# Requirements Document

**for**

# Hug the Rail IoT

**Version 3.0**

**Prepared by Tristan Kensigner, Grace Mattern, Zachary Schuh, & Christian Szablewski-Paz**

**Stevens Institute of Technology**

**March 21, 2021**

# Table of Contents

**Table of Contents** **i**
**Revision History** **iii**

**1. Introduction** **1**
1.1. Problem 1
1.2. Purpose 1
1.3. Team 1
1.4. Evolving Current Operations 1
1.5. Approach 1
1.6. Timeline 2
**2. Overview** **4**
2.1. Problem Statement 4
2.2. Data Required to Operate 4
2.3. Expected Output 4
2.4. Available Technologies 4
2.5. Systems 5
**3. Requirements** **9**
3.1. Non-functional Requirements 9
3.1.1. Security 9
3.1.2. Performance 9
3.1.3. Reliability 9
3.1.4. Other Non-Functional Requirements 9
3.2. Functional Requirements 10
3.2.1. Standing Objects 10
3.2.2. Moving Objects 10
3.2.3. Gate Crossing 11
3.2.4. Wheel Slippage 11
3.2.5. Weather Conditions 12
3.3. Hardware & Operating System 12
3.3.1. Operating System 13
3.3.2. Processor 13
3.3.3. RAM 13
3.3.4. Graphics Card 13
3.3.5. Storage 13
3.3.6. Sensors 13
**4. Requirements Modeling** **14**
4.1. Use Cases 14
4.2. Use Case Diagram 18
4.3. Class Diagram 19
4.4. CRC Model Index Cards 19
4.5. Activity Diagram 24
4.6. Sequence Diagrams 25
4.7. State Diagrams 29
**5. Software Architecture** **30**
**6. Code**

7. **Test Cases**
8. **Issues**
9. **Cost**

# Revision History

| Name(s) | Date | Reason For Changes | Version |
|---|---|---|---|
| Christian | 2/15 | Create document/begin section 1: Intro | 1.0 |
| Grace | 2/16 | Revise section 1: Intro | 1.0 |
| Zach, Christian, Tristan | 2/16 | Organize section 1: I into paragraphs, add notes | 1.0 |
| Grace, Christian, Zach, Tristan | 2/22 | Add section 2: Overview | 1.0 |
| Christian | 2/23 | Rewrite bullet points into paragraphs | 1.0 |
| Tristan | 2/23 | Add section 10: Cost | 1.0 |
| Grace | 3/1 | Revise/organize entire document, add graphics | 1.0 |
| Tristan | 3/1 | Elaborate section 2.4, add graphics | 1.0 |
| Grace and Christian | 3/2 | Add section 3: Requirements | 1.1 |
| Grace | 3/4 | Revise section 3: Requirements | 1.1 |
| Grace, Tristan, Christian | 3/9 | Update section 3: Requirements with CTO of HTR recommended features | 1.1 |
| Grace | 3/11 | Update timeline, slippage system diagram, section 3: Requirements | 2.0 |
| Grace | 3/16 | Incorporate TA 1.0 notes, add section 4: Requirements Modeling, fix formatting | 2.1 |
| Christian, Grace, Tristan | 3/17 | Add use cases in section 4: Requirements Modeling | 2.1 |
| Christian | 3/20 | Work on model index cards | 2.2 |
| Chrstian, Tristan, Grace, Zach | 3/21 | Finish section 4: Requirements Modeling | 2.3 |

# 1. Introduction

## 1.1 Problem

Trains are a very common and popular form of transportation that require analysis of data derived from the weather conditions, fuel consumption, and time prediction, among others, in order to ensure safe and successful travel. Current trains retrieve data from servers to receive updates that are needed to make calculations for the Locomotive Control System (LCS). One major setback to this system is that trains need to be connected to servers to keep receiving updates and consequently provide safe travel. This leaves the trains vulnerable when a cellular wifi or internet connection is lost. No new data can be retrieved without this connection and thus the safety of the passengers and operator are jeopardized.

## 1.2 Purpose

The Internet of things (IoT) Hug the Rail (HTR) project is a system that will allow the LCS to make decisions locally in absence of cellular and wifi connectivity to back offices. For example, it will provide the operator with information regarding if the train is slipping or if there is an obstruction. Furthermore, the operator will be able to make informed decisions regarding the problems and cautions specified by the IoT HTR thereby maintaining safe operation and avoiding potentially dangerous situations. The overall purpose of this product would be to make HTR safer, less costly, and more efficient.

## 1.3 Team

Our dedicated team of developers are skilled in areas such as problem solving, programming, and project management and are excited to participate in the project. Despite having little experience with embedded systems, GitLab, and IoT systems, we are driven to design a system that allows trains to efficiently operate without a dependence on a connection. We hope to innovate and reimage the future of the locomotive industry. Our team will develop and implement IoT Hug the Rail from February 9th to April 6st of 2021.

## 1.4 Evolving Current Operations

Current operations on trains receive data from the departure station and cloud. During the travel, updated data is sent from servers in range and the operator utilizes the information to control the LCS. Once at the destination, the train, station, and cloud exchange data. The cloud communicates this information over the larger network. While the LCS has evolved over the years as it now can supply lots of information such as heat of the engine, fuel consumption, and generally offers a failsafe-like “software” to prevent the engine from operating outside of predefined safe limits.

## 1.5 Approach

The edge devices will capture data from other HTR locomotives and the environment instead of relying on data from servers. Many sensors will be implemented to provide crucial information that has now become unavailable (due to the lack of connection). Examples include the front sensors and wheel sensors which will pass information to the LCS regarding if there is an obstruction on the railroads and how fast the wheels are spinning (rpm), respectively. The wheel sensor will communicate with the still working GPS (as it provides speed) to decide if the train is slipping. Additionally, a thermometer will measure the temperature and if

weather conditions are correct, then it will inform the operator the train is slipping on ice and that sand should be put on the railroad tracks. An analytic engine on board the train will process information from the IoT HTR and make suggestions that will be displayed on the IoT HTR screen. The product will provide the capability for the operators to receive statuses and download the latest rules for operation from the Fog/Cloud onto the IoT software. Consequently, a rule based system will be created, and such features will help make the appropriate suggestions to the operator.

## 1.6 Timeline*

The timeline of this project will closely follow an agile methodology; throughout the development process changes will likely be made to both requirements and capabilities. Within each iteration of development, the team will work to recognize problems early and adapt to any changes while continuing to work efficiently. These include but are not limited to software design, technical requirements, deliverables, and the timeline itself. We find that this model helps provide a high quality project that is attuned to needs and desires of the consumers and stakeholders while also reducing the failure to analyse risks.

Week 1: 2/2 - **Communication phase** (constantly revisited, understand new desires of stakeholders)
Form teams
Decide roles and responsibilities
Week 2: 2/9 - **Planning phase** (revisited to address issues)
Understand the problem
Create documentation template
Start Section 1: Intro
Research existing systems
Week 3: 2/16 - Review/revise Section 1: Intro
Start Section 2: Overview
Research sensors that can be used to provide information to IoT HTR
Begin Section 10: Cost
Week 4: 2/23 - Familiarize team with GitLab/Git
Upload version 1.0 on GitLab
Research wheel sensors
Research front sensors
Research GPS interaction with wheel sensor
Research thermometer interaction with wheel sensor
Understand how sensors capture data from other HTR trains and surrounding environment
Week 5: 3/2 - **Requirement Analysis** (revisited if old model fails to fit new requirements)
Start Section 3: Requirements
Implement sensors that enable the IoT HTR to make suggestions
Add CTO of HTR features to Section 3
Research gate status sensors
Research hardware and OS for IoT HTR
Week 6: 3/9 - **Requirements Modeling**
Start Section 4: Use Cases
Use analytics engine to process info from the sensors to the LCS
Week 7: 3/16 - **Software Architecture**
Continue Section 4
Enable operator to enter commands to the LCS
Enable operator to receive statuses when connected
Enable operator to download the latest data from Fog/Cloud onto the IoT HTR when connected

Week 8: 3/23 - **Coding**
Week 9: 3/30 - **Testing**
Revise/Review/Analyze
Week 10: 4/1 - Revise/Review/Analyze
Week 11: 4/6 - **Release first iteration**

*Subject to change

# 2. Overview

## 2.1 Problem Statement

Current IoT technologies on trains are focused on live data received from the Central Operation Center Servers via WiFi/Cellular networks. This heavy dependence on WiFi/Cellular connectivity introduces a massive weakness regarding safety when a stable connection is lost. Therefore it is imperative that a reliable IoT system based on local data be developed to allow for the continuation of a safe trip. For a train to operate properly external data must be provided. In the case that wifi is unavailable to provide the necessary data, a backup solution must be put in place.

## 2.2 Data Required to Operate

IoT HTR will need to collect data about the weather conditions (snow, ice, rain, wind speed) as it is essential in recognizing hazardous weather conditions. Additionally this data provides information regarding safe travel speeds and whether to put down sand or halt. Data regarding moving objects (front or back) is especially important if the obstruction is another train or moving object(s). The data will alert the operator to slow down, stop, or take another route.The train will also need information about the crossing gate's status (open or closed). If the gates are closed, the train will either need a way to open them or would need to stop. We assume that the train will still have access to the GPS/satellite. The GPS data can coordinate with the wheel sensor data which will provide data on the wheels' spin rate (rpm). In a sense, these two data will act to verify each other since when they contradict each other it should alert the operator that the train is probably slipping.

## 2.3 Expected Output

The data will be accessible to IoT HTR, from which the operator can read the suggestions from IoT HTR and make the appropriate decisions. Specifically, IoT HTR will interface with sensors to collect data, analyze it, and use a rule-based logic to issue recommendation or action to the operator via the IoT HTR screen. For example, if the front sensor detects an object is too close (will be defined later), it then will detect if it is moving. This information will be available on the IoT and provide a hazard warning to the operator, who can then stop, slow down, etc. the train. Another example is the wheel sensor which will verify its data with the still working GPS. The GPS provides the speed the train is going while the wheel sensor will provide how fast the wheels are rotating (rpm) and thus if one of the data fails to corroborate the other, then a warning will appear on the IoT HTR screen. The thermometer will provide additional information concerning what the likely cause is (i.e. ice, snow, rain, etc.). With both the knowledge that the train is slipping and the likely cause, say ice, the operator can slow down or pour sand on the railroad tracks. This same system of sensors can also notify the operator when slipping has stopped.

## 2.4 Available Technologies

There are a multitude of different sensors available. Thermometers, hygrometers, friction/shear force sensors, anemometer, and tachometers are the main sensors we intend to use to pass weather related information to the LCS and consequently to the operator. A hygrometer measures the humidity, an anemometer measures wind speeds, and a tachometer measures the rpm of a wheel. Another important sensor is a radar/sonar/ultrasonic-like device that can map out the environment surrounding the train for the purpose of warning/alerting the operator of obstructions. Popular technologies include LED time of flight sensors and LiDAR sensors, the latter of which are not uncommon on cars and airplanes. A LiDAR sensor continually

fires off beams of laser light and then measures how long it takes for the light to return to the sensor. In effect, a LiDAR sensor returns the distance of objects hit by the laser beam and thereby creating a 3D visualization of the environment. With continuous firing, it can also present the movement of objects surrounding the train just as it would notify a driver that a pedestrian, curb, or other vehicle is near. The GPS is a versatile technology that provides information about the position of the train. Not only is it highly reliable, but it also integrates well with the previous technologies. When used in conjunction with weather based sensors, it can help authenticate that a train is slipping. Additionally when used in conjunction with obstruction detection sensors, it can provide the appropriate speed to slow down the train.

The sensors for the IoT Hug the Rail Project would be mainly located in three spots. The top of the train will house most of the weather related sensors such as the anemometer, thermometer, and hygrometer. The tachometer and friction sensors will be located near the wheels of the train as these sensors require information from the wheel's rotation and friction. This data will provide the necessary information to IoT HTR about the train's speed, change of speed, and friction force. The front and back of the train will house sensors concerning 3D mapping for obstacle detection. Our IoT system will still have access to the GPS as it does not require a WiFi/cellular connection. The status (open/closed) of cross gates will be handled by object recognition cameras which continuously compare photos taken with a library of reference photos. **Figure 1** provides a general layout of the sensors on HTR trains.

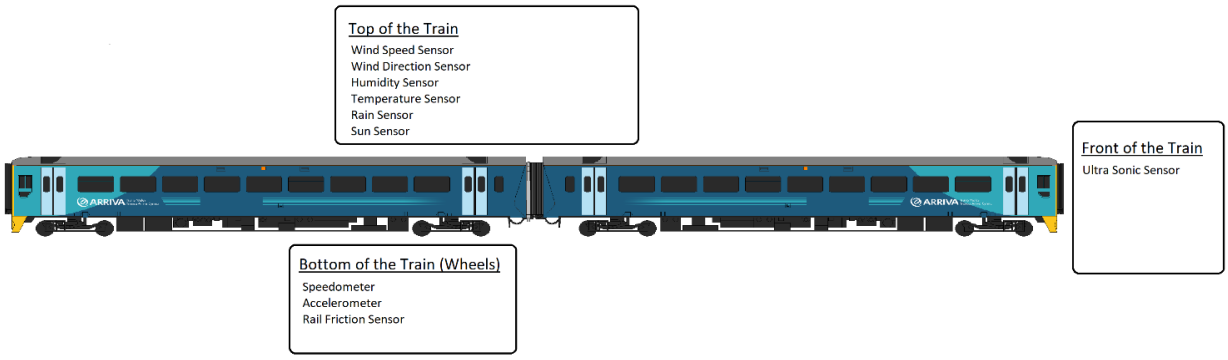



**Figure 1**

## 2.5 Systems

The IoT HTR system will collect data retrieved from the various sensors and interpret them accordingly. Results (i.e. weather, speed, obstructions, etc.) will be available via the IoT HTR screen and suggest operational changes to the operator based on its detections. Below are some conceptual architectures for our IoT. **Figure 2** and **Figure 3** communicate the general problem and solutions. **Figure 4** shows how IoT HTR would be used specifically regarding weather based slippage.

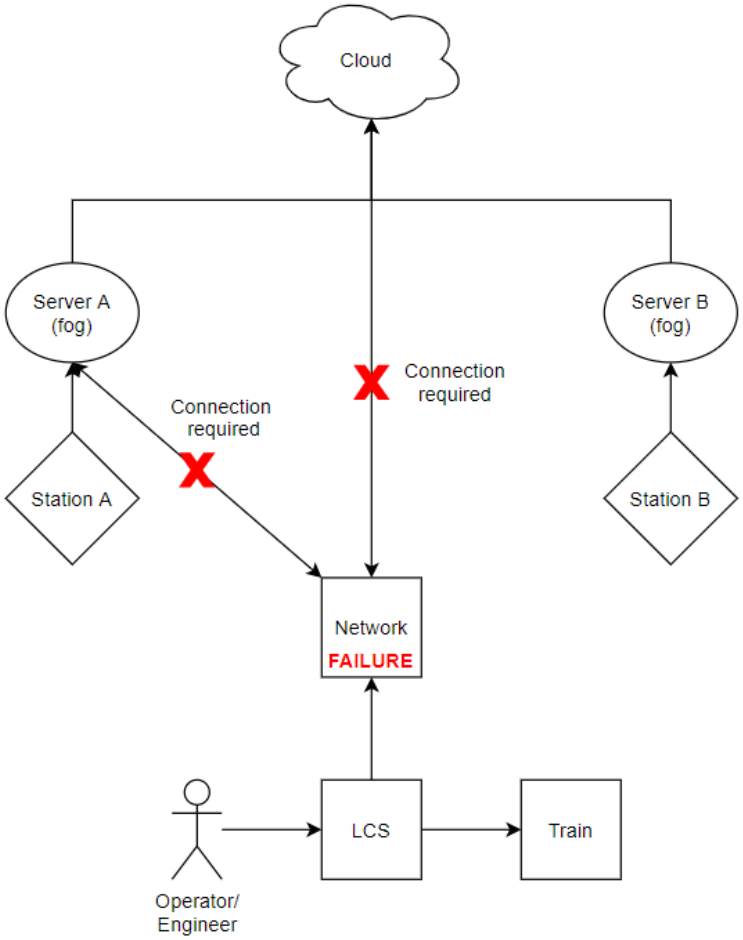



**Figure 2:** *shows the problems that occur if a connection is lost (represented by the red x's). Supposing the train starts in the range of station A, the LCS is connected to the network which receives data from server A and also the cloud. However when the connection is lost no new data is retrieved from server A or the cloud and the network fails.*

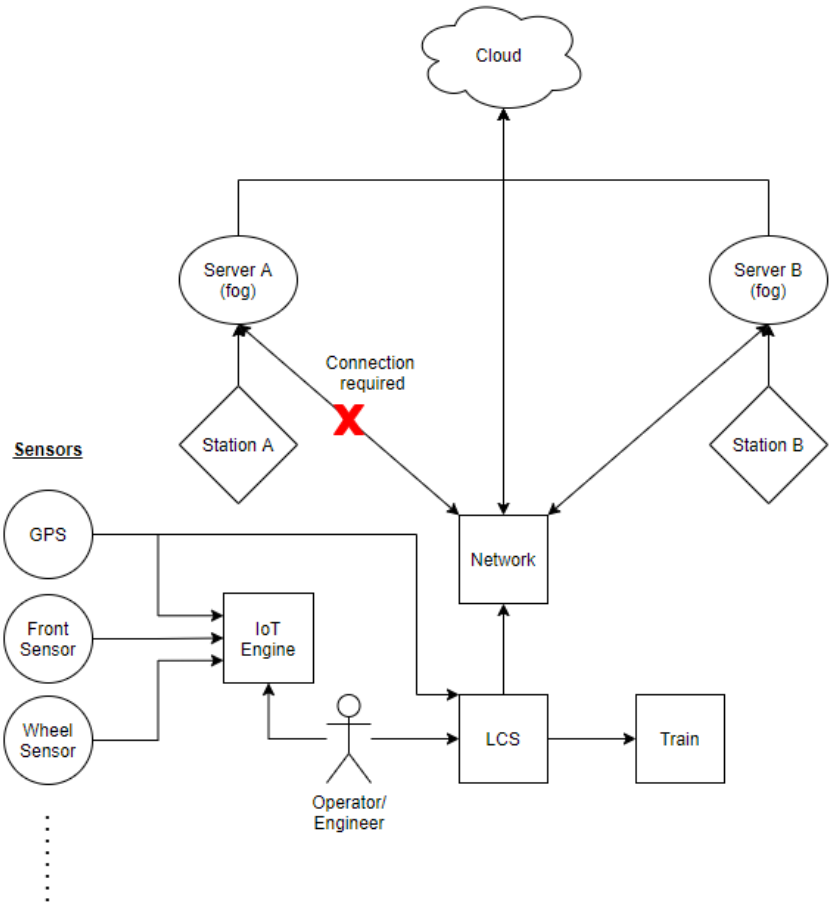



**Figure 3:** *shows the general solution IoT Hug the Rail Project will provide. Supposing the train is headed to station B without a connection, information from the GPS, front sensor, wheel sensor, and other sensors are processed by the IoT HTR which are read by the operator who can then make the appropriate decisions. Once at station B, a connection is reestablished. The train, server B, and the cloud share data.*

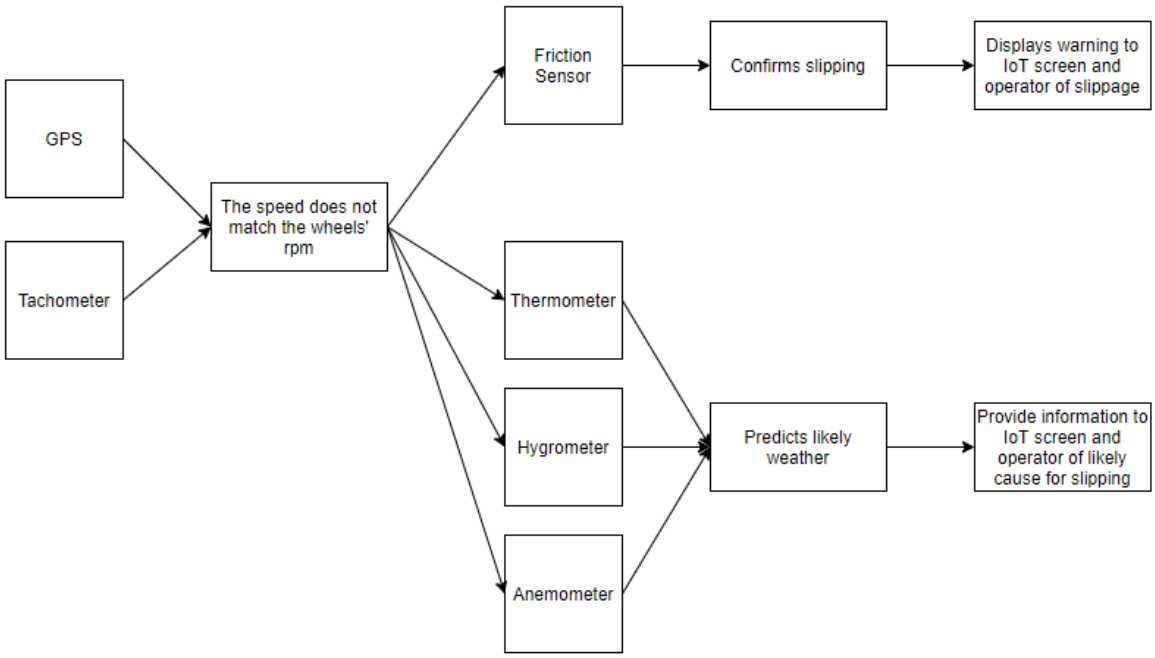



**Figure 4:** *Weather sensors can provide information about ice, snow, rain, slipping, storms, winds, etc. For example the GPS could report the speed of the train but the tachometer would contradict the data by showing that the wheels are rotating faster than the given speed. The thermometer, hygrometer, and anemometer could predict the weather conditions (i.e. snowing, raining, storming, etc.) and the friction/shear force sensor could verify that the train is slipping. The following data will be available to the operator on the IoT HTR screen.*

# 3. Requirements

## 3.1 Non-Functional Requirements

### 3.1.1 Security

R-1: IoT HTR shall only be accessed via user ID and password.
R-2: IoT HTR admin shall have secured (admin ID/Password) to all sensors and equipment.
R-3: Operators shall only see the operations of the train on the IoT HTR display .
R-4: Only admins shall have access to raw data and logs.
R-5: Only admins shall have access to system configurations.
R-6: Only admins shall be able to download/update the software.
R-7: Only admins shall be able to resolve the IoT HTR if it fails.
R-8: IoT HTR network shall be secured by a LoRaWan Protocol.
R-9: IoT HTR shall endode all user-supplied output.
R-10: IoT HTR shall lock the system after 5 failed attempts to login.
R-11: IoT HTR shall unlock after 5 minutes.
R-12: Only operators and admins shall have access to capabilities listed in 3.2.

### 3.1.2 Performance

R-13: IoT HTR sensors shall process an event within 0.5 seconds.
R-14: IoT HTR shall process an event within 0.5 seconds.
R-15: IoT HTR shall process 1,000 events per second without delay.
R-16: IoT HTR shall display processed events within 0.1 seconds.

### 3.1.3 Reliability

R-17: IoT HTR shall operate with no failures 99.99% of the time.
R-18: IoT HTR shall alert the operator how fast obstructions are moving with 95% accuracy.
R-19: The precision of distance-based calculations shall be at least 0.000001.
R-20: IoT HTR defect rate shall be less than 1 failure per 1000 hours of standard operation.
R-21: No more than 1 per 1,000,000 user interactions shall result in a failure requiring IoT HTR to restart.
R-22: IoT HTR shall activate within 100 ms after the power button is pressed.
R-23: IoT HTR shall deactivate within 100 ms after the power button is pressed.
R-24: IoT HTR shall restart within 100 ms after the power button has been pressed for over 5 seconds.
R-25: IoT HTR shall meet or exceed 99.9% uptime.
R-26: If the IoT HTR fails, it shall be identified in 0.01 seconds.
R-27: The network of sensors shall be operational at least 99.99% of the time.

### 3.1.4 Other Non-Functional Requirements

R-28: IoT HTR shall be equipped with a touch screen display.
R-29: IoT HTR shall be equipped with a power button.
R-30: IoT HTR shall be equipped with an auxiliary storage device input.
R-31: IoT HTR shall be able to turn on/off regardless of the status (on/off) of the HTR train.
R-32: Operators and admins shall be able to start and stop the IoT HTR via a power button.
R-33: Only admins shall be able to restart IoT HTR by holding the power button down for over 5

seconds and logging in with Admin ID/password.
R-34: After pressing on the power button, the operator shall be prompted with a login screen.
R-35: The Display shows recommendations from the IoT HTR.
R-36: The Display shows functionalities as stated by the IoT HTR requirements.
R-37: If no issues are presented by all the sensors, then a continue normal operation message shall be displayed.
R-38: In the event of hardware or software failure, IoT HTR shall send a warning message.
R-39: While IoT HTR is initialized, it continuously logs that data and sends it to a single log file to keep travel of all events.
R-40: IoT HTR shall have a clock to attach timestamps to the log file entries.
R-41: IoT HTR shall be able to download data while at a station (accessible only through admin).
R-42: IoT HTR shall be able to upload operational data when at a station (accessible only through admin).

## 3.2 Functional Requirements

### 3.2.1 Standing Objects

R-43: There shall be four cameras, two in front and two in the back of the train.
R-44: There shall be two LiDAR sensors, one in the front and one in the back of the train.
R-45: There shall be two ultrasonic sensors, one in the front and one in the back of the train.
R-46: Cameras shall work with LiDAR sensors to take pictures of the environment.
R-47: LiDAR sensors shall continuously take pictures to map surroundings.
R-48: Ultrasonic sensors shall continuously convert reflected sound into an electrical signal.
R-49: Cameras shall provide obstruction data to the IoT HTR every 0.05 seconds.
R-50: LiDAR sensors shall provide obstruction data to the IoT HTR every 0.05 seconds.
R-51: Ultrasonic sensors shall provide obstruction data to the IoT HTR every 0.05 seconds.
R-52: IoT HTR shall process obstruction data.
R-53: IoT HTR shall analyze for standing obstructions in a 2.0 mile radius of the train.
R-54: IoT HTR shall detect standing objects on the path of the train.
R-55: IoT HTR shall display the distance of standing objects on screen.
R-56: IoT HTR shall display that the object is moving at 0mph (as it is stationary).
R-57: IoT HTR shall display if the standing object is in front or behind the train on screen.
R-58: IoT HTR shall display an on screen warning message if an obstruction is detected.
R-59: IoT HTR shall suggest to the operator to slow down if there is an object within a 2.0 mile radius of the train.
R-60: IoT HTR shall suggest to the operator to brake if there is an object within a 1.0 mile radius of the train.
R-61: IoT HTR shall suggest to the operator to speed up when there is no object within a 2.0 mile radius of the train.

### 3.2.2 Moving Objects

R-62: There shall be four cameras, two in front and two in the back of the train.
R-63: There shall be two LiDAR sensors, one in the front and one in the back of the train.
R-64: There shall be four motion detection sensors distributed equally along the train's length.
R-65: There shall be two ultrasonic sensors, one in the front and one in the back of the train.
R-66: Cameras shall work with LiDAR sensors to take pictures of the environment.
R-67: LiDAR sensors shall continuously take pictures and map surroundings.

R-68: Motion detection sensors shall continuously detect infrared energy (heat).
R-69: Ultrasonic sensors shall continuously convert reflected sound into an electrical signal.
R-70: Cameras shall provide obstruction data to the IoT HTR every 0.05 seconds
R-71: LiDAR sensors shall provide obstruction data to the IoT HTR every 0.05 seconds
R-72: Motion detection sensors shall provide obstruction data to IoT HTR every 0.05 seconds.
R-73: Ultrasonic sensors shall provide obstruction data to the IoT HTR every 0.05 seconds.
R-74: IoT HTR shall process obstruction data.
R-75: IoT HTR shall analyze for obstructions in a 2 mile radius of the train.
R-76: IoT HTR shall detect moving objects (relative to the train) on the path of the train.
R-77: IoT HTR shall detect the speed of the moving object.
R-78: IoT HTR shall display a warning message if obstruction is detected.
R-79: IoT HTR shall display the distance of moving objects on screen.
R-80: IoT HTR shall display if the moving object is behind or in front of the train on screen.
R-81: IoT HTR shall display the speed of the moving object.
R-82: IoT HTR shall suggest to the operator to slow down if there is an object within a 2.0 mile radius of the train.
R-83: IoT HTR shall suggest to the operator to brake if there is an object within a 1.0 mile radius of the train.
R-84: IoT HTR shall suggest to the operator to speed up when there is no object within a 2.0 mile radius of the train.

### 3.2.3 Gate Crossings

A closed gate status means cars should stop at the railroad crossing as a train is approaching.
An opened gate means that cars can drive over the railroad crossing as no train is approaching.

R-85: There shall be one radar sensor located in the front of the train.
R-86: There shall be two ultrasonic sensors, one in the front and one in the back of the train.
R-87: There shall be four cameras, two in the front and two in the back of the train.
R-88: Radar sensors shall continuously emit radio waves and checks the change in frequency after omission.
R-89: Ultrasonic sensors shall continuously convert reflected sound into an electrical signal.
R-90: Cameras shall continuously take pictures to compare with a set of photos of crossing gates (opened and closed).
R-91: Radar sensors shall provide gate status data to the IoT HTR every 0.05 seconds.
R-92: Ultrasonic sensors shall provide gate status data to the IoT HTR every 0.05 seconds.
R-93: Cameras shall provide gate status data to the IoT HTR every 0.05 seconds.
R-94: IoT HTR shall process gate status data.
R-95: IoT HTR shall analyze crossing gate's status.
R-96: IoT HTR shall detect gate crossing's status.
R-97: IoT HTR shall display a warning message if the gate is open on screen.
R-98: IoT HTR shall display the distance of the open gate crossing on screen.
R-99: IoT HTR shall suggest to the operator to slow down if the gate is open within a 3.0 mile radius.
R-100: IoT HTR shall suggest to the operator to brake if the gate is open within a 2.0 mile radius.
R-101: IoT HTR shall suggest to the operator to speed up/maintain current speed when the gate is closed.

### 3.2.4 Wheel Slippage

R-102: IoT HTR is equipped with four tachometers on the four outermost train wheels.
R-103: IoT HTR is equipped with a one GPS.
R-104: Tachometers shall continuously measure the rotation speed of the train wheel in rpm.
R-105: GPS shall continuously send and receive signals to calculate the distance and position of the train.
R-106: Tachometers shall provide wheel slippage data to the IoT HTR every 0.05 seconds.
R-107: GPS shall provide wheel slippage data to the IoT HTR every 0.01 seconds.
R-108: IoT HTR shall process wheel slippage data.
R-109: IoT HTR shall analyze for wheel slippage.
R-110: IoT HTR shall detect if the train's wheels are slipping by verifying wheels' rpm from the tachometer and speed from the GPS .
R-111: IoT HTR shall display a warning message if the train's wheels are slipping.
R-112: IoT HTR shall suggest to the operator to slow down if the train is minorly slipping.
R-113: IoT HTR shall suggest to the operator to brake if the train is slipping severely.
R-114: IoT HTR shall suggest to the operator to speed up with caution if slipping stops.
R-115: IoT HTR shall suggest to the operator to speed up/maintain current speed if no wheel slippage is detected.

### 3.2.5 Weather Conditions

R-116: IoT HTR shall be equipped with one hygrometer.
R-117: IoT HTR shall be equipped with one thermometer.
R-118: IoT HTR shall be equipped with one anemometer.
R-119: IoT HTR shall be equipped with one rain gauge.
R-120: Hygrometer shall continuously measure the humidity.
R-121: Thermometer shall continuously measure the temperature.
R-122: Anemometer shall continuously measure wind speeds.
R-123: Rain gauge shall continuously measure the amount of rainfall.
R-124: Hygrometer shall provide weather data to the IoT HTR every 0.05 seconds.
R-125: Thermometer shall provide weather data to the IoT HTR every 0.05 seconds.
R-126: Anemometer shall provide weather data to the IoT HTR every 0.05 seconds.
R-127: Rain gauge shall provide weather data to the IoT HTR every 0.05 seconds.
R-128: IoT HTR shall process weather data.
R-129: IoT HTR shall analyze for weather conditions.
R-130: IoT HTR shall display likely weather conditions.
R-131: IoT HTR shall display a warning message if there are hazardous weather conditions.
R-132: IoT HTR shall suggest to the operator to slow down if there is mild rain.
R-133: IoT HTR shall suggest to the operator to slow down if there is a mild storm.
R-134: IoT HTR shall suggest to the operator to slow down if there is mild snow.
R-135: IoT HTR shall suggest to the operator to slow down if there is ice.
R-136: IoT HTR shall suggest to the operator to brake if there is heavy rain.
R-137: IoT HTR shall suggest to the operator to brake if there is a heavy storm.
R-138: IoT HTR shall suggest to the operator to brake if there is heavy snow.
R-139: IoT HTR shall suggest the operator to put sand on the rails if ice is present.
R-140: IoT HTR shall suggest to the operator to speed up/maintain current speed if nonhazardous weather conditions are present.

## 3.3 Hardware & Operating System

R-141: IoT HTR hardware shall be able to support at least 1,000 sensors.

R-142: IoT HTR shall support 5TB of data everyday.
R-143: IoT HTR shall be able to import and export information obtained from operating.
R-144: IoT HTR shall be updated yearly with patches in between.

### 3.3.1 Operating System

R-145: Microsoft Windows 10 IoT Enterprise
R-146: Microsoft Windows 10, 64-bit

### 3.3.2 Processor

R-147: ARM processor

### 3.3.3 RAM

R-148: 2GB

### 3.3.4 Graphics Card

R-149: GPU that supports DirectX 9

### 3.3.5 Storage

R-150: 32GB

### 3.3.6 Sensors

R-151: Tachometer - measures the rpm of a wheel
R-152: Hygrometer - measures humidity
R-153: Thermometer - measures temperature
R-154: Anemometer - measures wind speed
R-155: Rain gauge - measures/collects the amount of rain
R-156: LiDAR - maps surrounding environment
R-157: Radar - detect is the status of crossing gate
R-158: Ultrasonic sensor - detects a objects near the crossing gate, train
R-159: Cameras - for gate status and LiDAR
R-160: GPS - give global positioning
R-161: Motion detection sensors - for LiDAR and moving objects

# 4. Requirements Modeling

## 4.1 Use Cases

**Use case 1:** Initialize IoT HTR
**Primary actor:** Operator
**Secondary actors:** Admins
**Goal in context:** Turn on IoT HTR
**Preconditions:** IoT HTR is off
**Trigger:** Operator decides to turn on IoT HTR, that is, to press the power button
**Senario:**

1. Operator presses the power button
2. IoT HTR touch screen display turns on
3. IoT HTR displays login screen
4. Operators types in ID and password
5. IoT HTR displays operations of the HTR train
6. The log begins to record events starting at the time of initialization

**Exceptions:**

1. Incorrect login: Warning message “Incorrect user ID or password” is displayed, operator re-enters correct ID/password
2. Failed to login five times: Warning message “Incorrect user ID or password. Wait 5 minutes” is displayed
3. Fails to start: Display does not turn on, admin is required to fix issue(s)

---

**Use case 2:** Obstruction detection
**Primary actor:** Proximity sensors
**Secondary actors:** IoT HTR
**Goal in context:** Notify operator for any encounters of obstructions
**Preconditions:** IoT HTR is initialized
**Trigger:** Proximity sensors detects that there is a obstruction
**Senario:**

1. Sensor(s) detects that there is an obstruction
    a. Cameras detect an obstruction
    b. LiDAR sensors detect an obstruction
    c. Motion detection sensors detect an obstruction
    d. Ultrasonic sensors detect an obstruction
2. IoT HTR processes obstruction data
3. IoT HTR analyzes if there is an obstruction within a 2 mile radius
4. IoT HTR determines if the obstruction is moving or not
5. If the obstruction is moving the IoT HTR will determine its speed
6. IoT HTR displays if there is an obstruction in front or behind the train
7. IoT HTR  suggests that the operator slows down if there is an obstruction within a 2 mile radius
8. IoT HTR suggests to the operator to brake if there is an object within a 1 mile radius
9. IoT HTR suggests to the operator to continue/speed up if there is no obstruction within a 2 mile radius
10. Logger logs obstructions

**Exceptions:**

1. A broken/faulty sensor
    a. Cameras do not detect an obstruction
    b. LiDAR sensors do not detect an obstruction
    c. Motion detection sensors do not detect an obstruction
    d. Ultrasonic sensors do not detect an obstruction

---

**Use case 3:** Crossing gate status detection
**Primary actor:** Crossing gate sensors
**Secondary actors:** IoT HTR
**Goal in context:** Notify operator of gate status
**Preconditions:** IoT HTR is initialized
**Trigger:** Crossing gate based sensors detect the status of upcoming crossing gates
**Senario:**

1. Sensor(s) detect gate status
    a. Radar sensors retrieve gate status data
    b. Ultrasonic sensors retrieve gate status data
    c. Cameras retrieve gate status data
2. IoT HTR processes gate status data
3. IoT HTR analyzes gate status data
4. IoT HTR displays the status (opened/closed) of the crossing gate on screen
5. IoT HTR displays the distance of the gate on screen
6. IoT HTR displays a warning message if the gate is open on screen
7. IoT HTR suggests to the operator to slow down if the gate is open within a 3 mile radius
8. IoT HTR suggests to the operator to brake if the gate is open within a 2.0 mile radius.
9. IoT HTR suggests to the operator to speed up/maintain current speed when the gate is closed
10. Logger logs gate status

**Exceptions:**

1. A broken/faulty sensor
    a. Cameras do not detect gate status
    b. LiDAR sensors do not detect gate status
    c. Motion detection sensors do not detect gate status
    d. Ultrasonic sensors do not detect gate status

---

**Use case 4:** Wheel slippage detection
**Primary actor:** Wheel sensors
**Secondary actors:** IoT HTR
**Goal in context:** Notify operator for any encounters of wheel slippage
**Preconditions:** IoT HTR is initialized.
**Trigger:** Wheel based sensor detects wheel slippage
**Senario:**

1. Tachometer detects that there is wheel slippage
2. GPS shall provide wheel slippage data to the IoT HTR every 0.01 seconds
3. IoT HTR receives wheel rpm data
4. IoT HTR will process the data given about the wheel slippage by comparing the expected speed given the RPM compared with the actual speed determined by the GPS.
    a. Minor slippage would be a difference 1 and 20 mph
    b. Major slippage would be any difference above 20mph

5. IoT HTR will notify the conductor where the wheel slippage is located and the severity of it
6. IoT HTR will suggest to slow down if minor slippage
7. IoT HTR will suggest to stop/brake if major slippage
8. IoT HTR will suggest to the conductor to accelerate with caution once it determines the wheel slippage has mitigated enough to be safe for faster speeds
9. Logger logs slippage

**Exceptions:**

1. If a sensor or more is down and IoT HTR is unable to make calculations, then a message stating which sensors are down will appear on the screen.

---

**Use case 5:** Hazardous weather conditions detection
**Primary actor:** Weather sensors
**Secondary actors:** IoT HTR
**Goal in context:** Notify operator for hazardous weather conditions
**Preconditions:** IoT HTR is initialized
**Trigger:** Weather based sensors detect hazardous weather conditions.
**Senario:**

1. Weather sensor(s) detects that there is hazardous weather conditions
    a. Hygrometer can measure humidity
    b. Thermometer can measure temperature
    c. Anemometer can measure wind speeds
    d. Rain Gauge can measure the amount of rainfall
2. IoT HTR will process the data given about the weather conditions
3. IoT HTR will notify the conductor for mild weather conditions.
4. IoT HTR will notify the operator for heavy weather conditions.
5. IoT HTR will suggest to slow down for mild weather conditions
6. IoT HTR will suggest to stop/brake for heavy weather conditions.
7. IoT HTR will suggest to the conductor to accelerate once it determines the weather conditions are safe to do so
8. Logger logs hazardous weather conditions

**Exceptions:**

1. If a sensor or more is down and IoT HTR is unable to make calculations, then a message stating which sensors are down will appear on the screen.

---

**Use case 6:** Uninitialize IoT HTR
**Primary actor:** Operator
**Secondary actors:** Admins
**Goal in context:** Turn off IoT HTR
**Preconditions:** IoT HTR is on
**Trigger:** Operator decides to turn off IoT HTR, that is, to press the power button
**Senario:**

1. Operator presses the power button
2. IoT HTR displays “Do you want to turn power off” message
3. Operator presses “yes”
4. IoT HTR timestamps when IoT HTR was uninitiated into the logs
5. IoT HTR Display turns off

**Exceptions:**

1. "No" is selected: IoT HTR closes the message and continues to display the operations of the HTR train

---

**Use case 7:** Download logs
**Primary actor:** Admin
**Secondary actors:** IoT HTR
**Goal in context:** To check and download operational logs
**Preconditions:** There is data in the logs, that is, IoT HTR is been turned on at some point; IoT HTR is on
**Trigger:** Admin decides to download the operational logs to check data, that is, to connect an auxiliary storage device

**Senario:**

1. Admin connects an auxiliary storage device to IoT HTR
2. IoT HTR detects an auxiliary storage device
3. IoT HTR request for admin ID/password
4. IoT HTR requests log(s) from logger
5. Logger copies the log(s) onto the storage device
6. IoT HTR displays "Log file(s) was successfully copied" message

**Exceptions:**

1. Incorrect admin login: Warning message "Incorrect user ID or password" is displayed, operator re-enters
2. Failed admin login five times: Warning message "Incorrect admin ID or password. Wait 5 minutes" is displayed
3. Error copying log file(s) onto the auxiliary storage device: Warning message "Error copying log file(s) onto storage device" is displayed, abort use case

---

**Use case 8:** Change configurations
**Primary actor:** Admin
**Secondary actors:** IoT HTR
**Goal in context:** Change the configuration of the HTR train
**Preconditions:** HTR train has IoT HTR system installed, IoT HTR is initiated
**Trigger:** Admin decides to change the configurations of the HTR train

**Senario:**

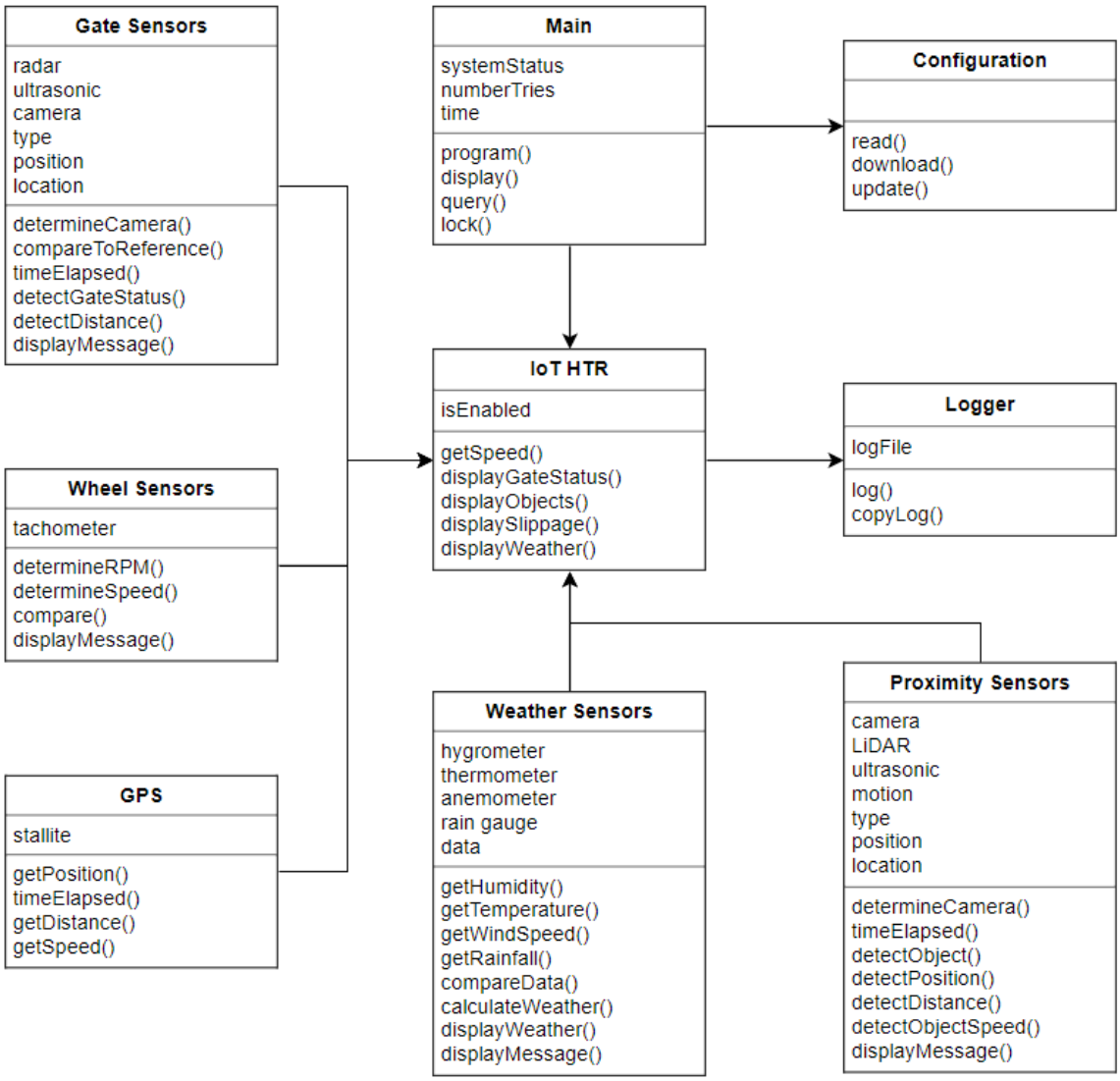
1. Admin supplies correct admin ID/password to IoT HTR
2. IoT HTR prompts admin if they would like to make changes to configurations
3. Admin selects "yes"
4. Admin makes appropriate changes to IoT HTR
5. Admin confirms changes and updates the system with new changes
6. Admin closes the configuration interface

**Exceptions:**

1. Incorrect login: Warning message "Incorrect user ID or password" is displayed, operator re-enters correct ID/password.
2. Failed to login five times: Warning message "Incorrect user ID or password. Wait 5 minutes" is displayed.
3. Fail to make configurations: Warning message "Failed to configure" is displayed, abort use case.
4. "No" is selected: IoT HTR closes the configuration interface and continues to display the operations of the HTR train.

## 4.2 Use Case Diagram

## 4.3 Class Diagram




## 4.4 CRC Model Index Cards

| Operator | |
|---|---|
| User who has limited access to IoT HTR | |
| Responsibility | Collaborator |
| Initiate IoT HTR | IoT HTR |
| Uninitiate IoT HTR | |

| Admin | |
|---|---|
| User who has complete access to IoT HTR | |
| Responsibility | Collaborator |
| Check logs | Logger |
| Change configurations | Configurations |

| IoT HTR | |
|---|---|
| Takes in all the data from the sensors, processes it, and determines if the operator needs to be alerted about hazards | |
| Responsibility | Collaborator |
| Initiate IoT HTR | |
| Uninitate IoT HTR | |
| Continuously log | Logger |
| Display obstructions if present | Proximity sensors |
| Display speed | GPS |
| Display weather | Weather sensors |
| Display slippage if present | Wheel sensors |
| Display gate status when applicable | Gate sensors |

| Proximity sensors | |
|---|---|
| Sensors that can detect objects (stationary and moving) within a 2 mile radius of the train | |
| Responsibility | Collaborator |

| | |
|---|---|
| Check if there are stationary obstructions | IoT HTR |
| Check if the stationary object is in the front or back of the HTR train | |
| Display the distance of stationary objects | |
| Check if there are moving obstructions | |
| Check if the moving object is in front or back of the HTR train | |
| Display the distance of moving objects | |
| Display the speed of obstructions | |
| Display warning message for obstructions and their position relative to the HTR train | |
| Display suggestion | |

| Wheel sensors | |
|---|---|
| Sensors that can detect if the train's wheels are slipping | |
| Responsibility | Collaborator |
| Determine the rpm of the train's wheels | IoT HTR |
| Determine the speed based on wheel's rpm | |
| Compare wheel sensor's speed with GPS's speed to check for slippage | |
| Display warning message for slippage | |
| Display suggestion | |

| Gate sensors |
|---|
| Sensors that can detect a crossing gate' status within a 3 mile radius of the train |

| Responsibility | Collaborator |
|---|---|
| Determines if crossing gates are opened/closed | IoT HTR |
| Determine the distance of a crossing gate | |
| Display warning message for open crossing gate | |
| Display suggestion | |

| Weather sensors | |
|---|---|
| Sensors that take in data regarding the current weather and its impact on the train | |
| Responsibility | Collaborator |
| Determine likely weather conditions | IoT HTR |
| Display likely weather conditions | |
| Display warning message for hazardous weather conditions | |
| Display suggestion | |

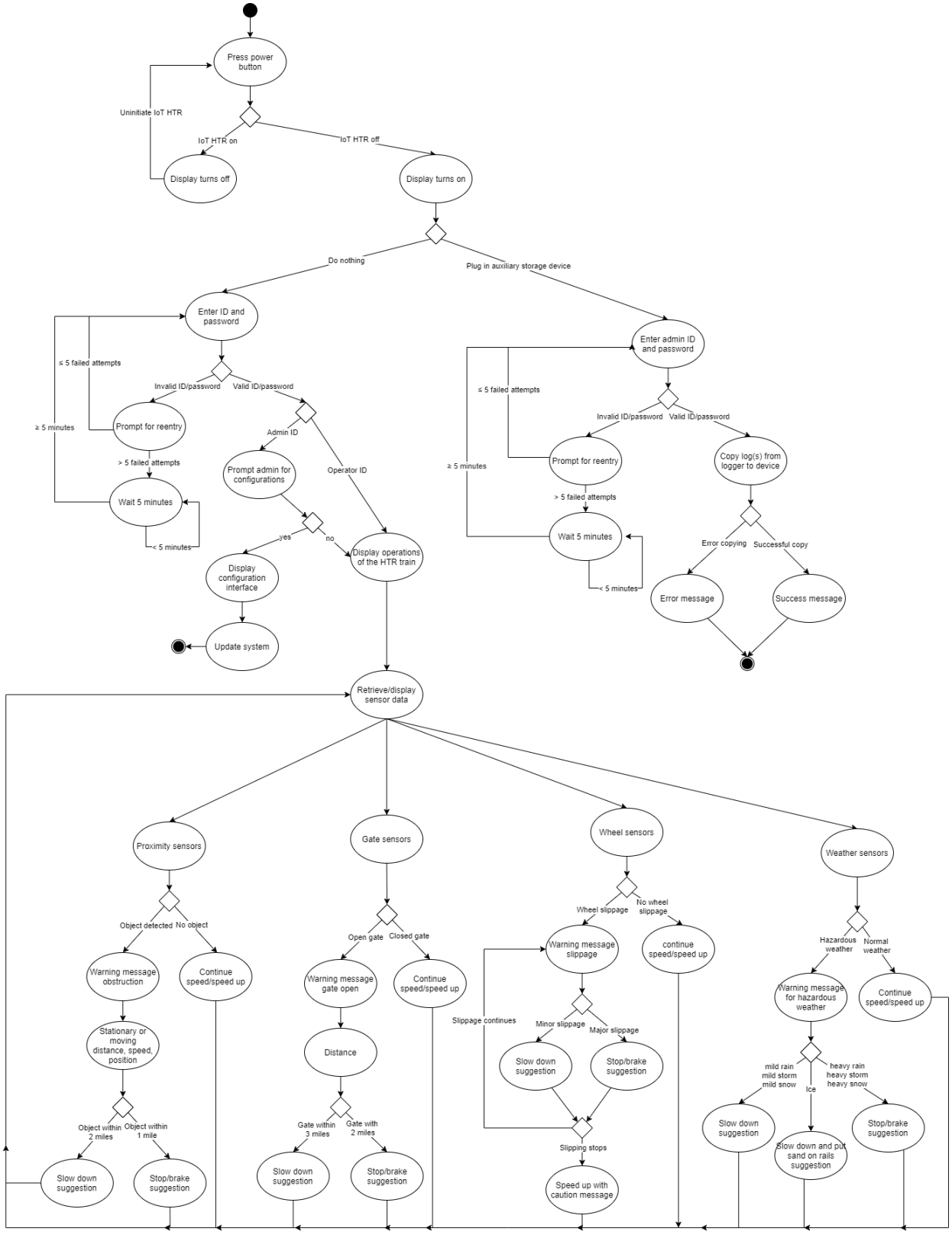
| GPS | |
|---|---|
| Gets the global positioning of the HTR train | |
| Responsibility | Collaborator |
| Get the speed of the HTR train | IoT HTR |

| Logger | |
|---|---|
| Logs all events while IoT HTR is on | |
| Responsibility | Collaborator |

| | |
|---|---|
| Get current time<br>Log events | IoT HTR |

| Configurations | |
|---|---|
| User configurations for IoT HTR | |
| Responsibility | Collaborator |
| Read admin input to change/update system settings | IoT HTR |

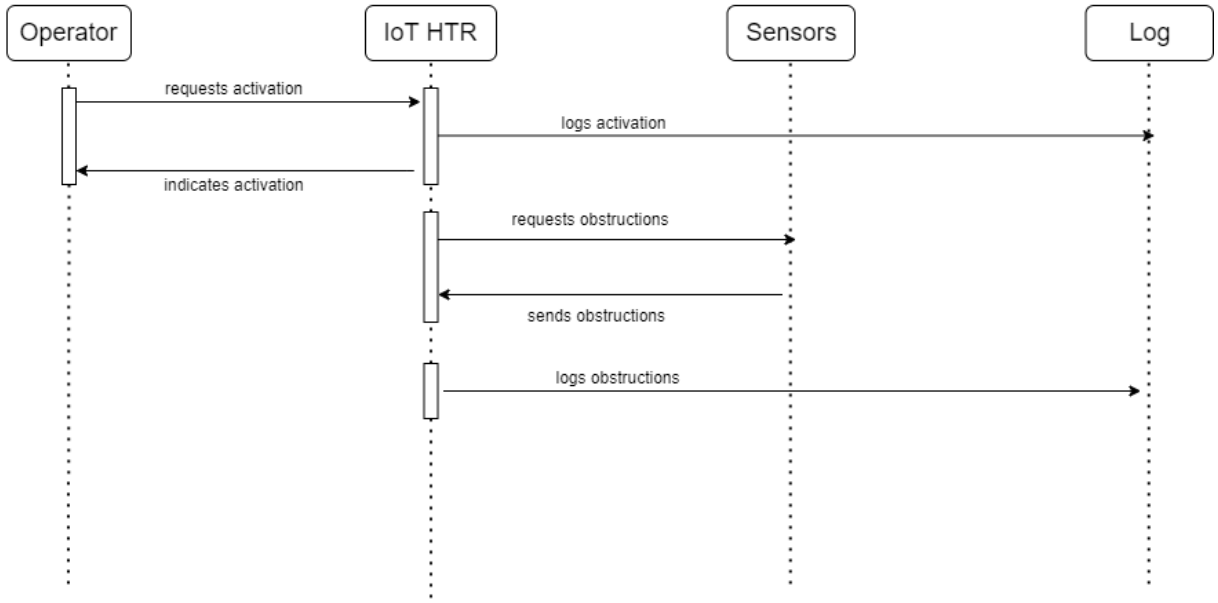
## 4.5 Activity Diagram



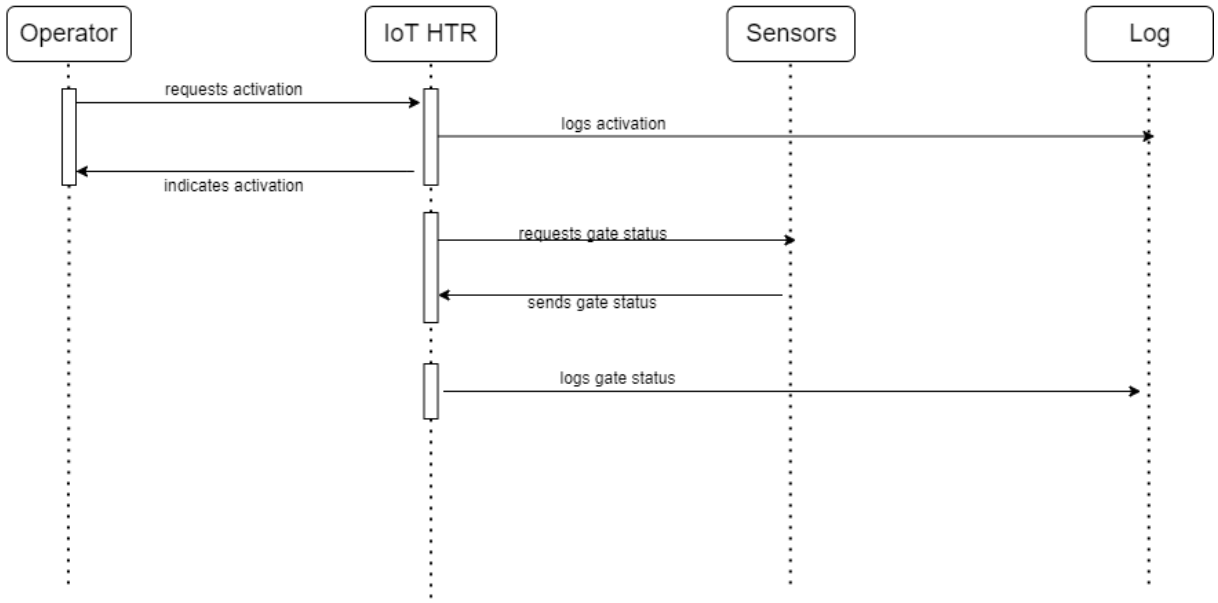
## 4.6 Sequence Diagrams

**Use case 1:**

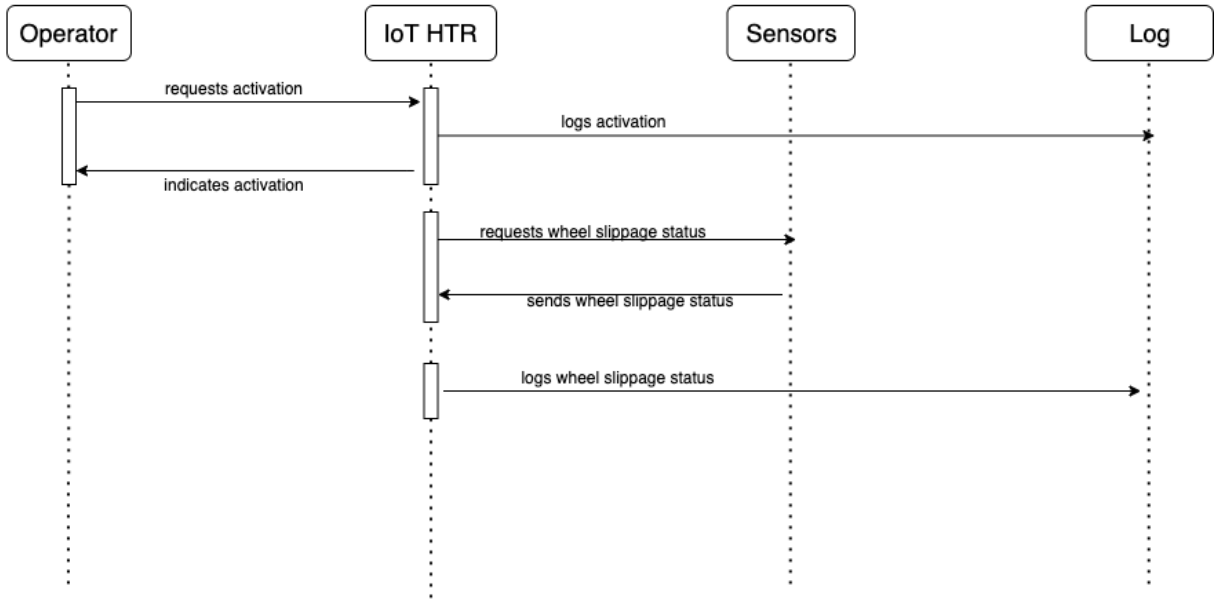
**Use case 2:**



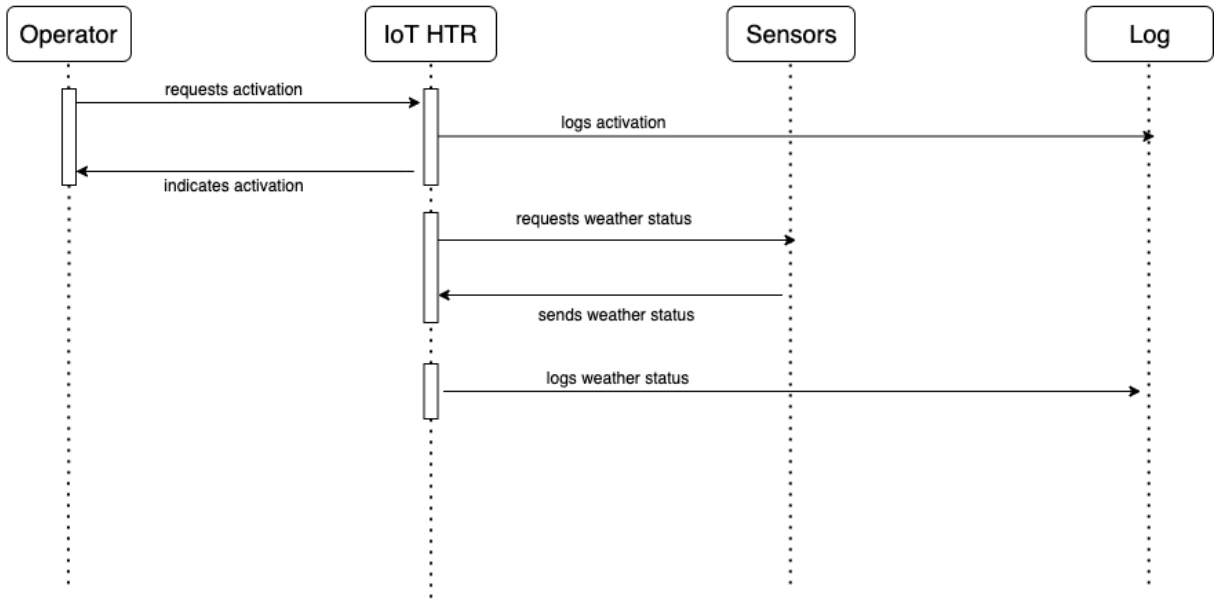
## Use case 3:



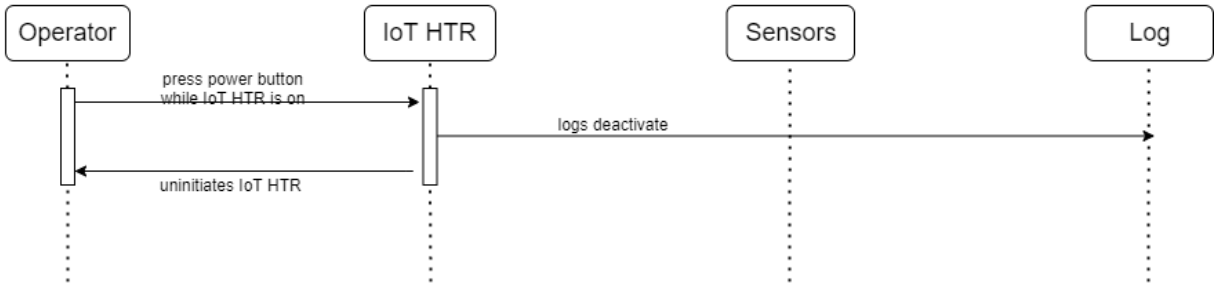

## Use case 4:



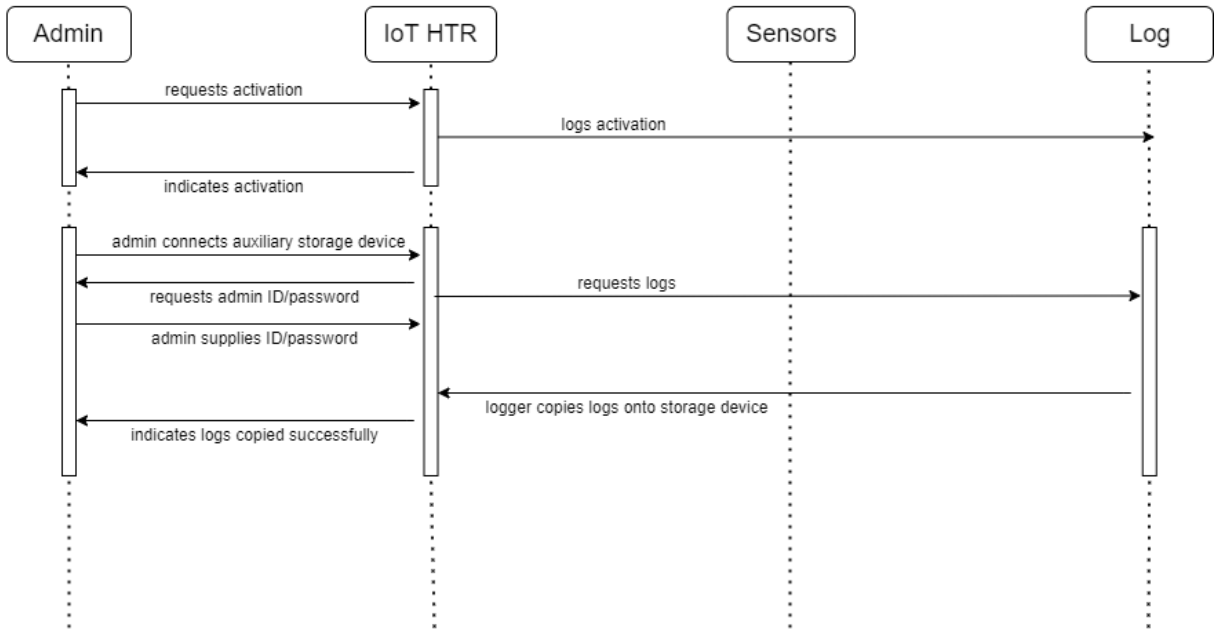
## Use case 5:



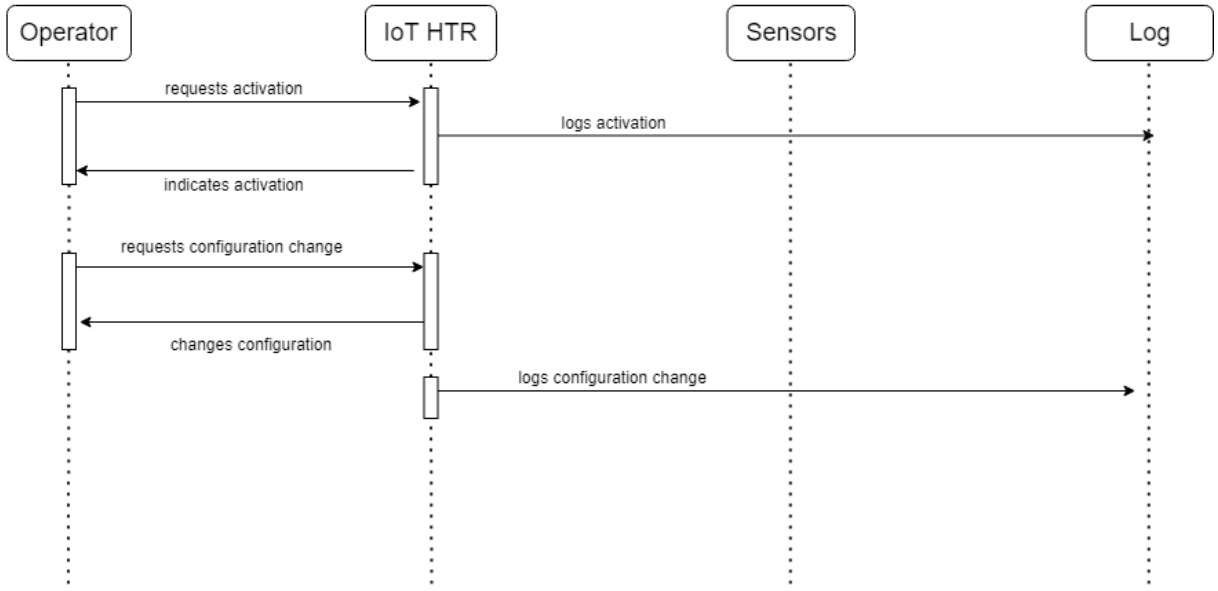

## Use case 6:



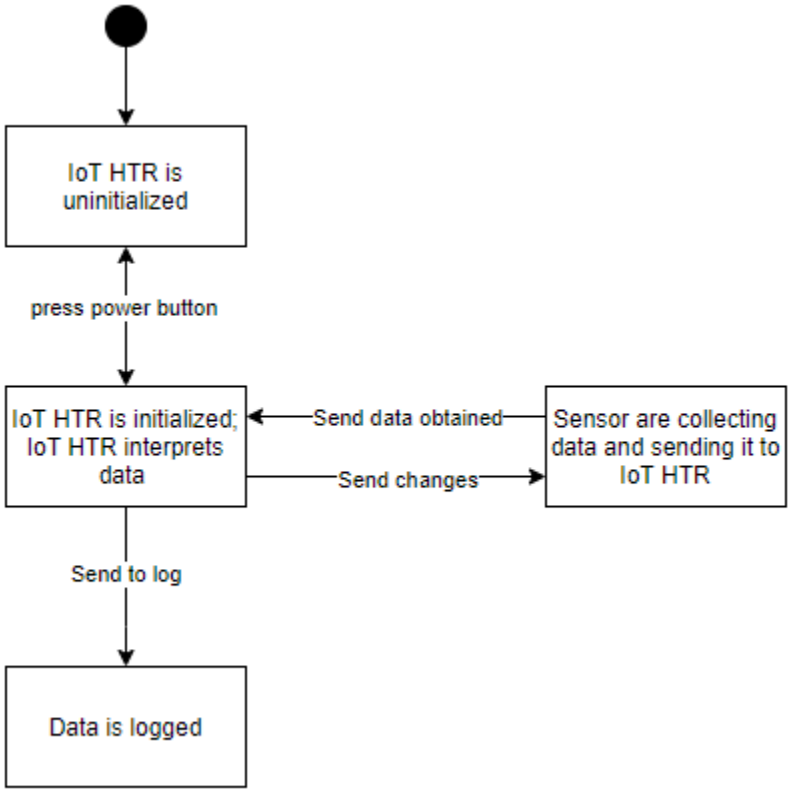
## Use case 7:




## Use case 8:



## 4.7 State Diagram



# 5. Software architecture

# 6. Code

# 7. Test Cases

# 8. Issues

# 9. Cost

Costs for any project can be daunting and often difficult to calculate because the project expands and evolves over time. We can give a rough cost estimation of this project based on a sample of devices one would buy. These costs are subject to change and there are multiple companies with varying prices that sell these sensors. It also should be noted that a large portion of the project will be software which does not have any external costs.

| **Sensor** | **Purpose** | **Unit Price** |
|---|---|---|
| Anemometer | Wind speed | $50-$800 |
| Hygrometer | Humidity | $200 |
| Rain Gauge | Rain | $300 |
| Thermometer | Temperature | $200 |
| Tachometer | Wheel's RPM | $50-$900 |
| LiDAR | 3D mapping | $500-$1,000 |
| Friction/Shear Force | Friction | $6,000-$8,000 |
| System of solid state circuits and pressure sensors | Gate | $1000+ |
| IoT screen and software | UI/UX | |
| GPS | Position/speed | $100-$1000 |
| Assemble | | |
| Radar | Status of crossing gate | |
| Ultrasonic sensor | Detect objects | |
| Cameras | Photos, object recognition | |
| Motion detection sensors | Detect moving objects | |