**🔧 Person A – Mechanik & Sortiermechanismus**

**🎯 Ziel:**

Ein funktionierender Prototyp mit:

* beweglicher Schale
* automatischer Sortierung in 3+ Behälter
* Gehäuse-Prototyp (Holz, 3D-Druck, Karton)

**📋 Aufgaben:**

* CAD-Planung (Fusion 360 oder ähnlich) für Schale & XY-Schlitten
* Bau der **Schale mit Absenk-Mechanismus** (z. B. Linearführung + Schrittmotor)
* Entwicklung eines **Ablasssystems unter Schale** (z. B. Klappe oder Drehteller)
* Entwurf **modularer Halterungen** für Müllbehälter
* Prototyp des **Gehäuses** (Abmessungen an Elektronik & Display anpassen)

**🔗 Schnittstellen:**

* Schale muss **genormt** sein, damit Kamera (Person B) von oben analysieren kann
* Sortierpositionen **genau abstimmen mit Steuerung von Person C**

**🤖 Person B – Kamera & KI-Bilderkennung**

**🎯 Ziel:**

Ein lauffähiges KI-Modell, das **mind. 4 Mülltypen korrekt erkennt**

**📋 Aufgaben:**

* Auswahl der Plattform: **Raspberry Pi + Kamera / Jetson Nano**
* Erstellung eines eigenen kleinen Datensatzes (Fotos von PET, Papier, Bio etc.)
* Training eines **YOLOv5 / TensorFlow Lite**-Modells auf PC oder Google Colab
* Test: Kamera erkennt z. B. „Papier“, gibt Label + Confidence zurück
* Vorbereiten für Integration in System (API oder einfache Datei-Ausgabe)

**🔗 Schnittstellen:**

* Kamera muss zu Schale (von Person A) passen → **Position abstimmen**
* Übergabe: JSON oder einfaches Label an Person C → **Steuersignal generieren**

**📡 Person C – Sensorik, Mikrocontroller & Kommunikation**

**🎯 Ziel:**

Ein ESP32 oder vergleichbarer Mikrocontroller steuert:

* Füllstandssensor
* Gewichtssensor
* NFC-Modul
* WLAN-Verbindung

**📋 Aufgaben:**

* Aufbau eines Testsystems mit:
  + **Ultraschallsensor (z. B. HC-SR04)** für 2+ Müllbehälter
  + **Gewichtssensor (HX711 + Wägezelle)** in der Schale
  + **NFC-Leser (z. B. PN532)** zur Nutzeridentifikation
* ESP32-Programmierung:
  + Messwerte auslesen
  + WLAN-Verbindung herstellen
  + Über MQTT oder HTTP Daten an Server senden
* Steuerlogik für:
  + „Wenn Plastik erkannt & Zielbehälter leer → auskippen bei X,Y“

**🔗 Schnittstellen:**

* Liefert Live-Daten an Person D (per MQTT oder REST-API)
* Muss Steuerung von Mechanik (Person A) ansprechen
* Empfängt Label von Person B → führt dann Aktion aus

**📱 Person D – Weboberfläche, Display, Cloud-Anbindung**

**🎯 Ziel:**

Visualisierung & Reporting von Füllständen + ersten Müllklassifizierungen

**📋 Aufgaben:**

* Aufbau eines **lokalen Dashboards** (React, HTML oder Flutter Web)
  + Anzeige von: Füllstand, Gewicht, letzter Einwurf, Mülltyp
* Anbindung an Backend oder lokalen Server
  + z. B. Firebase oder kleine Node.js-API
* Grafiken (Chart.js oder Recharts): Mülltrends, Entleerungsbedarf etc.
* Aufbau eines einfachen **Cloud-Backends (z. B. Google Sheets / Firebase Realtime DB)**
* Wenn möglich: Erste Version des **KI-Reports als PDF**

**🔗 Schnittstellen:**

* Verarbeitet MQTT/REST-Daten von Person C
* Zeigt Infos aus KI (Person B) auf Display an
* Gibt Rückmeldung an Benutzer: „Müll erkannt: Plastik“