Projekta Tehniskais Specifikācija: Gudrie Atkritumu Savākšanas Sistēma (GASS)

Projekta Tehniskais Specifikācija: **Reāllaika Autostāvvietu Pārvaldības Sistēma (RAPS)**

# 1. Pamatojums

**Problēma:** Pilsētas centrā nepietiek autostāvvietu, kas izraisa satiksmes aizsprostojumus un iedzīvotāju neapmierinātību.  
**Izpēte:**

* Aptauja 200 autovadītājiem: 85% norāda, ka ikdienā tērē vismaz 15 minūtes, meklējot stāvvietu.
* Pašvaldības dati: 40% satiksmes plūsmās veido autovadītāji, kas meklē stāvvietas.  
  **Mērķauditorija:** Autovadītāji, pilsētas satiksmes pārvaldes dienests.  
  **Pieprasījums:** Samazināt stāvvietu meklēšanas laiku par 50%, optimizēt stāvvietu izmantošanu.

# 2. Funkcionālās Prasības

**Nolūks:** Reāllaika stāvvietu pieejamības uzrādīšana mobilajā lietotnē.  
**Datu ievade:**

* Ultrāskaņas sensori (HC-SR04) stāvvietās nosaka aizņemtību.
* GPS modulis fiksē stāvvietas atrašanās vietu.  
  **Apstrāde:**
* Dati tiek pārraidīti uz serveri, izmantojot LoRaWAN.
* Algoritms prognozē stāvvietu pieejamību pēc vēsturiskajiem datiem.  
  **Rezultātu izvade:**
* Lietotāju lietotne rāda stāvvietu statusu (zaļš = brīva, sarkans = aizņemta).
* Paziņojumi par brīvām stāvvietām, kad autovadītājs tuvojas mērķim.

# **3. Lietošanas Gadījumu Diagramma**

Attēls, kurā ir teksts, ekrānuzņēmums, diagramma, rinda

Mākslīgā intelekta ģenerētais saturs var būt nepareizs.

# 4. Lietotāja Interfeisa Skice

**Mobīlā Lietotne:**

* Karte ar stāvvietu ikonām (krāsa atbilst statusam).
* Filtrēšana pēc stāvvietas tipa (invalīdiem, elektroauto).
* Navigācija uz izvēlēto stāvvietu.  
  **Administratora Portāls:**
* Reāllaika datu vizualizācija.
* Statistikas panelis (aizņemtības grafiki, prognozes).

# 5. Projekta Ierobežojumi

* Budžets: 2000 EUR (sensori, mikroshēmas, izstrāde).
* Laiks: 4 mēneši.
* Tehniskie: Sensoru bateriju izturība (1 gads), lietotnes atbalsts iOS/Android.

# 6. Tehniskā Specifikācija

**Sistēmas Arhitektūra:**

* **Sensoru Vienība:** ESP32 + HC-SR04 + LoRa modulis.
* **Komunikācija:** LoRaWAN (enerģijas taupīšana).
* **Backend:** Node.js API, PostgreSQL datu bāze.
* **Mobile App:** React Native (iOS/Android).  
  **Datu Plūsma:**

1. Sensori → LoRa Gateway → Serveris.
2. Serveris apstrādā datus, ģenerē paziņojumus.
3. Lietotāji piekļūst datiem caur lietotni.

# 7. Izstrādes Metodika

**Agile (Scrum):**

* 2-nedēļu sprinti.
* **Sprint 1:** Sensoru datu iegūšanas prototips.
* **Sprint 2:** Backend API izstrāde.
* **Sprint 3:** Lietotnes bāzes funkcionalitāte.
* **Sprint 4:** Integrācija un lietotāju tests.