Group 6

Amraiza Naz
Aliya Iqbal
Adrian Gomes
Matthew Cunningham

Hug The Rails

IoT Project Overview

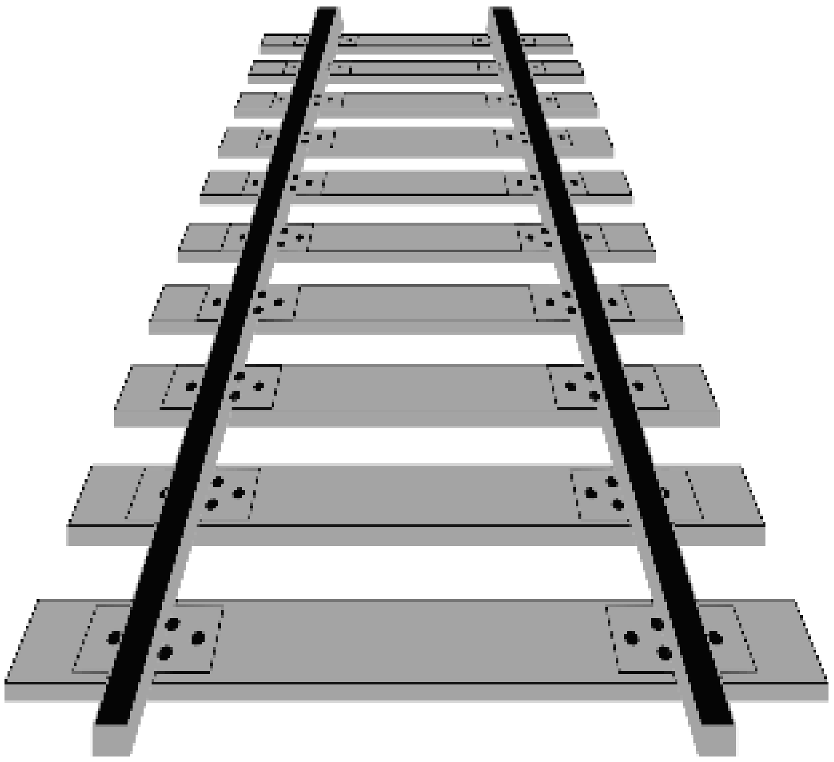

Table of Contents

1: Introduction

1.1 Problem Statement

Hug The Rails (HTR) uses the Internet of Things (IoT) to make HTR safer, less costly, and more efficient. By using IoT to make decisions locally in absence of cellular and wifi connectivity to Back Offices, Hug The Rails can capture data from locomotives and the environment. Unlike other train software, Hug The Rails uses an analytic engine, called an IoT Engine, to process information and make decisions passed on to the Locomotive Control System. We provide capability for the Locomotive operator to enter commands and receive status. Using IoT will allow the operator to maintain operations of the train in the event of a loss of wifi or cellular data, in order to ensure the safety of themselves, the passengers, and the cargo. To maintain code on the trains, operators can always download the latest rules for operation from the Fog/Cloud into the IoT Engine.

1.2 Our Stakeholders

Our stakeholders in this project include the train operators, who will operate the software; the owners, who will manage and oversee the Locomotive Control System; and the surrounding environment impacted by decisions made by the software developers, including riders on the train and customers relying on trains.

1.3 Our Audience (Users)

This software will be sold to train companies. The users are the owners of the train companies and the train operators who will use the software.

1.4 Importance and Value to Users

IoT is a decentralized train system that is safer for users and more efficient for train operators. Current train systems need a central authority to make decisions, such as a master control room, and this costs money, and wastes time and resources. It also means that users need constant access and connectivity with the central authority, and if trains lose this connection, trains are vulnerable to fatal consequences.

By contrast, IoT makes decisions locally in absence of cellular and wifi connectivity. This will allow for the operators to be aware of their surroundings and efficiently handle emergency situations.

1.5 Approach To Solution

The IoT approach is different from other train software. Sensors are at the foundation of all IoT design, allowing devices to collect data and interpret the environment. Therefore, our software will use sensors to measure distance between it and other objects to decide whether to accelerate or decelerate. It also has a touch-screen display that accepts and displays for an operator. The software can also send and receive between a central authority and other locomotives.

IoT can handle emergency situations. It’s weather, speed, and infrared sensors are among the few that will be able to operate in the absence of wifi and cellular data to ensure all train operations can still be handled at all times.

We will be implementing the unified process model in order to handle this project. This will allow us to continuously update and review our requirements, and make any changes if necessary. Additionally we will be able to maintain communication and involvement as a team throughout the project.

2: Overview

2.1 Summary Of The Problem

Many existing softwares for train operations don't include a safety mechanism when there is a loss of wifi or cellular data which could endanger both the operator and consumer. Our software will specifically address this issue, and we will ensure our software will continue to run even when there is no wifi or cellular data by installing IoT features.

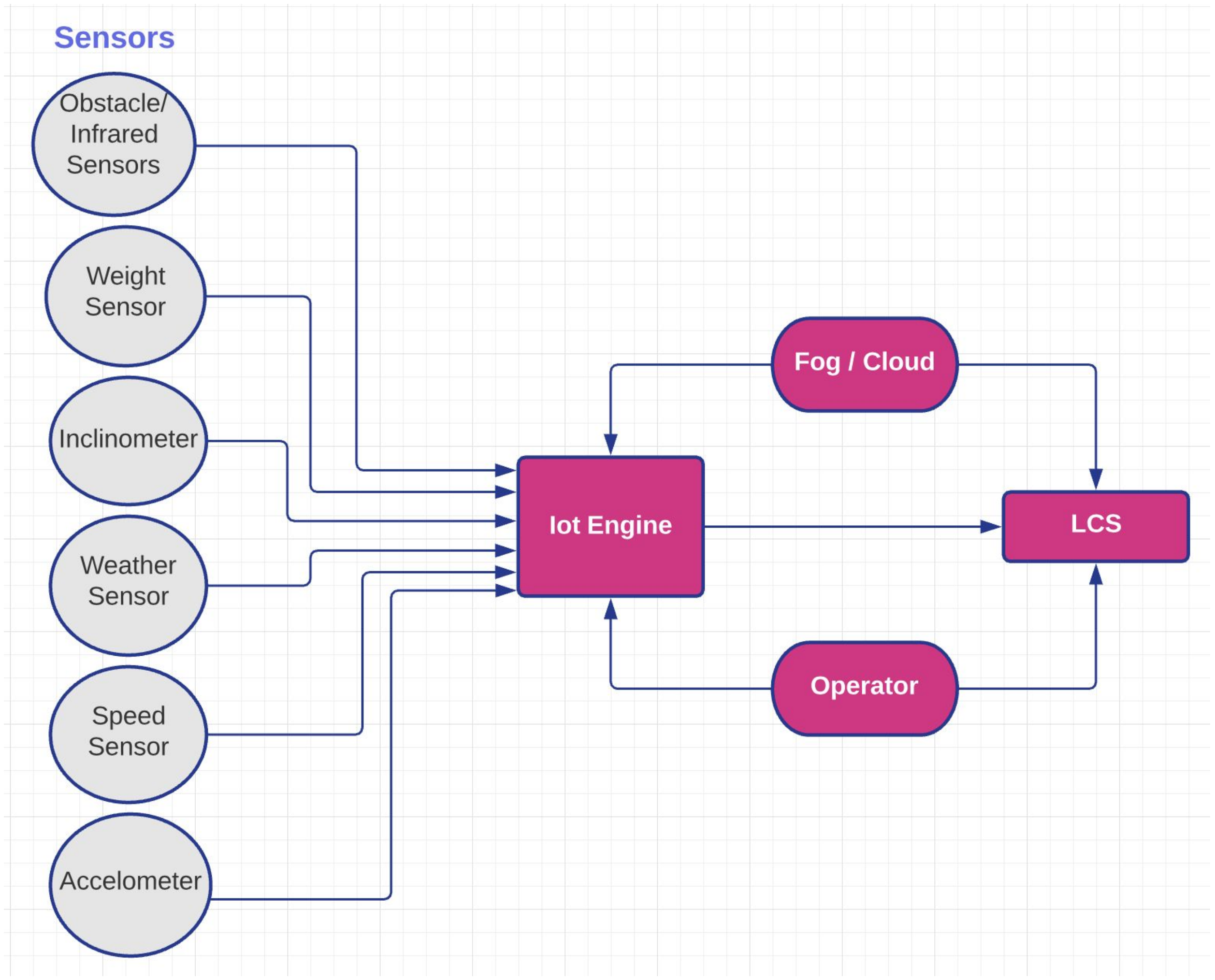

Figure 2.1.1: Conceptual architecture of IoT system.

2.2 IoT Features to be Implemented:

1. **Obstacle/Infrared Sensors:** These sensors will be placed on all sides of the train, and can emit/detect infrared radiation to measure objects up to 1000ft away. They will be used to inform the operator if there is something on the track that they need to slow down to avoid (such as animals, other trains, etc.).
2. **Weight Sensors:** These weight sensors are triggers that will be placed along the track at a fixed distance away from railroad crossings on both sides. When the train crosses each trigger, it will toggle the barriers, which will lower once the train is a certain distance away and raise when the train has passed.
3. **Inclinometer:** This sensor will be placed on the side of the train and be able to detect when the train is tilting due to curves in the track, obstacles, etc. If the sensor detects the train is at an angle beyond a certain threshold (with respect to gravity), it will alert the operator so its speed can be adjusted.
4. **Weather sensor:** This sensor will be placed on the windshield and will be able to detect air pressure, humidity, rainfall, snowfall, and wind speed. If the sensor detects extreme levels of weather, adjustments to the speed can be made as needed.
5. **Speed sensor:** These sensors will be placed on the wheels of the train and will detect how fast the train is going based on the rpm. This will be able to detect if the train is slipping due to snow, oil, etc. and allow adjustments to be made.
6. **Accelerometer:** Can detect if the train is speeding up or slowing down at an extreme rate so adjustments to the speed can be made as necessary

2.3 Phases Of The Process:

I. **Inception: 2/9 - 3/1**

   A. Establish project scope and boundaries

   B. Identify key features of project

   C. Define resources and technology needed for development

II. **Elaboration: 3/2 - 3/13**

A. Specify programming language and overall design

B. Define the methods that the application can communicate with other assets

C. Expand model into a **Life-Cycle Architecture** that captures most of the software's functional requirements

III. **Construction: 3/14 - 3/31**

A. Continuously check for any errors and ensure target goals are being met

B. Careful documentation of any changes made to code/requirements

C. Establish **Initial Operational Capability** that ensures the software is fully operational in a beta environment

IV. **Transition: 4/1**

A. Make software available to consumers

B. Check for deployment issues

2.4 Plans for Future Development:

1. **Automatic Driving Software**

Sometimes people have a fear that an automatic driving software will not work in time or result in an accident. To ensure the customer feels safe while sitting in the train we have implemented a design where if the operator wishes to do so he can override the software by simply just starting to drive themselves. When they drive, the software will not control the driving temporarily.

2. **"Auto Pilot"**

A train is always starting from one destination point and travelling to another. For the comfort of our customer we have installed a system where the software will be able to detect where the train is initially starting from, and by simply putting in a destination, the train will "auto pilot" and navigate itself to the location.

3: Requirements

3.1 Non-Functional Requirements

3.1.1 Reliability Requirements

R-1: IoT HTG shall be operable under extreme weather conditions in temperatures ranging from -50 to 150 F

R-2: IoT HTG shall stand drops (up to 5 feet)

R-3 IoT HTG system shall have reliability of 0.999.

3.1.2 Performance

R-4: IoT shall have a response time of 0.5 second, assuming IoT has been on.

R-5: IoT shall be able to support up to 1000 sensors

3.1.3 Security

R-6: IoT shall be accessed only by User ID/Password.

R-7: IoT shall be temporarily disabled after 3 failed login attempts

R-8: IoT shall require monthly updates to User ID/Password

R-9: IoT shall be able to register up to 3 fingerprints in lieu of a User ID/Password

3.1.4 Operating System

R-10: Reliable operating system shall be chosen for the IoT Engine.

R-11: Operating system executes IoT Engine indefinitely and locally.

R-12: Operating system shall be portable to allow for transfer between trains

R-13: Operating system shall be efficient to ensure reliable processing of different sensory data

3.1.5 Hardware

R-14: Weather sensors

R-15: Infrared sensors

R-16: Weight sensors

R-17: Accelerometer

R-18: Inclinometer

R-19: Speed sensor

R-20: Display

3.1.6 Network

R-21: IoT shall use a wired network to communicate with both the sensors and the display

R-22: IoT shall process and receive data from the sensors

R-23: IoT shall send the given data from the sensors to the display

3.2 Functional Requirements

3.2.1 Display on or off

R-24: operator shall be able to turn IoT on or off

R-25: IoT shall initialize required sensors

R-26: When turning software off the IoT shall deactivates the sensors

R-27: Also when turning the software off the screen shall display a goodbye message and a short clip of a train driving off before the screen shuts off

3.2.2 Display start up

R-28: IoT shall display a welcome message and logo of Hug the Rails and then start up the train

R-29: IoT shall require user ID and password after start up

3.2.3 Weather Conditions

R-30: Weather sensors shall process external temperatures

R-31: Weather sensors shall detect rain, snow, etc. on the windshield

R-32: Weather sensors shall send weather conditions to IoT

R-33: IoT shall display weather conditions to the operator

3.2.4 Obstacle Detection

R-34: Infrared sensors shall process infrared light

R-35: Infrared sensors shall determine distance of objects on track up to 1000ft

R-36: Infrared sensors shall determine the speed of objects moving ahead/behind the train

R-37: Infrared sensors shall send IoT data about any obstacles which may come in the way of the train

R-38: The IoT shall display data gathered from infrared sensors to the operator

3.2.5 Railroad Crossing Trigger

R-39: Weight sensor shall identify triggers on the track before approaching a railroad crossing

R-40: Weight sensor shall send a signal to railroad crossing barriers which shall send a message to the railroad crossing to raise/lower the barrier

3.2.6 Speed Control

R-41: Speed sensors shall detect the speed at which the wheels are rotating

R-42: Speed sensors shall send the train speed to IoT

R-43: IoT shall display speed of train to operator

3.2.7 Acceleration Control

R-44: Accelerometer shall detect the rate at which the train is speeding up/slowing down

R-45: Accelerometer shall send the data to IoT

R-46: IoT shall display the data to the operator

3.2.8 Curve Detection

R-47: Inclinometer shall detect when the train is at a sharp curve when its angle (with respect to the direction of gravity) exceeds 8˚

R-48: Inclinometer shall send an alert to the operator that the train is at a tilt so the speed can be adjusted

4: Requirements Modeling

4.1 Use Cases

Use Case: IoT startup

No. : 4.1.1

Primary Actor: Operator

Secondary Actor(s): IoT, Sensors

Goal: Activate IoT and begin processing data gathered from sensors. Display request for User ID/Password

Preconditions: Train has power

Trigger: Train ignition is on

Scenario:

- IoT system is powered by train
- IoT starts up and turns on sensor and Time-Sensitive Networking router (TSN)

Use Case: Validate User ID/Password

No. : 4.1.2

Primary Actor: Operator

Secondary Actor(s): IoT

Goal: To authenticate user credentials

Preconditions: IoT has power and is operating, operator has valid credentials

Trigger: IoT startup

Scenario:

- Display requests User ID/Password
- Operator enters credentials
- IoT processes credentials and determine validity

Use Case: Display data

No. : 4.1.3

Primary Actor: IoT

Secondary Actor(s): Sensors, TSN

Goal: To provide regular information about the train for the operator

Preconditions: Both train and IoT have power and are operating

Trigger: Train departs from station

Scenario:

- Sensors process data from surrounding area
- Sensors send data to TSN
- TSN sends data to IoT
- IoT processes and displays data to the operator

Use Case: Display warning

No. : 4.1.4

Primary Actor: IoT

Secondary Actor(s): Sensors, TSN, Log File

Goal: To alert the train operator about a possible emergency situation

Preconditions: Both train and IoT have power and are operating; sensors are operating and connected to IoT.

Trigger: Data from sensors meet conditions that would require the operator to slow down the train

Scenario:

- Sensors detect extreme conditions (eg. snow, rain, moving objects) on the track
- Sensors send data to TSN
- TSN sends data to IoT
- IoT processes data and displays a warning message to the operator
- Log file records incident with timestamp

Use Case: Access log data

No. : 4.1.5

Primary Actor: Technician

Secondary Actor(s): IoT, Log file

Goal: To view all recorded data from sensors

Preconditions: Train is on, Logged in as technician

Trigger: Technician requests log file

Scenario:

- Display log file data

4.2 Use Case Diagram

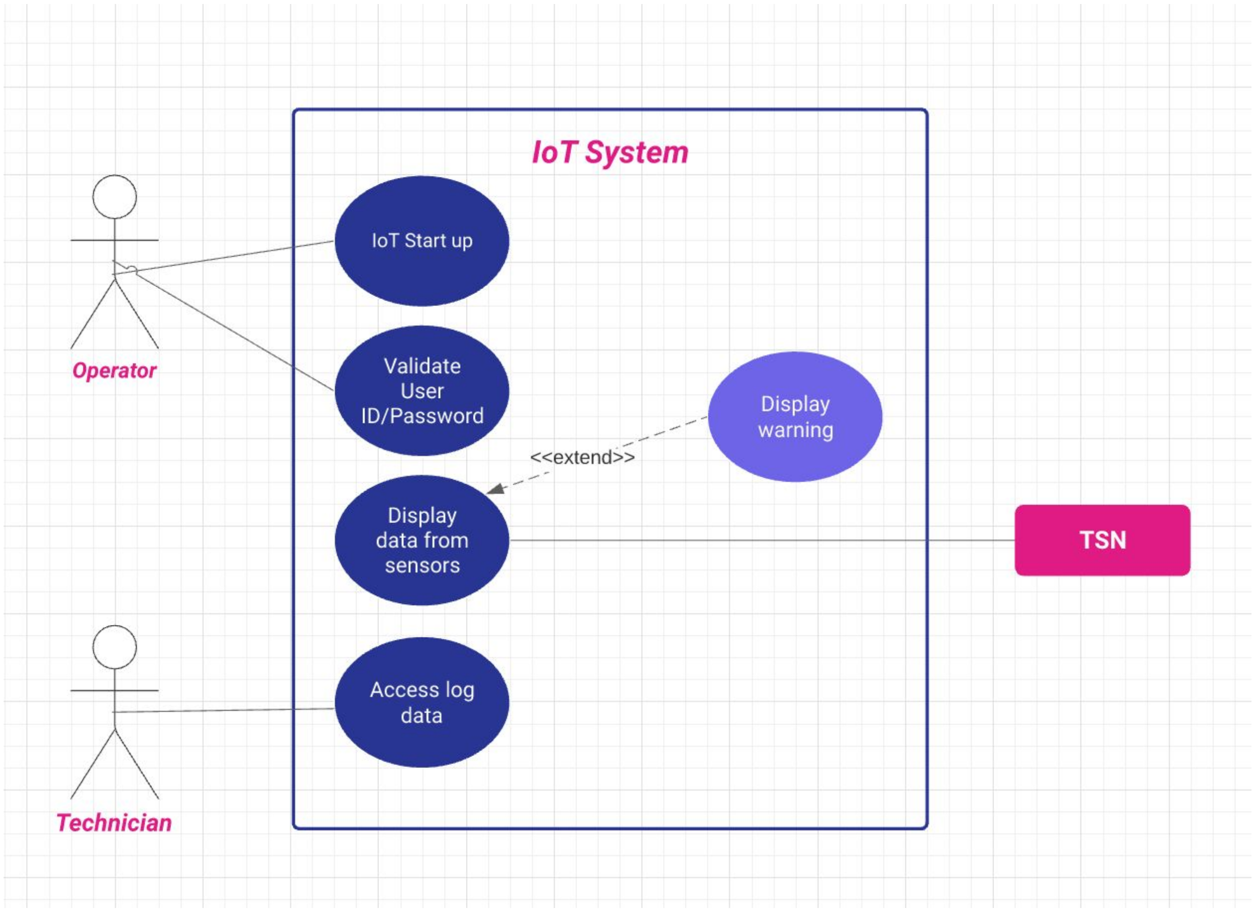

Figure 4.2.1: UML diagram for IoT system use cases.

4.3 Class-Based Modeling

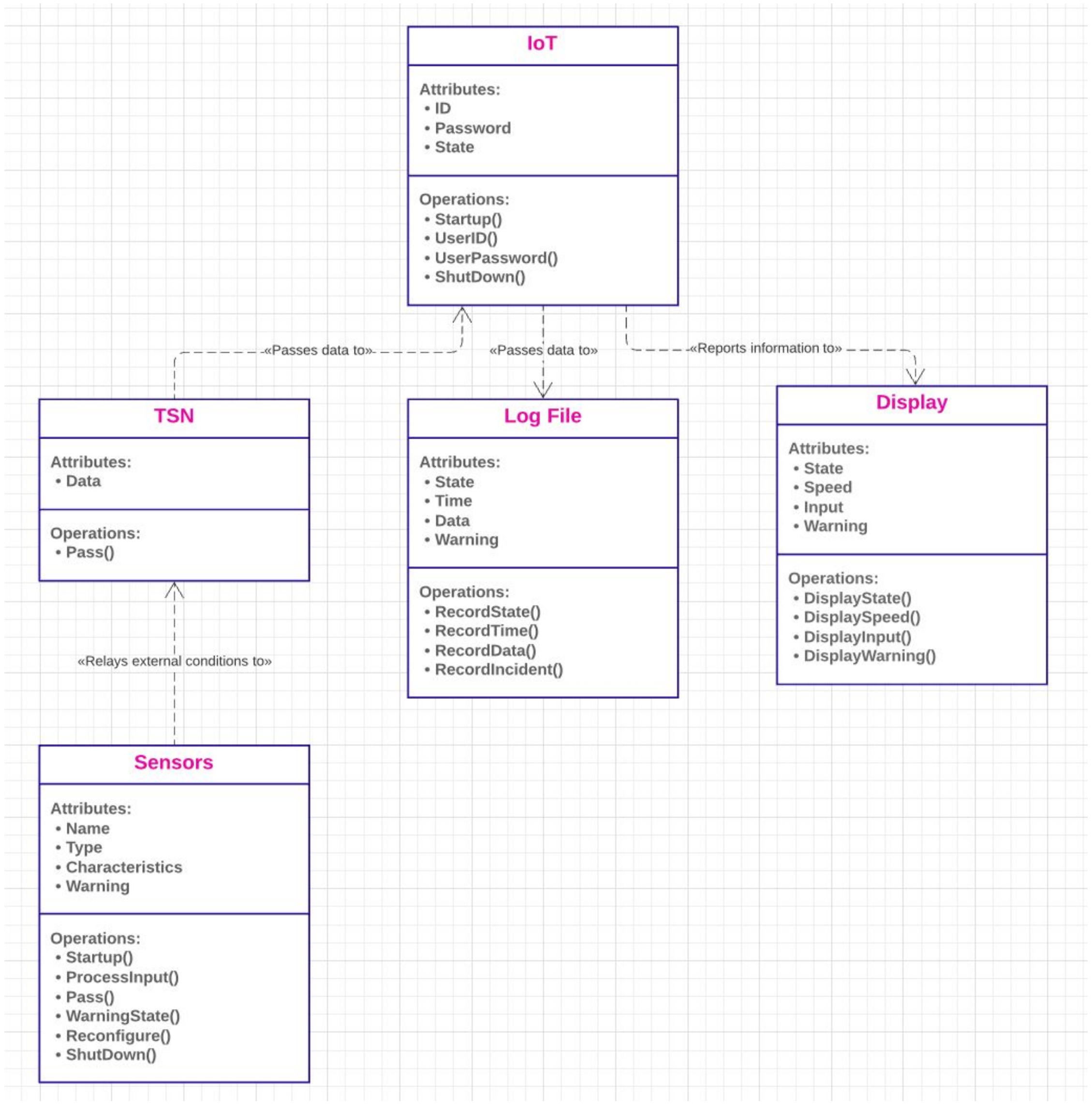

Figure 4.3.1: UML Class-Based Model of IoT system.

4.4 CRC Modeling/Cards

Class: IoT
Validate user and process sensor data retrieved from TSN
Responsibility: Request User ID/Password from operator **Collaborators:** display
Responsibility: Gather sensor data from TSN **Collaborators:** TSN, sensors
Responsibility: Display sensor data to operator **Collaborators:** display
Responsibility: Display warnings/suggest speed changes based on sensor data **Collaborators:** display, sensors

Class: Sensors
Handles functions and attributes for every sensor
Responsibility: Detect conditions from surrounding area **Collaborators:**
Responsibility: Add sensor data to the technician log **Collaborators:** Log File
Responsibility: Send data to TSN **Collaborators:** TSN

Class: Display
Display sensor data to the operator
Responsibility: Report speed of train **Collaborators:** IoT, TSN
Responsibility: Report weather conditions (rain, snow, etc.) **Collaborators:** IoT, TSN
Responsibility: Report any moving obstacles on track

Collaborators: IoT, TSN

Class: TSN
Processes and sends data from sensors
Responsibility: Receives data from all the sensors and sends it to IoT **Collaborators:** IoT, sensors

Class: Log File
Keeps track of speed changes, general sensor data, and warnings
Responsibility: Record speed change **Collaborators:** sensors
Responsibility: Record sensor data at ___ time intervals **Collaborators:** sensors
Responsibility: Record warnings sent by sensors to IoT **Collaborators:** sensors

4.5 Activity Diagram

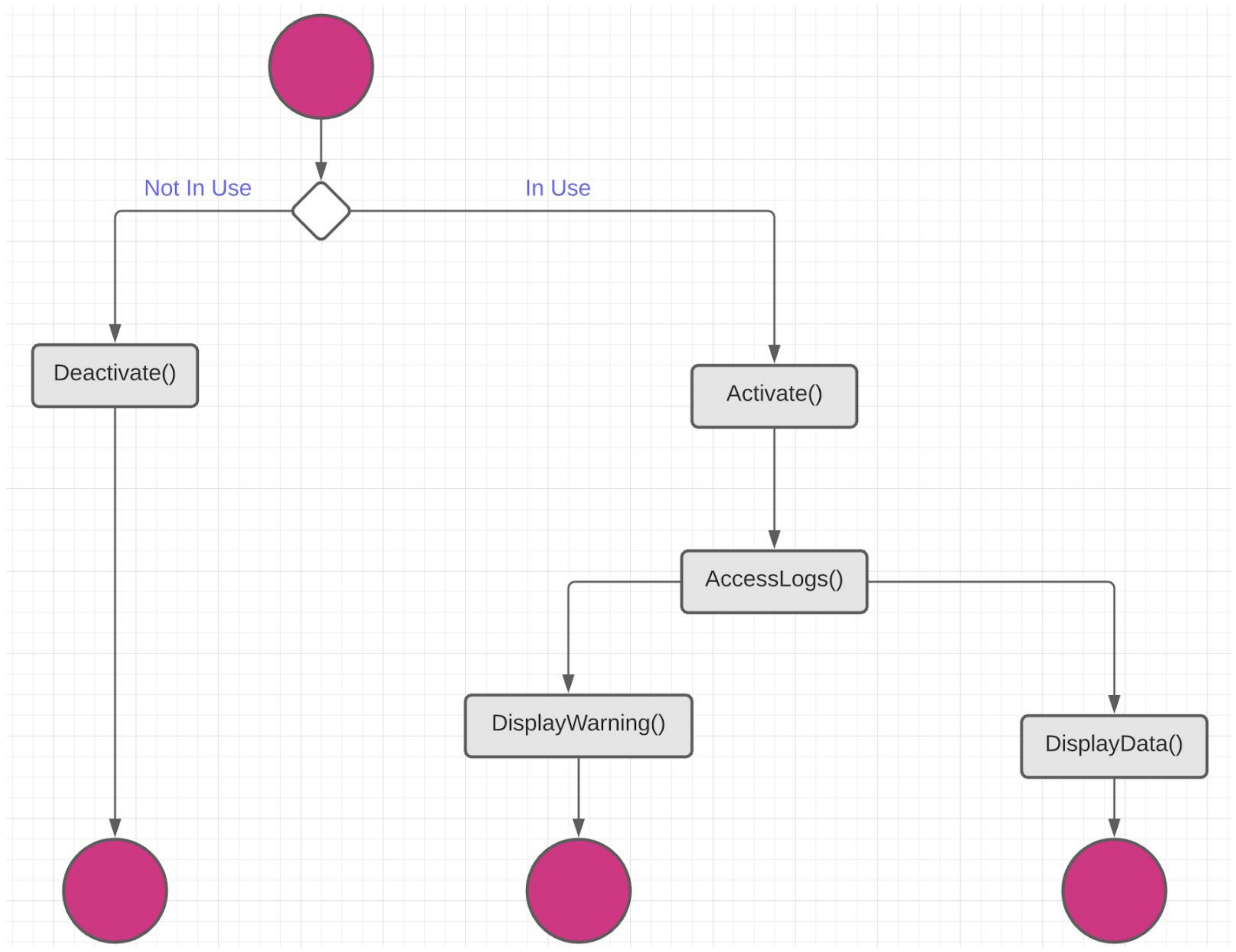

Figure 4.5.1: UML activity diagram for IoT system.

4.6 Sequence Diagram

Use Case: IoT startup (4.1.1)

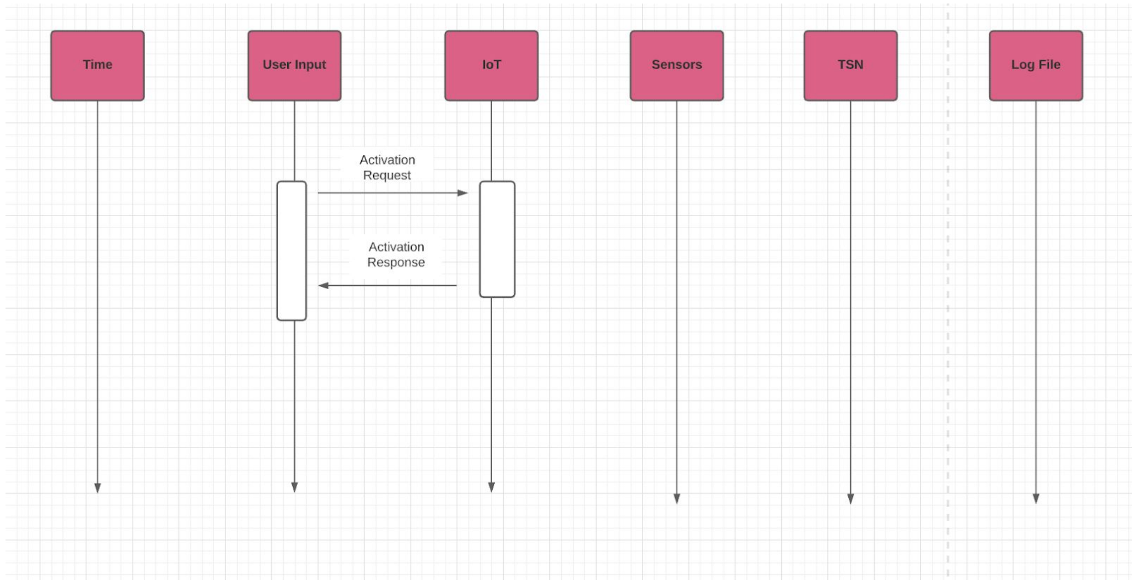

Figure 4.6.1: UML sequence diagram for IoT system use case IoT startup (4.1.1).

Use Case: Validate User ID/Password (4.1.2)

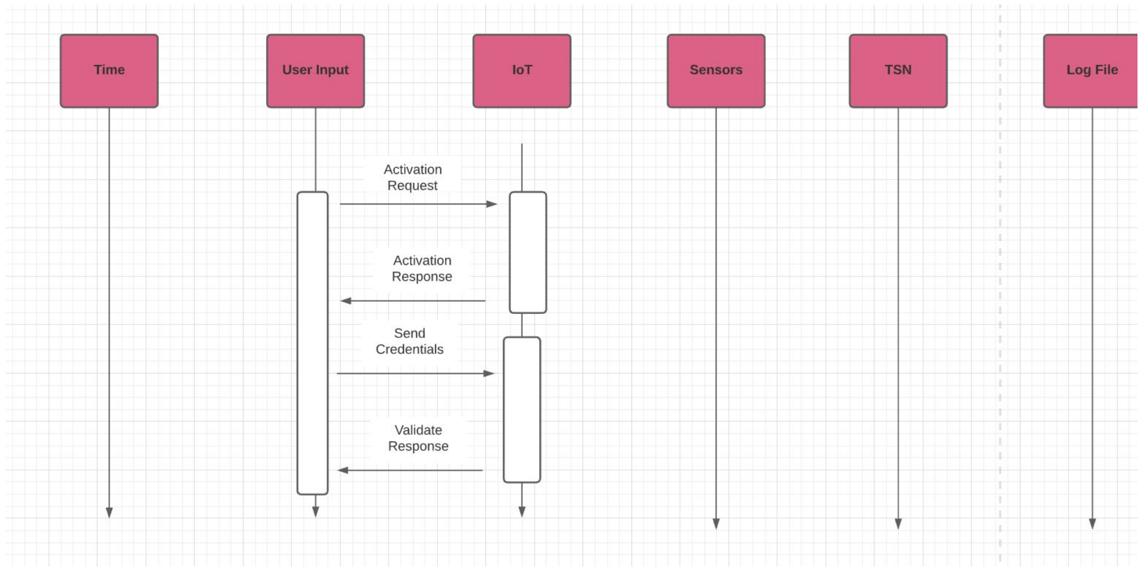

Figure 4.6.2: UML sequence diagram for IoT system use case validate user id/password (4.1.2).

Use Case: Display data (4.1.3)

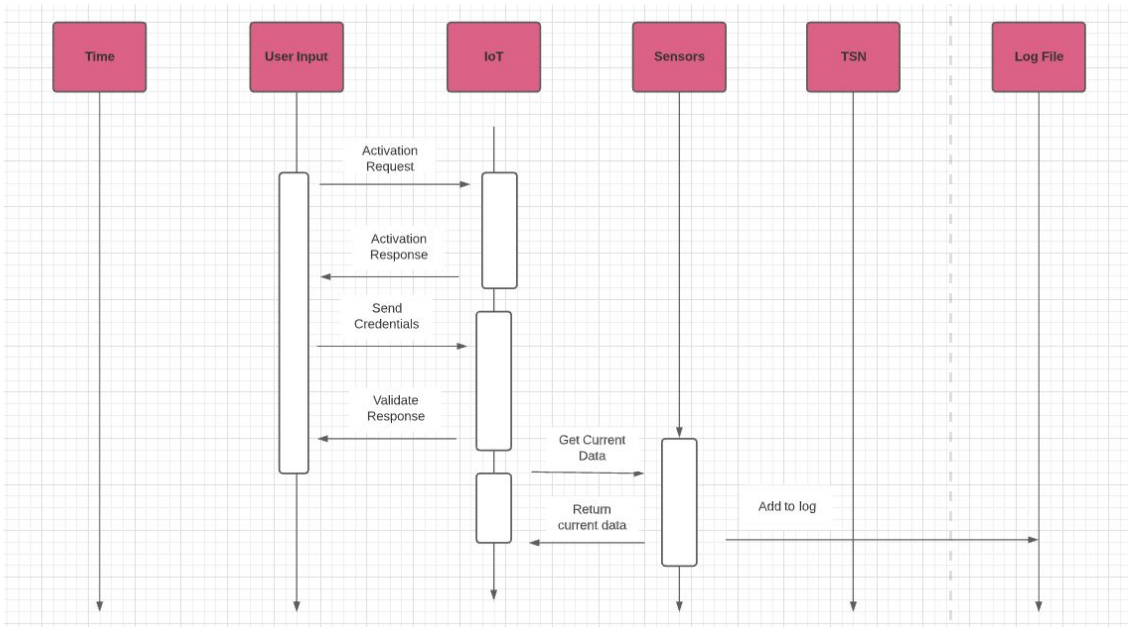

Figure 4.6.3: UML sequence diagram for IoT system use case display data (4.1.3).

Use Case: Display warning (4.1.4)

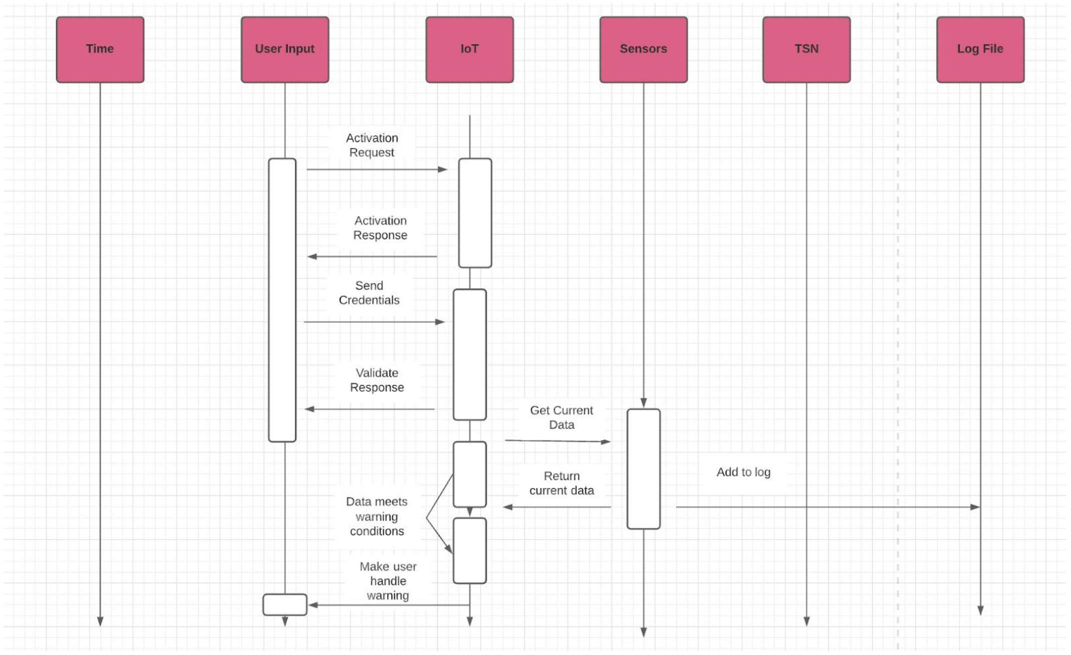

Figure 4.6.4: UML sequence diagram for IoT system use case display warning (4.1.4).

Use Case: Access log data (4.1.5)

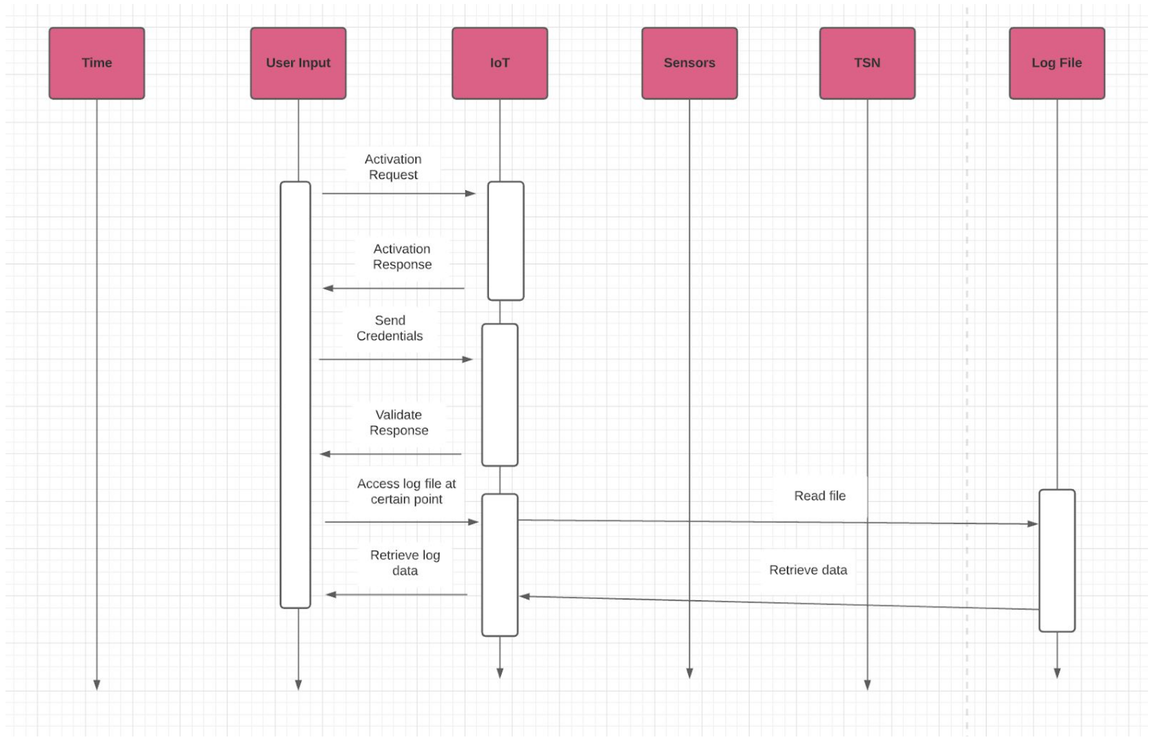

Figure 4.6.5: UML sequence diagram for IoT system use case access log data (4.1.5).

4.7 State Diagram

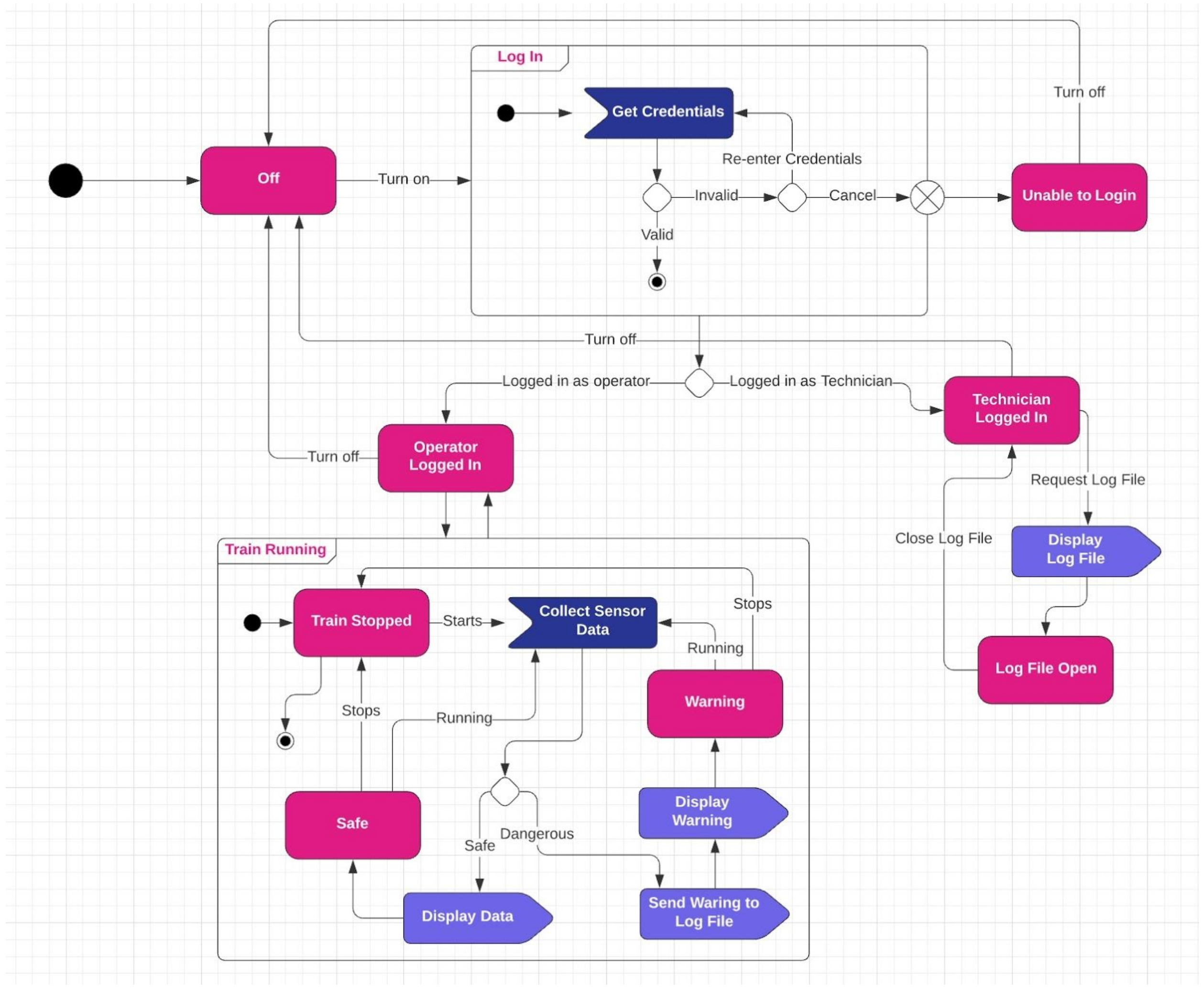

Figure 4.7.1: UML state diagram for IoT system.