Project Workshop

Technische Hochschule Deggendorf

Faculty of Applied Computer Science

Master of Automotive Software Engineering

Mobile applications & interaction design in vehicles

Topic: DriveGuard – Overparking Detection & Safety Assistant

Name:
Deep Bharatbhai Savaliya

Supervisor:
Prof. Dr. Goetz Winterfeldt

Matriculation number:
12501180

Submission date: Deggendorf, 09.07.2025

DriveGuard – Overparking Detection & Safety Assistant

System Overview:

DriveGuard is a proximity-based parking assistant Android application that leverages a TI CC3200 microcontroller paired with an ultrasonic sensor to monitor the rear distance of a vehicle. The system operates entirely over a WiFi Access Point created by the CC3200, requiring no mobile data or internet. The app communicates with the sensor board to retrieve distance values and alerts the driver via visual UI components, sound, and vibration when the vehicle is approaching an obstacle.

It is designed to simulate real parking-assist systems used in modern vehicles and is ideal for both learning and practical applications.

Technologies Used:

- Android + Jetpack Compose
- HTTP Communication (Ktor)
- Media Player, Vibrator
- Canvas Proximity Zones

Primary Users:

- Personal vehicle owners looking for a retrofitted parking assistant
- Commercial drivers (e.g. taxi, delivery) who need better parking accuracy
- Smart car project developers or students in embedded systems courses

Use Case 1: Live Rear Proximity Monitoring

Objective: To provide real-time feedback to the driver based on sensor readings.

Element	Description
Actor	Driver
Trigger	User opens the DriveGuard app while connected to CC3200 WiFi
Input	Distance (in cm) from ultrasonic sensor
Process	App sends HTTP GET request to http://192.168.1.1/sensor repeatedly every 1s
Output	Displays distance, status (SAFE/DANGER), proximity animation
Feedback	- UI with colored triangular zones (green → red) - Sound alert and vibration for < 20 cm - Flashing alert text

❖ **Visual Highlights:**

- Animated car-top image
- 3-layer triangle proximity fan
- Status change triggers alert:
  - Red zone (<10 cm): Buzzer + vibration + alert
  - Yellow zone (10–19 cm): Single warning sound only
  - Green zone (>30 cm): No alerts

Use Case 2: WiFi Connection and Instructions Access

Objective: Help users connect to the CC3200 sensor system for the first time.

Element	Description
Actor	User
Trigger	Taps “info” icon in toolbar
Input	None
Process	App opens a new StepsActivity with instructional content
Output	Step-by-step connection instructions
Feedback	- Scrollable instruction list - Developer photo & name - Supervisor and college logo

Visual Highlights:

- Easy-to-understand steps:
  - Power on CC3200
  - Connect to WiFi (e.g. DistanceMeter)
  - App reads sensor data from 192.168.1.1/sensor
- Displays developer's photo
- Shows academic supervisor and institution logo for authenticity

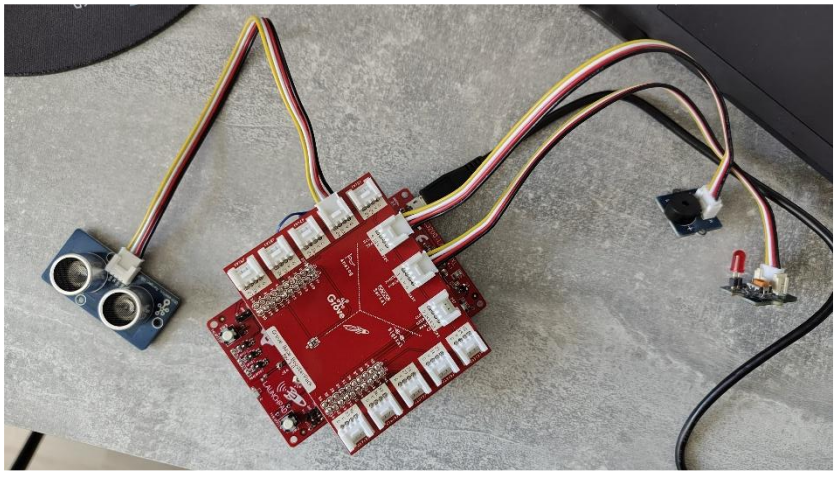
Use Case 3: Yellow Zone Pre-Danger Alert

Objective: Warn the user when an object is close but not yet critically close.

Element	Description
Actor	Driver
Trigger	Distance enters range 10–19 cm
Input	Distance (via sensor reading)
Process	App detects transition into yellow zone
Output	Plays short warning beep once
Feedback	- Orange fan zone - One-time warning tone - No vibration unless it becomes DANGER

Logic Details:

- Uses a temporary MediaPlayer for warning sound
- Sound only plays once per yellow-zone entry
- Sound stops automatically after completion
- Ensures no overlapping with red zone alarm



DriveGuard System Architecture (CC3200 ↔ Android App)

1. Sensor Hardware (CC3200)

- Device: TI CC3200 WiFi-enabled microcontroller
- Connected Sensor: Ultrasonic sensor (e.g., HC-SR04)
- Logic:
  - Measures object distance
  - Runs a local HTTP server in Access Point (AP) mode
  - Serves data via endpoint:
    http://192.168.1.1/sensor
    → JSON format: { "distance": 25 }

2. Communication Layer

- Mode: WiFi Access Point (CC3200 acts as a hotspot)
- Connection:
  - User connects Android phone to DistanceMeter (CC3200's WiFi)
- Protocol: HTTP GET
- Request Interval: 1 request per second from app to CC3200

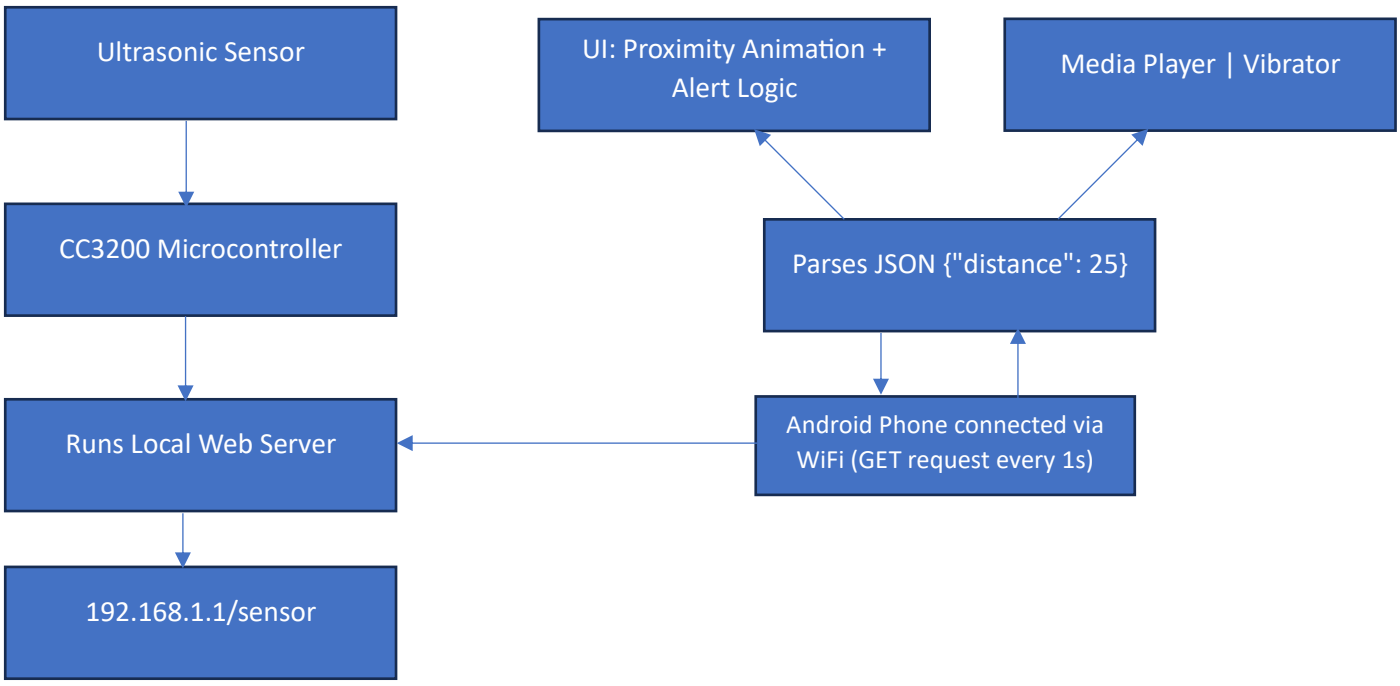
3. Android App (DriveGuard)

- Developed With: Kotlin + Jetpack Compose
- Key Components:
  - Ktor Client: Sends GET requests to fetch distance
  - MediaPlayer: Plays audio alerts
  - Vibrator: Vibrates in danger zones
  - Canvas UI: Animates rear fan zones dynamically (green → red)

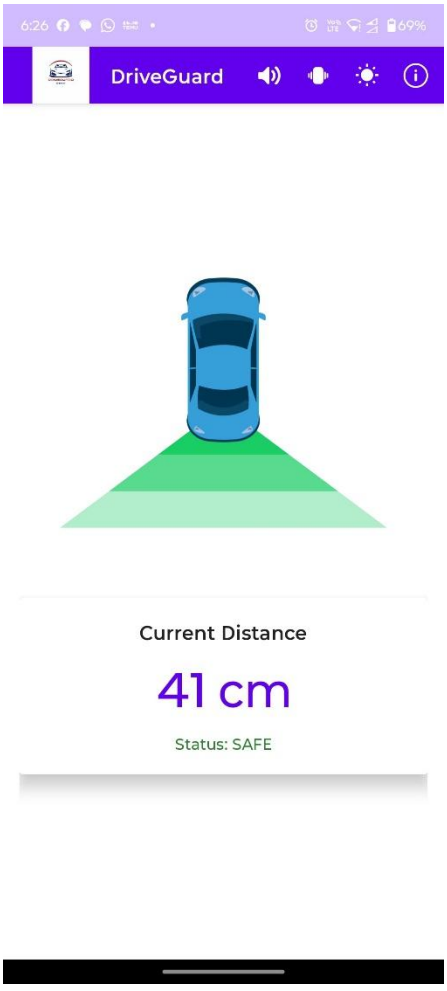
Data Flow Steps:

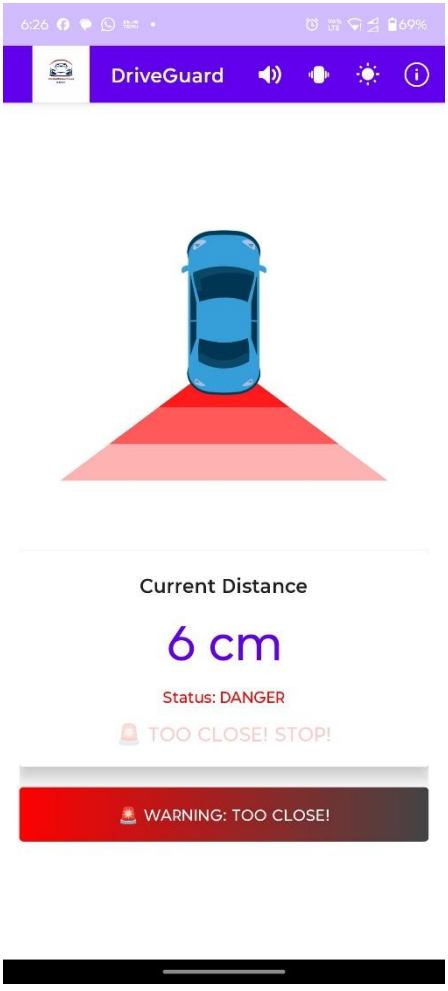
Step	Description
1. Distance Measurement	The ultrasonic sensor connected to the CC3200 detects the object distance in centimeters.
2. Local Web Server	The CC3200 runs in **Access Point Mode**, hosting a local HTTP server at IP 192.168.1.1.
3. Data Formatting	The CC3200 prepares the sensor value as a JSON response: **{ "distance": 25 }**
4. WiFi Connection	The Android phone connects to the CC3200's hotspot (e.g., SSID: DistanceMeter)
5. App Polling (Client)	The DriveGuard app sends **HTTP GET requests** to http://192.168.1.1/sensor every second.
6. JSON Parsing	The app parses the returned JSON and extracts the distance value using **Ktor + kotlinx.serialization**.
7. Real-Time Feedback	Based on distance: - **Green**: Safe - **Yellow**: Warning tone - **Red**: Buzzer + vibration
8. UI Rendering	The app updates the canvas with triangular fan zones behind the car and shows the numeric distance.

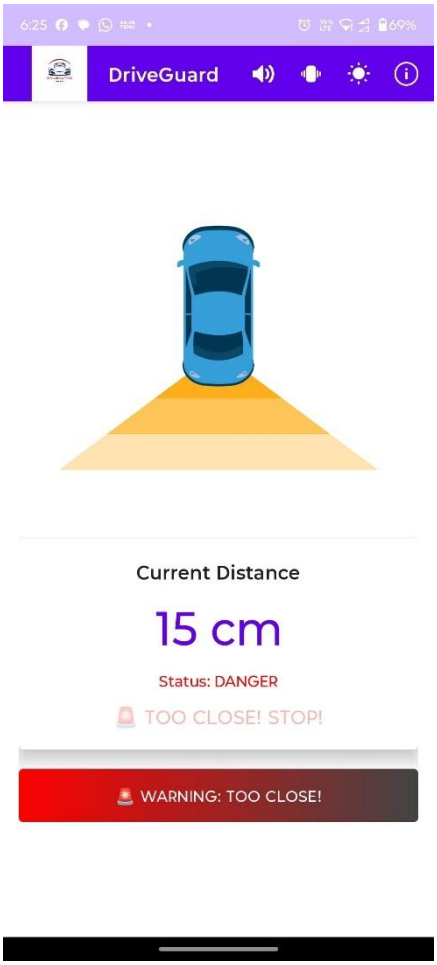
Visual Representation

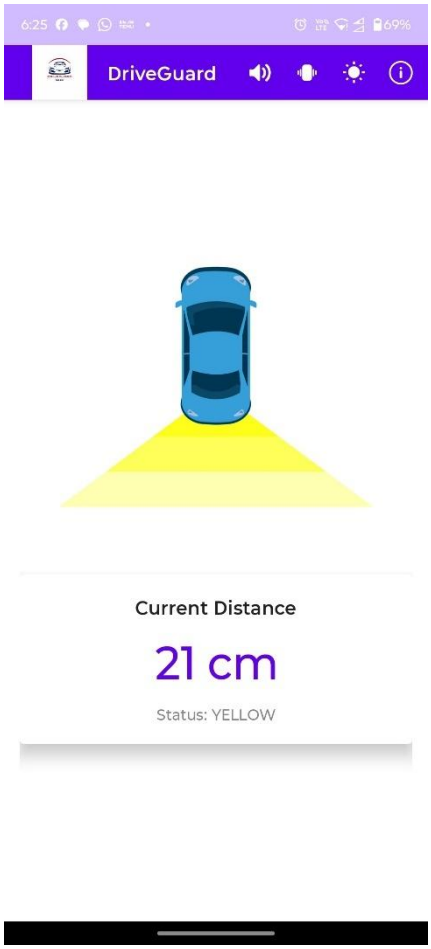


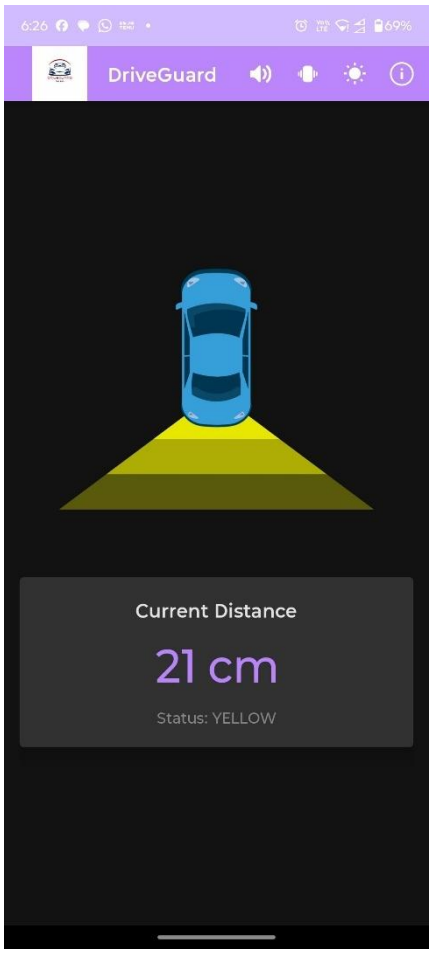
User Interface Screens

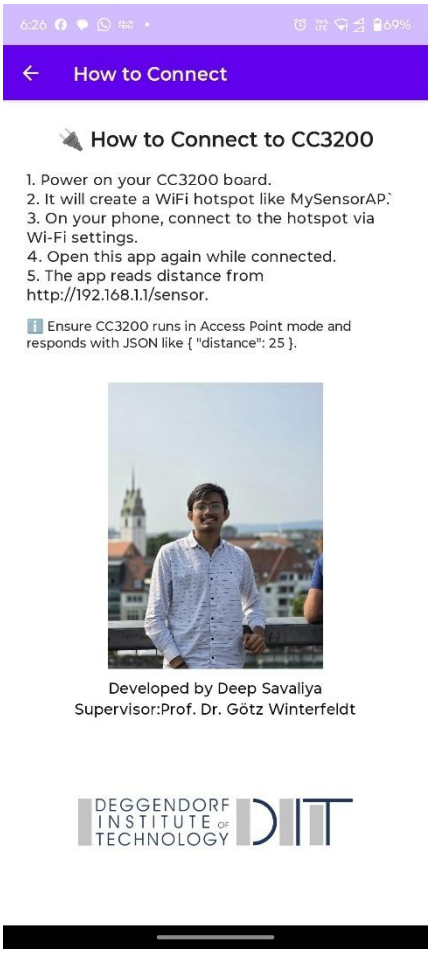





6:25
69%
DriveGuard
Current Distance
15 cm
Status: DANGER
TOO CLOSE! STOP!
WARNING: TOO CLOSE!


6:25
69%
DriveGuard
Current Distance
21 cm
Status: YELLOW


6:26
69%
DriveGuard
Current Distance
21 cm
Status: YELLOW


6:26
69%
How to Connect
How to Connect to CC3200
1. Power on your CC3200 board.
2. It will create a WiFi hotspot like MySensorAP.
3. On your phone, connect to the hotspot via Wi-Fi settings.
4. Open this app again while connected.
5. The app reads distance from http://192.168.1.1/sensor.
Ensure CC3200 runs in Access Point mode and responds with JSON like { "distance": 25 }.
Developed by Deep Savaliya
Supervisor:Prof. Dr. Götz Winterfeldt
DEGGENDORF INSTITUTE OF TECHNOLOGY
DIT