**Introduction to Computer Engineering:**

**Assignment 1**

**Unit:** Introduction to Computer Engineering

**Unit Code:** 8223 (UG)

**Student ID:** u3216190

**Submission Due:** 20/08/25

**PART 1: On the Engineering Method**

**Step 1: Exploring the Problem**

1. **Problem Restatement**

The goal is to create a safety system that works automatically opens and closes the gates at a railway crossing based on logic. The system needs to lower the gates when a train is coming or when a car is still on the tracks. It should only lift the gates when it is fully safe. The concept is to keep traffic moving smoothly while making it as safe as possible.

1. **Inputs and Outputs**

**Inputs:**

* + Train sensor (detects train approaching or passing)
  + Vehicle sensor (detects if a vehicle is on the tracks)
  + Track-clear sensor (confirms no vehicles remain)

**Outputs:**

* + Gate status (lowered or raised)
  + Warning signals (lights or alarms)

1. **Context, Constraints, Stakeholders**

**Context**: There is a considerable risk of accidents at railway crossings. Introducing a reliable logic system can help prevent possible collisions between trains and other vehicles.

**Constraints:**

1. **Technical:** Sensors must be accurate, gates must respond quickly.
2. **Economic:** Affordable implementation for many crossings.
3. **Social:** Must protect drivers, passengers, and pedestrians.
4. **Environmental:** Energy-efficient and weather-resilient
5. **Legal:** Must comply with road and railway safety regulations.

**Stakeholders**: Railway authorities, train operators, road users (drivers, cyclists, pedestrians),, local government, and emergency services.

**Step 2: Exploring Alternatives**

**Alternative 1:**

Use a train sensor with the vehicle detection system. The gates also lower if a vehicle is detected to be on the tracks. The barriers only rise once the train has passed and no vehicle is on the tracks.

**Alternative 2:**

Implement predictive timing and continually improve the model. The vehicle sensors additionally check whether the tracks are clear before opening the gates.

* **Real-World Example:**

In Australia, most railway level crossings apply a system of track circuits and/or axle counters for train detection (RISSB, 2020). The system engages the flashing lights and warning bells before lowering the gates. The systems also incorporate sensors that detect the presence of a vehicle or pedestrian to ensure the gates do not lift prematurely.

Some pedestrian crossings are equipped with CCTV cameras to improve safety and monitor activity. Fail-safe design principles ensure that gates remain closed in the event of a system failure. The method is consistent with safety regulations in the railway industry, which always come first.

**Steps 3 and 4: Evaluating Alternatives + Engineering Decision**

* **Evaluation:**

***Alternative 1*** is simple, reliable, and ensures gates are always lowered if a hazard is present.

***Alternative 2*** adds predictive timing but increases complexity and the chance of errors.

* **Engineering Decision**

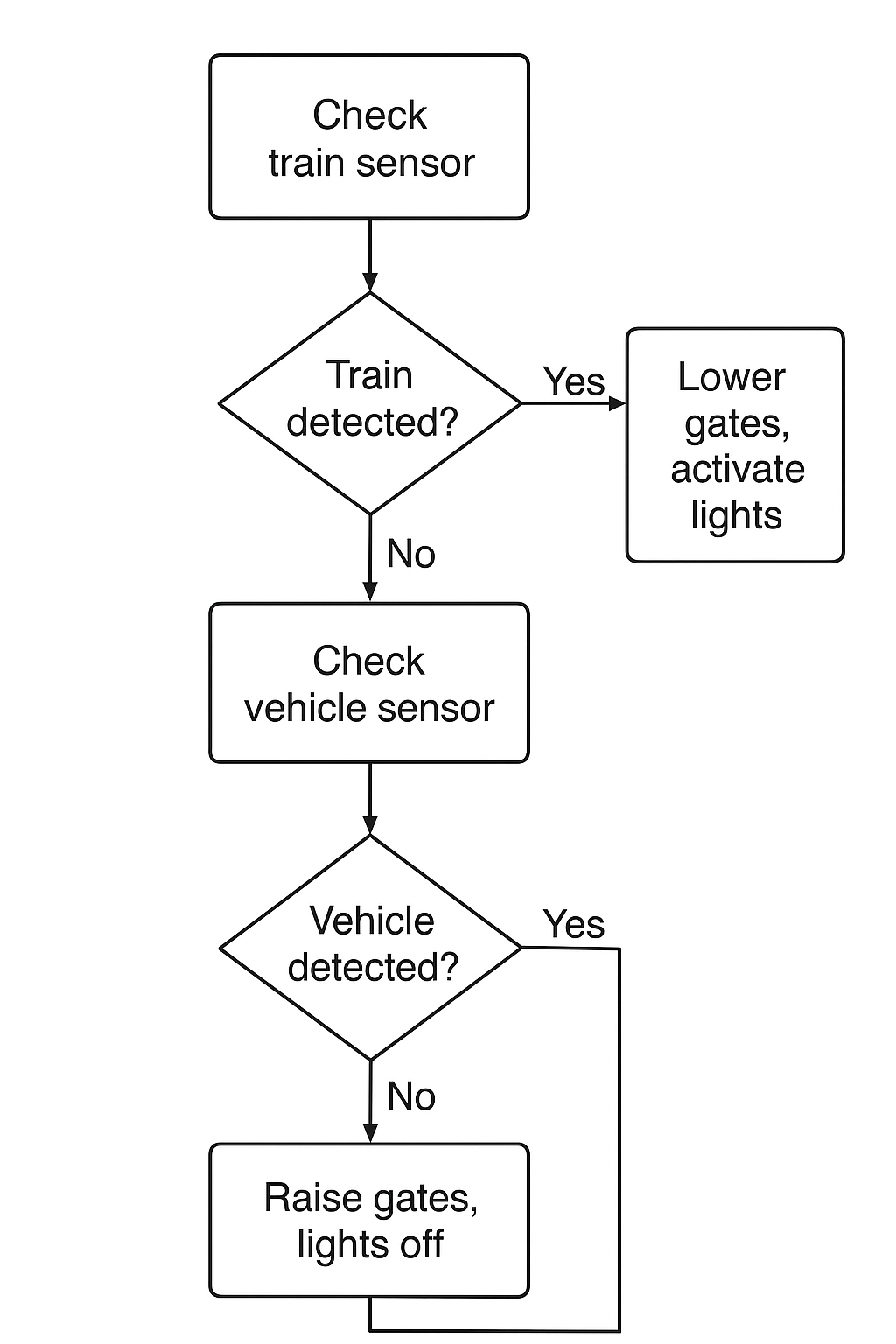
I will use **Alternative 1** as it is safer, reliable and easier to implement. It directly addresses the two main risks: an approaching train and a vehicle stuck on the tracks. Its simplicity reduces the risk of malfunction and increases reliability.

**Step 5 and 6: Planning and Implementing**

**Plain-English Algorithm:**

1. Start system.
2. Continuously check train sensor and vehicle sensor.
3. **IF** train is approaching **OR** vehicle is detected on tracks → LOWER gates and activate warning lights.
4. WHILE train is passing → KEEP gates lowered.
5. AFTER train has passed, check vehicle sensor.
6. IF no vehicle is detected → RAISE gates and turn off warning lights.
7. ELSE keep gates lowered until track is clear.
8. Repeat process continuously.

**Flow Chart of the logic:**



**Figure1.0**: Logic Flow Chart

**Step 7: Testing and Refinement**

* **Test Cases based on input combinations:**

| **Test Case** | **Train Sensor** | **Vehicle Sensor** | **Exected Output** | **Actual Output** | **Notes/Improvements** |
| --- | --- | --- | --- | --- | --- |
| 1 | Train approaching | No vehicle | Gates lowered, lights on | Gates lowered, lights on | Works as expected |
| 2 | No Train | Vehicle on track | Gates lowered, lights on | Gates lowered, lights on | Works as expected |
| 3 | Train approaching | Vehicle on track | Gates lowered, lights on | Gates lowered, lights on | Priority safety confirmed |
| 4 | Train Passed | No vehicle | Gates raised, lights off | Gates raised, lights off | Works as expected |

* **Refinements**
* Add an emergency manual override for railway staff.
* Include redundant sensors to prevent false negatives.
* Introduce delay before raising gates to ensure extra safety margin.

**PART 2: Use of Technology (GitHub)**

* **GitHub repository link:** is included.
* Student’s GitHub is well structured

Repository Setup

* Created repository: railway-gate-safety-system
* Added folders:
  + /Step1\_Analysis
  + /Step2\_Alternatives
  + /Step3\_Flowchart
  + /Step4\_WordCode
  + /Step5\_Testing
* Uploaded README.md with project overview.
* Added tutor and lecturer as collaborators.

[**Read.ME**](http://read.me) **file prompts:**

# Railway Gate Safety System

This project demonstrates a simple logic-based safety system for railway level crossings.

## Features

- Lowers gates when a train is approaching or a vehicle is detected on the tracks.

- Keeps gates lowered until the train passes and tracks are clear.

- Prioritises safety through simple, reliable logic.

## Repository Structure

- Step1\_Analysis – Problem restatement, inputs/outputs.

- Step2\_Alternatives – Brainstormed solutions and research.

- Step3\_Flowchart – Flowchart logic (Draw.io PNG/PDF).

- Step4\_WordCode – Plain English algorithm.

- Step5\_Testing – Test cases and refinements.

## Authors

- Student Name: [Your Name]

**PART 3: AI Agent Integration**

**Reflection** To improve my assignment, I used Microsoft Copilot as an AI agent. I asked two main prompts:

* *“Can you review my plain-English algorithm for the railway gate safety system and suggest improvements?”* Response: It suggested adding an emergency override and a short delay before reopening gates to ensure extra safety.
* *“How could this system be implemented in real hardware such as Arduino or Raspberry Pi?”* Response: It explained how sensors (infrared, ultrasonic, or magnetic) could connect to a microcontroller, with relay switches to operate gates and LED lights for warnings.

These interactions gave me useful insights. The first prompt helped me refine the algorithm by making it more robust and safety-oriented. The second expanded my understanding of real-world implementation and how this system could move from theory to practice.

Using Copilot also highlighted limitations: AI suggestions require human review to ensure feasibility and safety. Overall, AI supported my work by offering alternative ideas and improving documentation while reinforcing the importance of human responsibility in safety-critical systems.

**References**

Microsoft. (2023). *Microsoft Copilot* [Accessed on August 17, 2025].<https://www.microsoft.com/en-us/copilot>

Rail Track Association Australia. (2024). *Low-cost railway level crossings* [Conference paper]. RTSA. [[Accessed on August 19, 2025] https://www.rtsa.com.au/wp-content/uploads/2024/06/Low-Cost-Railway-Level-Crossings.pdf](https://www.rtsa.com.au/wp-content/uploads/2024/06/Low-Cost-Railway-Level-Crossings.pdf)

RISSB. (2020). *AS 7651:2020 Axle counters* [[Accessed on August 19, 2025](https://www.rtsa.com.au/wp-content/uploads/2024/06/Low-Cost-Railway-Level-Crossings.pdf)]. <https://www.rissb.com.au/wp-content/uploads/2020/10/AS-7651_2020_Axle-counters_Preview.pdf>