**Steps 3 and 4:**

**Evaluating Alternatives + Engineering Decision**

1. ***Evaluating Alternatives***

**Justification of Logic Solutions Based on Simplicity, Safety, and Reliability**

**Solution 1: Basic Safety-First Logic**

**Why It’s Justified:**

1. **Simplicity** ✅
   * Uses minimal sensors (train + vehicle detection) and straightforward Boolean logic (IF train OR vehicle → lower gates).
   * Easy to implement, debug, and maintain—ideal for low-budget or low-traffic crossings.
2. **Safety** ✅
   * **Fail-safe default:** Gates stay down if sensors malfunction.
   * No delay in reacting to hazards—immediate gate lowering maximizes collision prevention.
3. **Reliability** ✅
   * Fewer components mean fewer failure points.
   * Proven in real-world systems (e.g., classic AHB crossings in the UK).

**Best For:** Rural or low-traffic crossings where cost and simplicity outweigh minor delays.

**Solution 2: Smart Timing & Cross-Verification Logic**

**Why It’s Justified:**

1. **Simplicity**  (Moderate)
   * More complex due to train speed calculations and dual-sensor verification.
   * Still, it uses clear rules (e.g., "lower gates if train is 30 sec away") for predictability.
2. **Safety** ✅ (Enhanced)
   * **Reduced false alarms:** Dual vehicle sensors prevent unnecessary gate drops.
   * **Speed-adaptive timing:** Gates lower precisely when needed, avoiding premature closures.
   * **Safety buffer:** 5-second delay before raising prevents accidents from sensor lag.
3. **Reliability** ✅ (High)
   * Cross-verification reduces sensor errors.
   * Logs emergencies for post-incident analysis, improving long-term reliability.

**Best For:** High-traffic or urban crossings where minimizing delays and false alarms is critical.

***2.Engineering Decision***

* **Choose Solution 1** if the priority is **low cost, simplicity, and ultra-reliability** (e.g., remote areas).
* **Choose Solution 2** if **traffic flow and precision** are critical (e.g., suburban crossings near cities).

-Samir(u3312930)