effective solution to enhance safety and optimize parking efficiency. In the next phase, the hardware will be improved with compact microcontrollers like the FireBeetle Board 32P and more durable enclosures. A backend database using SQL or Firebase will support historical data storage and analytics. Plans also include demo installations to engage stakeholders and explore partnerships for broader adoption.

9. Conclusion

The first semester of BSMS development has resulted in a functional, modular prototype combining sensor-based logic with a dynamic web dashboard. This integrated system provides a foundation for future scalability, backend integration, and real-world deployment. The work completed thus far sets the stage for both technical growth and business exploration in the next phase.

10. References

UIC Honors College Capstone Guidelines
Arduino Uno R3 Documentation
HC-SR04 Ultrasonic Sensor Datasheet
SoftwareSerial Library Reference
React.js Official Documentation
Web Serial API MDN Documentation