IoT HTR: Formal Review Document

This document will serve as the minutes of the formal review of the IoT HTR project.

Meeting date: May 8th, 2021
Meeting duration: 6 p.m. – 7:30 p.m.

Review Roles:

- Producers: Arushi Garg, Lasya Josyula, Sanjana Madhu, Troy Tomasch
- Review Leader: Arushi Garg
- Reviewers: Lasya Josyula, Sanjana Madhu, Troy Tomasch, Amna Ahmad (student from CS 347 section B), Jacob Naeher (student from CS 347 section B)
- Recorder: Sanjana Madhu

Meeting Agenda:

- View a demo of the project
- Review requirements of the project
- Review the software development process
- Discuss pros and cons of the implementation and possible improvements
- Review issues list
- Review outcome and recommend action to the Hug The Rails CTO

1. View a demo of the project

A demo of the product was viewed with different testing variables. The source code was also viewed with detailed explanations of each section and class.

Pros:

- The team liked the display of data and believe that it easily shows the conductor what kind of action needs to be taken
- The code is documented well and very easy to understand for people who were not a part of this project
- The flow of data is clear and easy to understand. The team took advantage of the temporal flow of data and utilized that in the source code
- The code has gone through heavy testing and the team has checked the functionality during many edge cases

Cons:

- It is difficult to simulate how the actual system would work without a true connection to the sensors. The csv file works well, but the team is unsure how the software will work when there is an actual connection to the sensor
- The source code may look neater if the separate classes are split into separate files. This way, a dependency map can be created and only classes that interact with each other will have access to each other

2. Review requirements of the project

The requirements of this project were read out and assessed by the team.

- R1: Needs further testing. The issuing company verified that the sensors are waterproof, but this has not been fully tested.
- R2: Needs further testing. The issuing company verified that the sensors can last 1000 hours and operate 99.99% of the time, but this has not been fully tested.
- R3: Fulfilled.
- R4: Fulfilled.
- R5: Fulfilled.
- R6: Needs further testing. The issuing company verified that the sensors are failure-free for 1000 hours, but this has not been fully tested.
- R7: Needs further testing. The issuing company verified that the sensors gave a 99.99% reliability but this has not been fully tested.
- R8: Requires improvement. The team has implemented a username and password for conductors and administrators, but the method is not secure. Firebase and two-factor authentication can be used to make the system more secure.

- R9: Requires improvement. Currently, the only way to manually override the system is to exit the program.
- R10: Fulfilled.
- R11: Fulfilled.
- R12: Fulfilled.
- R13: Fulfilled.
- R14: Fulfilled.
- R15: Fulfilled.
- R16: Fulfilled.
- R17: Fulfilled.
- R18: Requires improvement. Currently, the display does not show the current speed of the train resolved into a whole number.
- R19: Fulfilled.
- R20: Fulfilled.
- R21: Fulfilled.
- R22: Needs further testing. The camera is separate from the system and needs to be tested during a legitimate ride.
- R23: Fulfilled.
- R24: Fulfilled.
- R25: Fulfilled.
- R26: Fulfilled.
- R27: Requires improvement. Currently, the size of the wheel is not manually inputted for different types of trains.
- R28: Fulfilled.
- R29: Fulfilled.

3. Review the software development process

The software development process was reviewed. This process included Use Cases, Class Diagram, CRC Modeling, Activity diagram, Sequence Diagrams, State Diagrams, and Architecture Models.

Pros:

- The use cases and the use case diagram are very detailed. They showcase strong command of the problem and comprehensive approaches to the solution
- The use case diagram shows how the actors interact with the system and the use cases
- The class diagram is organized and shows how each class interacts with one another in a high-level view
- The activity diagram is nicely organized and color-coded. It is easy to follow the flow of this project

- The sequence diagrams are detailed and show how each component interacts with the system
- The state diagrams are detailed and show how each component interacts within itself

Cons:

- The CRC modeling should also include the variable definitions and their purpose
- The activity diagram can also include other decisions in the process

4. Discuss pros and cons of the implementation and possible improvements

The pros and cons of the implementation were discussed next by the group.

Pros:

- The implementation follows each of the requirements set by the CTO of Hug The Rails and meets the expectations of the CTO
- The implementation was completed on time according to the timeline and requires minimal improvements
- The implementation is organized and easy to test, evolve, improve, and push to production

Cons:

- Some minor improvement is required for some of the requirements

5. Review issues list

This is the comprehensive list of improvements that can be made.

- Test the system, sustainability of sensors, and camera during a legitimate train ride
- Split source code into multiple files for neatness
- Implement two-factor authentication
- Find a way to manually override the system
- Constantly show the speed of train on the display
- Allow the conductor to manually input the size of the wheel

6. Review outcome and recommend action to the Hug The Rails CTO

After the changes have been made to this project, the team has decided that the project can be accepted provisionally. There are a few minor errors and enhancements the team can make to this project. A follow-up procedure will be instated to ensure that the errors are fixed and another formal review will be not conducted. To view the issues, the team can view the cons, unfulfilled requirements, and the comprehensive review list.