Bus Passenger Counter with Analytics & Dashboard Visualization

ECE4871 Senior Design Project

IntelliBus
Project Faculty Advisor, Dr. Vijay Madisetti

Shadman Ahmed, CMPE, sahmed85@gatech.edu
Noah Chong, CMPE, nchong8@gatech.edu
Yue Pan, CMPE, ypan331@gatech.edu
Thomas Talbot, CMPE, ttalbot7@gatech.edu

Submitted

April 14, 2021

Table of Contents

Executive Summary

The IntelliBus system is an IoT device and software suite that will track bus location and the aggregate number of riders on buses. The ongoing pandemic has shown that human congestion and high traffic on public areas and transportation can lead to higher virus transmission and exposure. Real-time data on the number of passengers on a bus can help with this issue.

The IntelliBus system plans to solve this problem through an IoT ecosystem comprised of a microcontroller with sensors connected through cellular LTE-M networks and a cloud-based web application that will analyze and visualize incoming sensor data. Each IoT module will cost $161.00 per unit with an expected $30,915.00 cost of labor to install, design and develop the IoT ecosystem. Performance will be based on the accuracy and latency of both the IoT devices and web application. Acceptable tolerances are ±5% and 5 minutes, respectively.

Solutions similar to the proposed project exist outside of the United States, including China-based “Beijing Bus”, that displays public transportation capacity via a mobile app. Real-time bus data has measurable benefits outside of the pandemic as well, including optimization of public transportation fleet schedules and valuable meta-data that could be applied towards better city planning, infrastructure, and economics. Integration with a larger traffic authority, such as Georgia Tech PTS or MARTA, would be the logical next step for the IntelliBus project.

Nomenclature

AWS (Amazon Web Services) - a subsidiary of Amazon that offers cloud computing platforms and application programming interfaces to individuals, governments, and businesses

CapEx (Capital Expenditure)- spends to buy, maintain, or improve fixed assets, such as servers or equipment.

CPU (Central Processing Unit) - the portion of a computer that fetches and executes instructions

GPIO (General-Purpose Input/Output) - uncommitted digital signal pin on an integrated circuit that has no predefined purpose and is unused by default

GPS (Global Positioning System) - satellite-based radionavigation system owned by the United States government

GUI (Graphical User Interface) - a user interface that includes graphical elements, such as windows, icons and buttons

HTTP (Hypertext Transfer Protocol) - application layer protocol designed for communication between web browsers and web servers

IoT (Internet of Things) - a network of interconnected devices – from simple sensors to smartphones and wearables

IC (Integrated Circuit) - a set of electronic devices integrated onto a small piece of material (usually silicon) to achieve a certain function

LTE-M (Long Term Evolution for Machines) - low power wide area technology that enables a wide range of cellular devices and services

MARTA (Metropolitan Atlanta Rapid Transit Authority) - the principal public transport operator in the Atlanta metropolitan area

ML (Machine Learning) - is a method of data analysis that automates analytical model building noSQL (Non Structured Query Language) - provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

OpEx (Operational Expenditure)- is an ongoing cost for running a product, business, or system.

PERT (Program Evaluation and Review Technique) - variation of the critical path method that examines tasks and dependencies to calculate the minimum time to finish a project

PIR (Passive Infrared) - a type of motion sensor that detects motion by sensing its higher infrared emission than the surrounding environment

PTS (Parking and Transportation Services) - the primary traffic authority at Georgia Tech

QFD (Quality Function Deployment) - method used to identify specific links between customer attributes and engineering requirements

RAM (Random Access Memory) - computer memory that can be read in any order - often used to store dynamic data and machine code

RTOS- A real-time operating system is an operating system intended to serve real-time applications that process data as it comes in, typically without buffer delays.

SDK (Software Development Kit) - a collection of software development tools in one installable package

SIM (Subscriber Identification Module) - a smart card inside of a mobile device that carriers an identification number unique to the user

TCP/IP (Transmission Control Protocol and Internet Protocol) - the two primary transport and network layer protocols that govern computer connections to the internet

USB (Universal Serial Bus) - industry standard that establishes protocols for data transfer and power supply between devices

Wi-Fi - a group of wireless network protocols based on Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards

Bus Passenger Counter with Analytics & Dashboard Visualization

1. Introduction

The team will design IntelliBus, an affordable solution for real-time people counting and data analysis system for buses. It is designed to aid transportation authorities achieve higher utilization of fleet resources and provide better service to their customers. The team requests $500 to prototype the IntelliBus.

1.1 Objective

The objective of IntelliBus is to create an affordable solution for bus authorities to track the number of passengers on board a certain bus route. Two main components are needed to implement this design.

First, the hardware end consisting of a sensor array, a microcontroller with built-in LTE-M and GPS support will be installed on the buses. The hardware end is responsible for using the microcontroller to process the sensor readings and calculate the number of passengers on board the bus. A software routine will periodically transmit the people count and current GPS location to the cloud server. To keep power and Internet data usage within a reasonable threshold, the system will transmit live people count data every five to ten seconds.

Second, the software end deployed on the cloud will be responsible for processing and visualizing the data transmitted from buses. Using cloud technologies, the team attempts to create a web application backed by a cloud-based database. Upon receiving updates from the hardware installed on buses, the cloud software will produce a real-time visualization of the changes throughout a bus's route.

1.2 Motivation

The team's motivation for IntelliBus comes from the current state of public transportation, which the COVID-19 crisis has tremendously altered. Regulations regarding the population density within confined spaces such as buses are in place globally. With decreased vehicle capacity and fluctuating demand, it is crucial for transportation authorities to revise their route schedules for better profitability, efficiency, and service. Accurate data support is the key. A tool for analyzing and visualizing the number of passengers on board a specific bus will help decision makers understand passenger traffic along the routes and identify hotspots or underutilization. In non-pandemic times, IntelliBus can also provide general assistance to transportation authorities in optimizing fleet resource deployment, for example, evaluating the effectiveness of a new route.

1.3 Background

The need for people counting systems is prevalent. During COVID-19, stores can use it to limit the number of customers inside, and bus companies can use it to notify people how crowded an oncoming bus is. Traditionally, there are mainly two types of designs for a people counting system. The first type utilizes PIR sensors and is adopted in convenient stores and theft-prevention products. The technology relies on interpreting a person's movement into an infrared source different from the surrounding environment. The second type uses video-based recognition technologies. This type can be seen in bus companies; for example, public transportation authorities in Beijing and Singapore display the crowdedness of buses on their mobile app. Adding an edge computing device to the already installed security camera on buses, companies like Beijing Transport can achieve real-time decoding of the video feed, thus producing a live count of the number of passengers on board. However, implementing the same technology may face problems in America due to concerns regarding personal privacy, and potential costs of revising existing structures may jeopardize the low-cost intention of IntelliBus.

People counting systems are ineffective if the result is only shown locally to the driver or store owner. The connection between the hardware end consisting of microcontroller and sensors and the cloud is the core of IoT systems. In the recent decade, with the lowering cost of digital ICs and their growing versatility, IoT devices have been increasingly integrated into society as a stepping stone toward the future interconnected world. The emergence of IoT devices enables and inspires the design of IntelliBus. Alongside the hardware components, the cost of cloud service has also been dropping, with many providers such as Google Cloud and AWS offering free credits for users. The organization and properties of IoT systems make designs like IntelliBus low-cost and mass-deployable.

2. Project Description, Customer Requirements, and Goals

The goal of this project is to design and prototype an IoT device that will track bus location and the aggregate number of riders. The design includes two main components: an embedded passenger counting system and a cloud-based web application. The counting system will contain a microcontroller, infrared sensors, and an LTE-M and GPS module to send data to the cloud service. Infrared sensor arrays will be installed by the bus doors to capture passengers entering and exiting. The microcontroller will process the sensor inputs and calculate the aggregate passenger count. The LTE-M module will transmit the current number of passengers and GPS coordinates of the bus to an AWS gateway as quickly as the network will allow.

The cloud-based web application will consist of a noSQL database to store time-stamped passenger counts and a web application that will display dashboards to end-users. AWS IoT will connect the cellular IoT device and the back-end noSQL database. A web dashboard will actively refresh with the latest passenger counts and bus locations from the database. Transportation departments and bus riders will analyze the passenger traffic statistics and bus maps on a web application that displays the dashboards.

In general, stakeholders are the people who will be affected by the project. They include people who have considerable influence over the project or those who are interested in its outcome. High power, highly interested stakeholders will engage the most with the project. The majority of the project effort will be to meet the goals of the project advisor and team members. High power, less interested stakeholders are heavily invested in the project's outcome but do not need to be notified of all the details. Capstone Design Expo judges and state and local governments will influence the project's success, and it is essential to keep them satisfied. Low power, highly interested stakeholders need to be informed about the project's progression and help with technical and marketing details. Regular communication with transportation departments, bus passengers, and competitors can help the project avoid significant problems. Low power, less interested stakeholders should be monitored occasionally, but regular communication is not necessary. It will be moderately important to consider the project's impact on the Georgia Tech Community, prospective customers, the media, and future recruits.

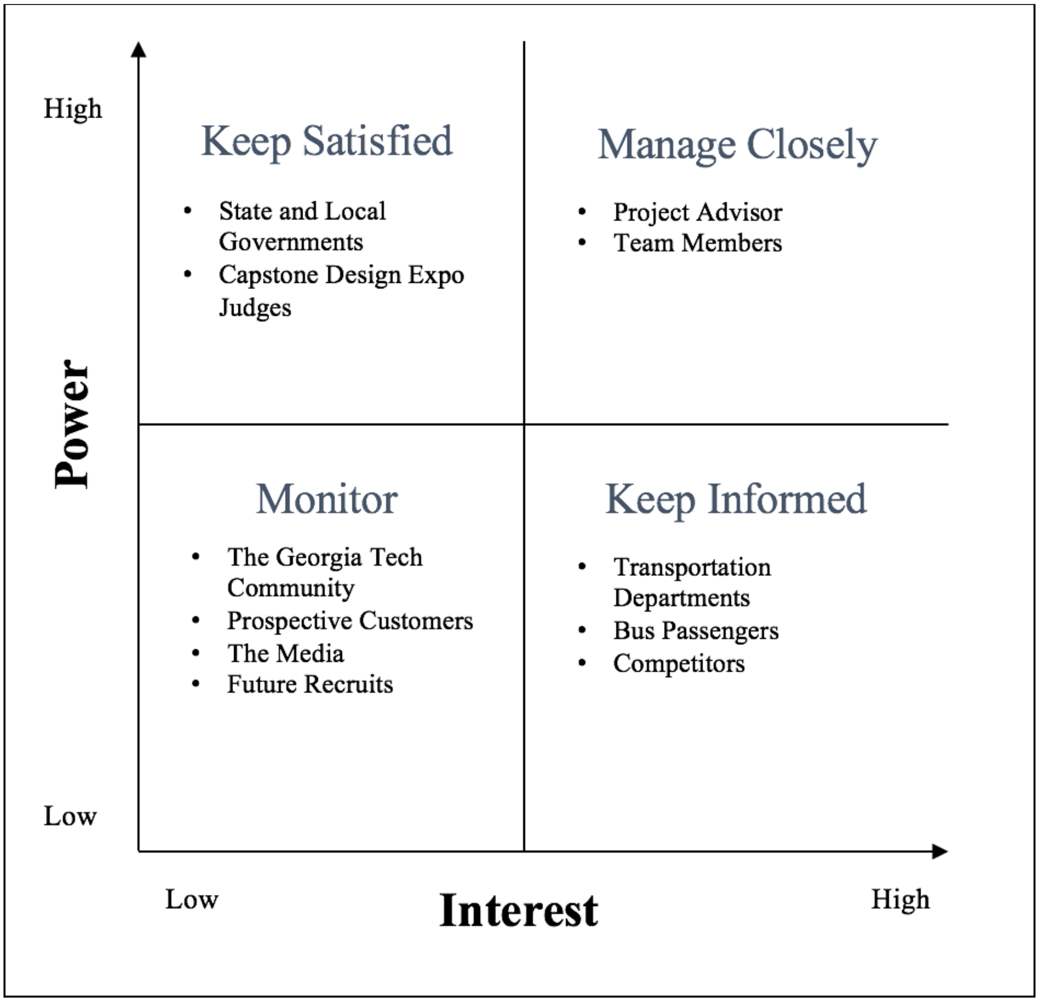


Figure 1. Stakeholder “2x2” Chart.

The target user of the product is transportation departments that provide bus services. The IoT device should use an existing bus power supply and be installed in under four hours. To be implemented across an entire bus fleet, the device should cost no more than $800. Passengers accessing the web application for aggregate rider counts and location services should receive updates within 5 s. The customer needs for the project include the following:

Customer Needs

- Accurate display of passenger counts and bus location
- Total cost less than $800
- Charging capability

- Setup IoT device and connect to cloud service in under 4 hours
- Fit inside the doorway of a bus
- Timely web application response

The product will fail if it cannot respond to user requests in a punctual fashion. There are two potential network bottlenecks within the proposed system. Between the IoT device and the cloud database, there should be a minimum end to end throughput of 5.6 kbps to allow enough time for data processing. Transportation departments and passengers should receive HTTP responses from the AWS virtual machine within 5 s. Beyond timing performance, the device should easily integrate into a modern bus infrastructure. The embedded system should charge using a 5 V USB cable and should not hinder passenger movement. To be scalable, the total cost of the cloud microservices and hardware should be less than $250, and the back-end should support up to 15 IoT device connections. The engineering requirements for the project include the following:

Engineering Requirements

- Transmit from the IoT device to the cloud with a minimum throughput of 5.6 kbps
- Embedded system cost under $200
- Respond to thousands of HTTP requests in less than 5 s
- Total weight should be less than 256 g
- Total area of the embedded system should be less than 680 cm^2
- Support up to 15 device connections per node
- Charge using a 5V USB power supply
- Analyze passenger count data on a cloud service for less than $50

The relationship between the customer needs and engineering requirements were mapped into the Quality Function Deployment (QFD) Chart in Appendix A.

The target customer for this project is transportation companies with hundreds of busses. The per-bus cost of the product must be kept within an acceptable threshold to make it economically viable. Due to the project's limited timeframe, the team has decided to use infrared sensors rather than image tracking software to monitor passenger movement. The use of cameras to identify passengers and their traveling habits leads to privacy concerns. Such data collection cannot be justified for this project; thus, pivoting towards infrared, non-identifying sensors helps achieve the project's goals without fear of litigation. The embedded processor in the cellular IoT development kit contains 256 KB of RAM [1]. The limited processor memory will relegate data analysis and visualization to the cloud service. The embedded processor will do little more than receive sensor inputs, package the data, and send it to the web app over the LTE-M network.

The project's computer networking standards impose additional timing constraints. The data transfer between the IoT device and the cloud system will be limited by the 200 kbps maximum uplink throughput of the LTE-M network [2]. User HTTP requests will be restricted by the response time and end to end throughput of the underlying TCP/IP infrastructure. The constraints for the project include the following:

Constraints

- Per-Bus cost
- Accuracy of sensors
- Privacy concerns
- Throughput between IoT device and cloud system
- HTTP response time
- Embedded processor RAM

3. Technical Specifications

The two major components of the project include the embedded passenger counter and the cloud-based web application. The calculation of the aggregate passenger count should occur in under 2 s to allow enough time for transmission to the AWS virtual machine. The relationship between the response time of the device and the customer requirements can be seen in the Quality Function Deployment (QFD) chart in Appendix A.

Table 1. Embedded Passenger Counter Specifications

Feature	Specification
Calculation of aggregate passenger count	2 s
Power Supply	5 V USB
Dimensions	680 cm^2
Sensor field of view	120 degrees
LTE-M Connectivity	700 MHz -2.2 GHz LTE Carrier frequency
GPS Connectivity	Dedicated GPS antenna within IoT device
Embedded Processor RAM	256 KB

Table 2. Cloud-based Web Application Specifications

Feature	Specification
Model Size	10 GB
Live Map Update Frequency	5 seconds
Num. Supported Devices	15 connections/node

Analytics Dashboard Update Frequency	5 minutes

4. Design Approach and Details

4.1 Design Concept Ideation, Constraints, Alternatives, and Tradeoffs

System Overview

There are two main components to IntelliBus: the IoT embedded device and the web applications. The IoT device consists of a microcontroller, tracking sensors, and LTE-M module. The web application is responsible for providing a real-time dashboard of bus locations, passenger counts, and viewing pertinent graphical statistics and analytics. Figure 2 shows the overall system architecture of IntelliBus:

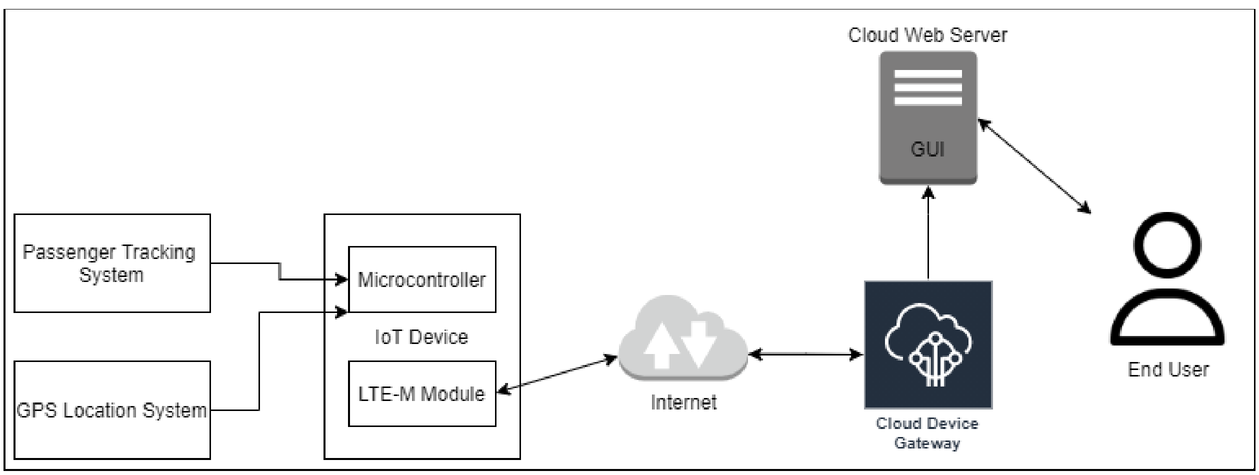


Figure 2. The overall system architecture of IntelliBus.

Passenger Tracking

Passenger tracking provides the count of the number of passengers on a bus. Passenger counts can be acquired by tracking when passengers exit and enter the vehicle. The embedded software program will calculate the aggregate number of passengers based on sensor inputs.

A pair of Infrared sensors will be used to track passengers as they enter and exit the bus. The Infrared sensor will connect to the microcontroller, which will handle incrementing or decrementing passenger counts. The order of when the sensors are triggered will indicate if a person is entering or exiting the bus, as shown in Figure 3.

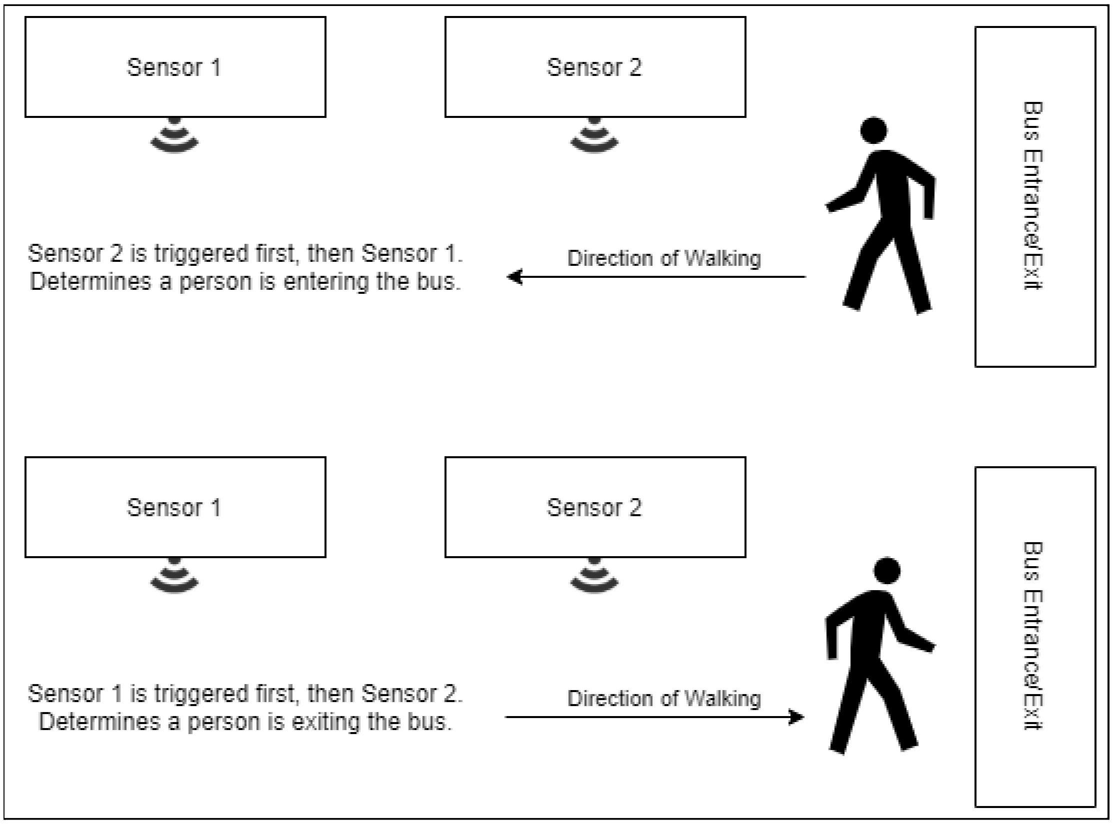


Figure 3. Infrared sensors can determine if a passenger is entering or exiting the bus.

Infrared sensors were chosen for their low cost, low power, and ease of setup at the expense of high accuracy. Other sensor devices that were considered:

- Pressure Sensors were considered for their low cost and high accuracy. Pressure sensors work similarly to infrared sensors to figure out if a person entered or exited a bus based on which pressure sensor was triggered first by footsteps. However, pressure sensors did not meet the constraints for ease of setup and compactness. Pressure sensors require more wiring overhead and higher maintenance cost due to wear and tear from passenger footsteps.
- Camera Sensors: A video-based human recognition model was considered for its high accuracy and ease of setup since existing camera hardware exists in most buses for security purposes.

However, ethical implications plague this solution for privacy concerns of running video feeds through Artificial Intelligence recognition models.

Location Service

Location services will provide the real-time position of the bus through latitude and longitude coordinates. The bus location will be sent over the internet to the cloud services to give the end-users real-time maps of the bus location and provide location-based data points for analysis.

A GPS module will be used to gather the real-time location of the bus. The GPS module will connect to the IoT device microcontroller using serial communication to pass the latitude and longitude position as needed. GPS modules are highly accurate, low cost, low power, and easy to set up modules. No alternatives were considered as GPS is the de facto method for gathering location positions through satellite.

Internet Connectivity and Communication

Internet connectivity allows pertinent data gathered from the passenger tracking system and GPS system to be sent to the cloud services to process, view, store, and analyze real-time passenger counts and bus locations. Utilizing standard MQTT or HTTP protocols [3,4], data can be sent over multiple mediums of communication.

An LTE-M module will be utilized for bi-directional communication to the cloud device gateway with passenger count data and GPS location. The LTE-M module will connect to the IoT device microcontroller using serial communication to pass data it receives and transmits. The module will have its own microcontroller and firmware to control radio communications to the internet and provides a level of abstraction by communicating with the IoT device microcontroller through serial commands.

LTE-M communication was considered for its high accuracy and high reliability but at a higher cost. The LTE-M module can provide consistent connection and range to the internet due to 99.9% coverage of at least a 3G network in the United States [5]. Consistent internet connections are critical for an IoT device that produces real-time sensor data that needs to be processed. WiFi was considered as an alternative wireless network medium because of its low cost. However, WiFi will restrict the IoT device's mobility. A WiFi-enabled device must stay within a fixed range of a base station to maintain a consistent internet connection.

IoT Device Microcontroller

The IoT Device will require a RTOS microcontroller board to facilitate central processing of all data to and from GPIO peripherals. The microcontroller contains software programs to receive data from the passenger tracking system and GPS module, which it will use to send data (and receive data) from the cloud device gateway through the LTE-M module. The microcontroller will utilize standard GPIO signals and serial communications to gather input and output from I/O devices.

Device Gateway

Device gateways provide a way to consume IoT device data to the cloud for collection, analysis, storage, and near real-time updates to end-users. Device Gateways maintain long-lived, bidirectional connections, enabling IoT devices to send and receive messages at any time with low latency. Device gateways act as the integrator for communication between end-users and IoT devices. Gateways use HTTP or MQTT to receive and send data from IoT devices and allow for a standard to process payloads sent through the internet.

Cloud IoT services will be used as the device gateway. All cloud providers have IoT services to handle tracking, communication, monitoring, and management of IoT devices in the field. The cloud IoT service also provides an easy way to connect to other cloud services for web interfaces, database storage and data analysis. Cloud IoT services also have extensive Software Development Kits (SDK), management and networking dashboards that allow for rapid deployment and management of IoT devices and their data.

Cloud IoT services were chosen as the device gateway for their low cost, ease of setup, and ease of use. The cloud provides extensive savings by not requiring large initial CapEx investments and allowing OpEx spending by paying only how much compute resources are used. Cloud services are easy to set up from their web console and SDKs. They have extensive documentation and support for industry standards with no significant trade-offs for this project's implementation. All cloud services can be easily integrated and scaled to meet the demand for resources. With 99% uptime [6], redundancy, and scalability, cloud services were preferred over dedicated servers. Servers require maintenance and reduce agility in the development lifecycle and have a significant upfront investment.

Graphical User Interface (GUI)

The GUI allows intuitive interactions for users to view and perceive data requested. The GUI serves as the interface for passenger end-users to view live maps of the bus location and its passenger count. Transportation department end-users can view statistical and graphical dashboards to analyze useful data from bus routes and passengers. Analytics GUI will include bar graphs for daily overall passenger counts, line graphs of passenger counts by time intervals and graphs of machine learning models of peak travel times.

A web interface will be utilized to present users with the GUI. Cloud web services will be used to host the web application and easily integrate with the IoT service on the same cloud provider. The web interface can utilize APIs to communicate with IoT services and provide an accurate and safe method to present users with the IoT device data.

Hosting the web application on the cloud provides the same benefits as mentioned for cloud IoT services. Web applications also provide ease of use from any web-enabled device. Cloud services scale as the demand from users increases. Scalability is achieved by ramping up web and IoT services during peak times and ramping down at lower usage times to save money.

4.2 Preliminary Concept Selection and Justification

Passenger Detection System

An array of PIR sensors is the optimal choice for implementing the Passenger Detection System on IntelliBus due to its low cost, ease of setup, and ability to identify the direction of human traffic flow. The placement of PIR sensors is designed as shown in the following figure.

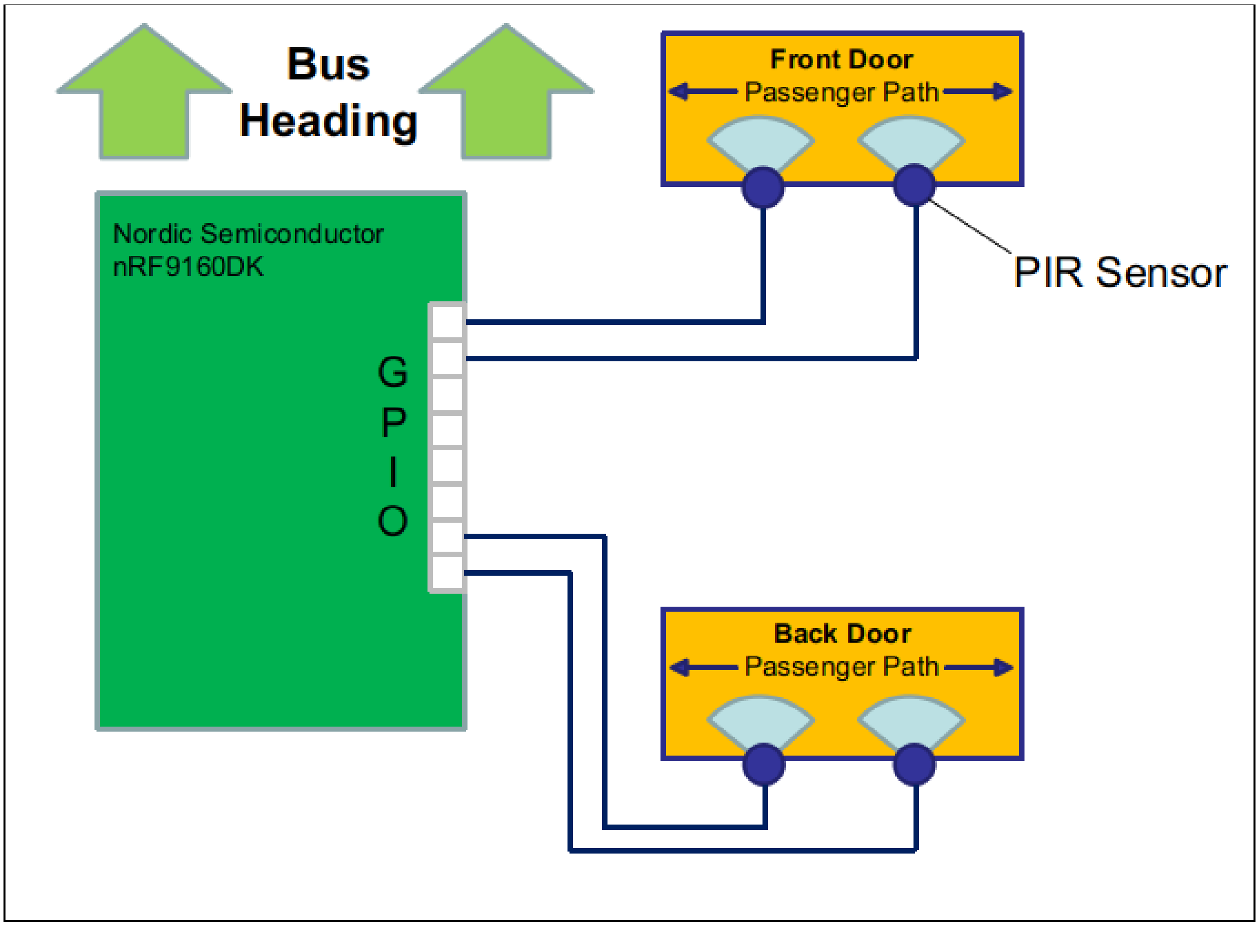


Figure 4. Layout of PIR Sensors and connections to IoT cellular Device.

Two sensor arrays, each consisting of two individual PIR sensors, will be installed on the bus's front and back doors. The PIR sensors will face perpendicular to the motion of passengers. When a PIR sensor detects a human presence in its range, it will send a logically high signal to the GPIO pin on the microcontroller. Using the timing difference on the two PIR sensors in a sensor array, the program will know which sensor is triggered first and distinguish between a person getting on and getting off the bus. With this setup, the team can develop a software program for keeping a count of passengers on board, and IntelliBus will work for buses whose doors are used for both entering and exiting.

The Nordic Semiconductor nRF9160DK

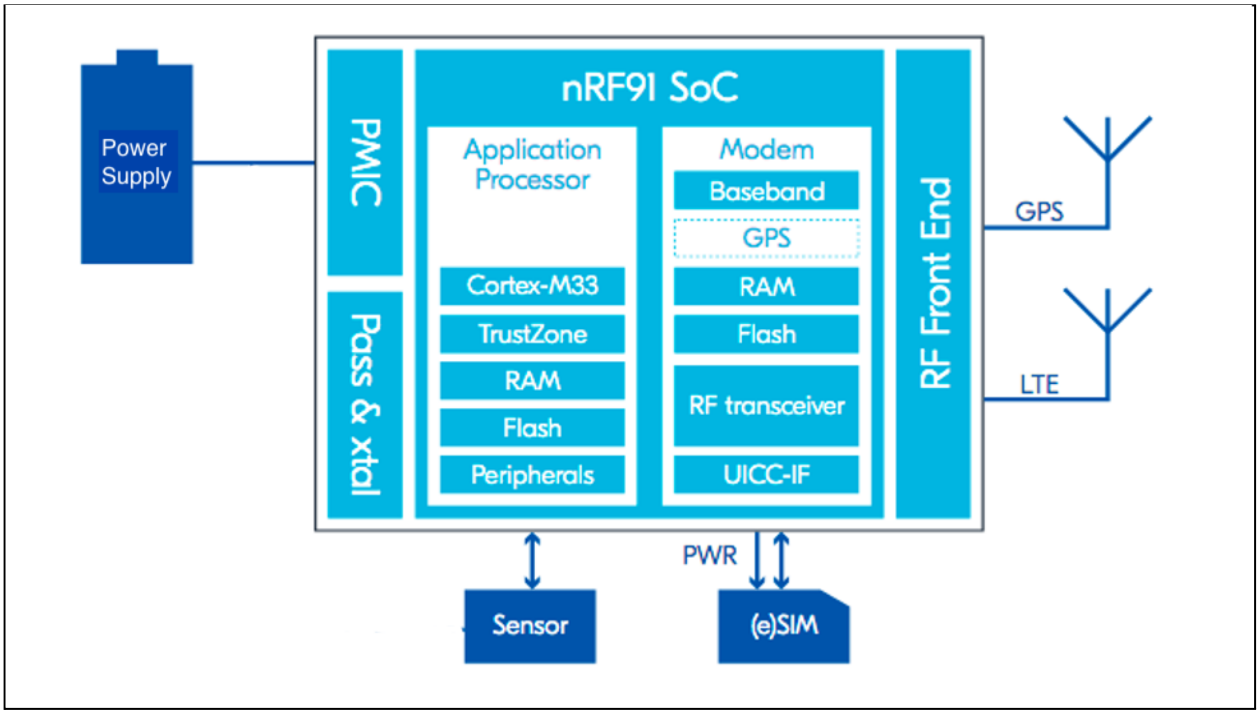


Figure 5. nRF9160 DK cellular IoT development kit with dedicated LTE and GPS antennas [7].

The DK cellular device is the optimal choice for the project because of its built-in LTE and GPS capabilities, application processor, and 24 GPIOs. The LTE antenna will be optimized for global bus operations supporting all LTE frequency bands in the 698-960 MHz and 1710-2200 MHz ranges. For ease of development, the nRF9160 comes bundled with an iBasis SIM card preloaded with 10 MB [7]. Other commercial cellular IoT kits do not come with GPS support or data SIM cards and require additional purchases. The device features a 3.0-5.5 V power supply from an external battery but can also be powered using 5 V USB. The application processor includes a 64 MHz Arm Cortex-M33 CPU with 1 MB of flash and 256 KB of RAM dedicated for the application, which are adequate hardware resources for the passenger counting program. The 24 GPIOs can be configured to interface with the infrared sensor arrays [1].

AWS Microservices

Amazon's cloud services, AWS, is the ideal choice for the project because it provisions and manages an elaborate ecosystem to host all of our needed computational resources. The AWS web dashboard and SDK provide an easy way to develop and manage all applications. AWS takes advantage of a microservice architecture and promotes development in modular parts. These microservices can be interconnected through APIs to make IntelliBus's GUI. The overall AWS service architecture for this project can be seen in Figure 6.

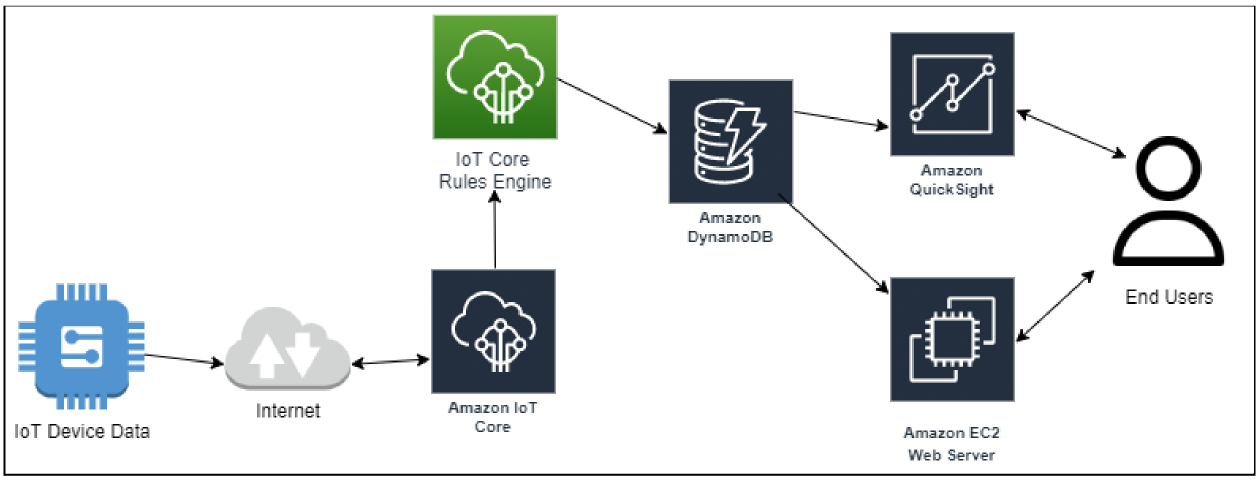


Figure 6. The AWS Architecture of IntelliBus.

AWS IoT Core service is the chosen device gateway for the IoT device connection. The AWS IoT Core provides a Rules Engine [8], software to process incoming IoT device data and forward it to Amazon DynamoDB. DynamoDB is a noSQL database that will store incoming device data in tables [9]. The DynamoDB table is then accessible to the AWS Elastic Compute Cloud (EC2) virtual machine. The EC2 instance is dedicated hardware to host the web application for the GUI [10]. The web application pulls from the database to present the end-users with live maps and passengers counts.

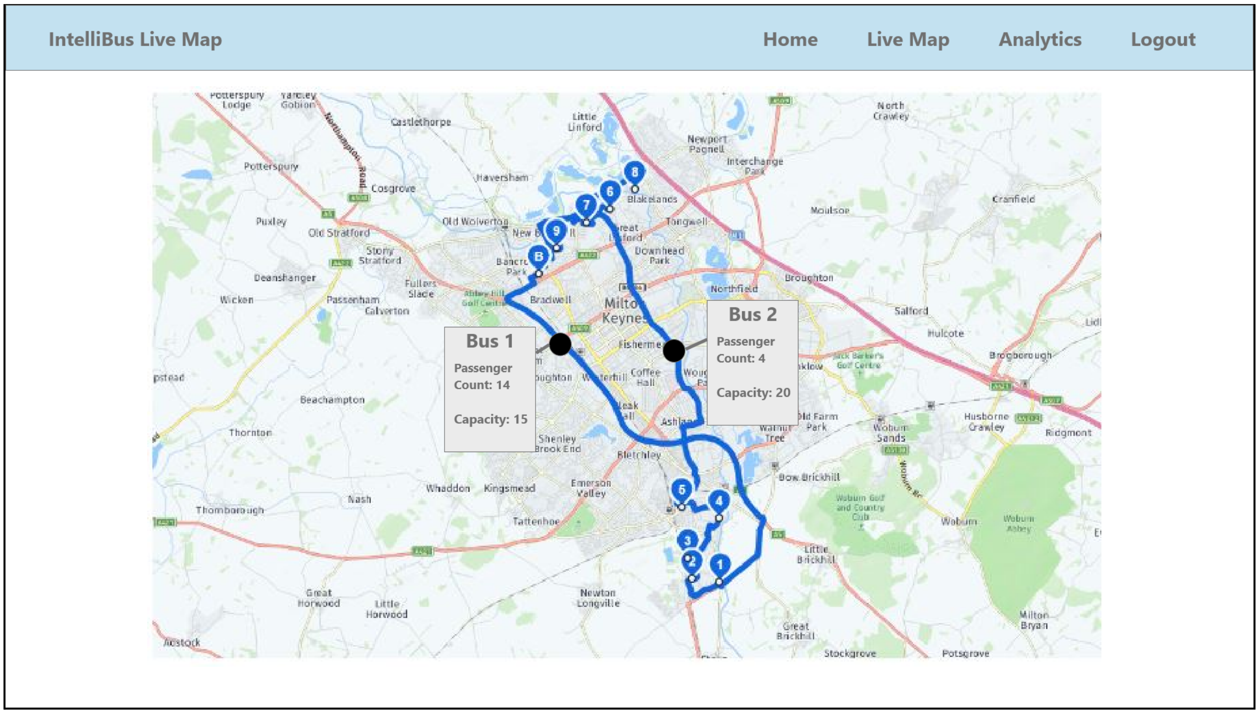


Figure 7. The proposed GUI for the live map and passenger count.

The DynamoDB table also connects AWS Quicksight to provide web-embeddable machine learning business intelligence (BI) GUI dashboards. Quicksight allows for deeper insights using machine learning (ML) to perform advanced analysis on IoT devices stored in the database [11].

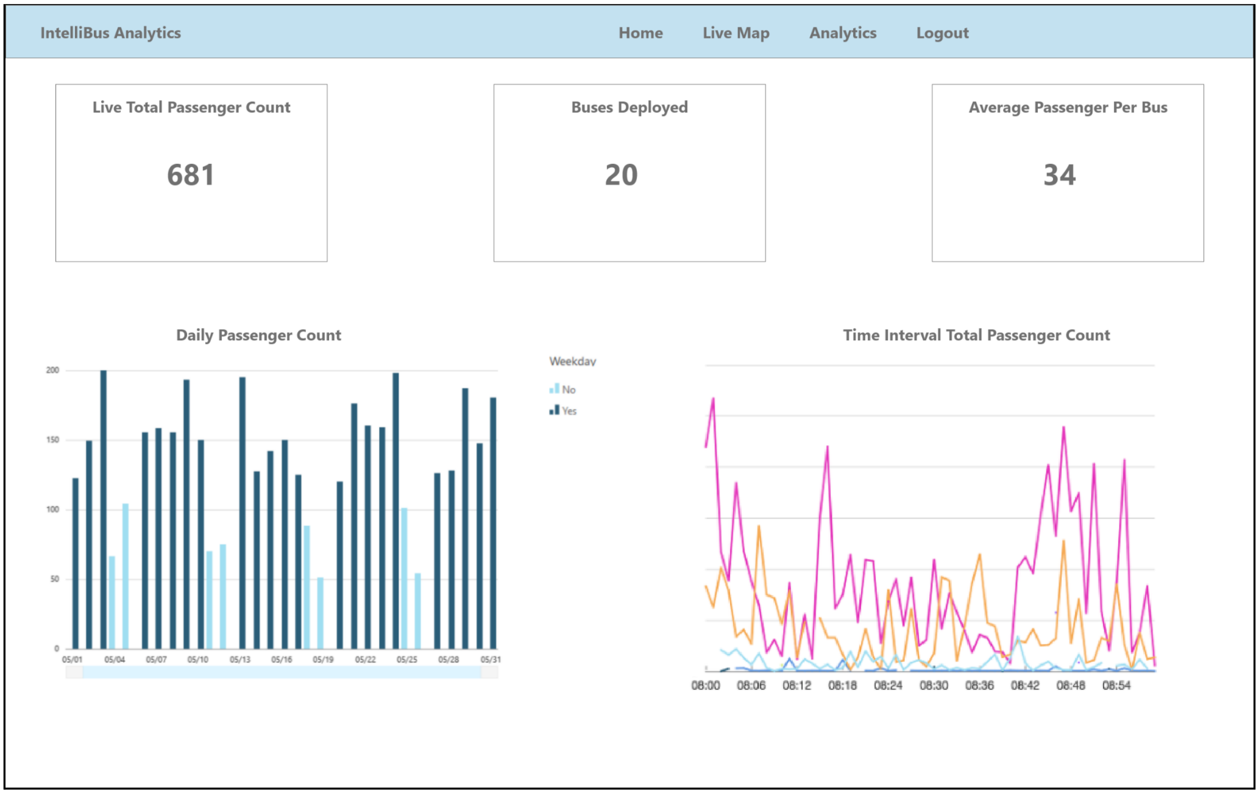


Figure 8. The proposed Quicksight GUI dashboard.

Critical Path

The technical critical path for this project involves setting up cloud services, including the web app, servers and database, setting up the IoT hub and management software with the project's various devices, and developing and testing integrations between the web app and analytical software.

These key technical issues will be addressed early on in the design process. Integration between the various PIR sensors and DK cellular devices with the IoT hub and cloud services will not happen in isolation. Instead, any shortcomings in technical design will be discovered as soon as both hardware and backend services are configured. Device connectivity and IoT communication issues, for example, would have to be addressed after only a few weeks into the project to provide ample time to strategize and pivot towards a solution.

Contingency Plan

The team anticipates a low probability of failure in the project's various subsystems, as seen in Table 3.

Table 3. Contingency Plan

Risk	Probability	Response
PIR sensors comprise	Low	Resort to other options for motion sensing that may come at a higher cost, for example an infrared source and receiver solution.
LTE-M configuration failure	Low	Utilize WiFi modules to connect to AWS IoT core and send/receive data over the internet.
AWS Downtime	Very Low	Utilize Azure IoT and web services as the next best alternative.
IoT Device failure	Low	Contact Nordic Semiconductor with technical questions and inquire about receiving a refund or new device.

4.3 Engineering Analyses and Experiment

4.3.1 Prototype Testing and Analysis

Prototype testing will be accomplished using unit testing, system integration testing, and user acceptance testing. These testing methods are standard practices for product and software testing and will provide a framework to validate this design.

4.3.1.1 Unit Testing

Unit testing will be used to verify all software aspects of IntelliBus. This method will ensure that software development is broken into modules and that each module is verified and tested for project functionality and constraints. IoT device unit testing will be used to verify individual I/O devices are sending and receiving data to the microcontroller as expected.

Web application unit testing will involve dependency injections to validate individual software components independent of their external dependencies to other web applications parts or APIs.

Dependency injections allow for isolation code and test if its data control and manipulation are as expected.

4.3.1.2 System Integration Testing

System integration will be essential in testing that all subsystems communicate with each other as expected. System integration can be broken into two aspects: hardware integration testing and software integration testing.

Hardware integration testing ensures that all I/O devices have the physical capabilities to communicate with the microcontroller. Passenger tracking integration can be tested and verified by ensuring GPIO signals are triggered when a person walks past it and is sent to the microcontroller with logic analyzers. GPS and LTE-M module integration can be tested with serial communication debuggers to ensure data is transferred to the microcontroller.

Software integration testing ensures IoT devices, cloud gateway, and web applications are able to function together to receive and share data. Software integration testing verifies data is being sent from the IoT device through MQTT and received by the cloud gateway. LTE-M module's SDK provides console debug options to verify connections using network tests which provide network download, upload, and latency (ping). LTE-M configurations will be tested to ensure it has upload speeds greater than 5.6 kbps (Appendix A).

AWS CloudWatch service provides debugging and event listening to ensure network packets are being delivered to the IoT Core's gateway. Event listening will verify that IoT device data is transmitted and received over the internet to and from AWS. AWS web application integration testing can be done by ensuring API responses from DynamoDB and QuickSight using AWS CloudWatch for network analysis.

4.3.1.3 User Acceptance Testing (UAT)

UAT is when actual users test if the software application can carry out its required tasks as it was designed for end-users. The UAT test will adhere to the attributes of the design requirements and interview end-users to gather feedback. This feedback will improve and validate the product until all customer requirements are satisfied (Appendix A).

UAT will test all web application aspects from two groups of end-user types: average public transportation passengers and transportation department management. Regular public transportation passengers will test the live map and passenger counts to ensure that the web GUI is intuitive and the map interface is accessible, convenient and error-free. The analytics dashboard GUI will be tested by transportation department employees to validate that the data presented is customizable, practical, and contextual.

4.4 Codes and Standards

The most significant codes and standards that apply to the IntelliBus project include the following:

1. Universal Serial Bus (USB) is an omnipresent power socket for cell phones, MP3 players, and microcontroller boards. Features of USB power delivery include maximum power of 100W, bidirectional charging capability, power management across multiple peripherals, and low-power device optimizations [12]. During prototyping, the IoT development board will connect to a PC for power and data delivery over USB.
2. Message Queuing Telemetry Transport (MQTT) and Hypertext Transfer Protocol (HTTP). MQTT is an ISO/IEC 20922:2016 standard that leverages TCP/IP (or other network transport protocols) to provide lightweight, open and simple publish/subscribe messaging transport [3]. HTTP is an RFC 2616 consensus standard issued by the Internet Engineering Task Force (IETF) in their published memorandum called Request for Comments (RFC). HTTP provides generic and stateless hypermedia information for virtually any digital data [4]. MQTT and HTTP are widely adopted and allow for rapid development and deployment to send and receive

data from IoT devices. MQTT and HTTP allow for bidirectional data transfer between IoT devices, Cloud Gateways, Cloud Services, and end-users.

3. I2C is a prevalent serial communication protocol for embedded devices. It is highly scalable and low-power. The two ports, SDA and SCL, are for clock and data. First, the address of the slave is sent on the bus. Then, data on the I2C bus is transferred in 8-bit packets (bytes), and each byte is followed by an acknowledge bit. This bit signals whether the device is ready to proceed with the next byte. With an address-based data transfer scheme, one I2C master on the microcontroller can connect to more than 100 I2C slaves. For the project, the GPS module can be an I2C slave to the main microcontroller, and communication between them must follow the I2C protocol [13].
4. LTE-MTC low power wide area (LPWA), or otherwise known as LTE-M, is a technology that leverages existing mobile networks to provide long range, low power IoT connectivity [14]. Current LTE-M technology is developed privately by businesses attempting to capitalize on the growing IoT market. Standard organization 3GPP specifies a 1.08 MHz bandwidth and 1Mbps peak download/upload rate for LTE-M [15]. Mainstream cellular providers such as AT&T and Verizon offer competitive rates for LTE-M connectivity. During prototyping, we will connect the infrared sensors to the LTE-M network and send data wherever the device is located.

5. Project Demonstration

The team plans to contact the PTS department, Georgia Tech's bus service provider, for demonstration assistance and acquiring a Stingerette bus. With approval from PTS, the team will install the required hardware components onto the bus and configure the LTE module to connect to the backend services. In the best-case scenario, the team expects the bus to complete a normal service around campus, and demonstrate that the sensor components can measure the number of passengers on board with an acceptable accuracy and transmit the data via LTE-M network to the backend. Meanwhile, on the web application side, the team will demonstrate that data collected from the hardware components installed on the bus can be successfully received and visualized. Additionally, AWS Cloud Watch can help analyze the throughput of data and check for errors in the transmitting process.

6. Schedule, Tasks, and Milestones

The IntelliBus team will be designing and implementing this prototype during the fall 2021 semester. Appendix B contains the list of project tasks, the person assigned, and the risk level associated with each task. Tasks in which there is a large amount of experience among team members are deemed Low Risk. Team members have limited background knowledge of High Risk tasks, and these components may take up most of the labor budget. Appendix C contains a GANTT chart for the project showing the start date, finish date, and predecessors for completion. Appendix D shows the PERT chart for the project. Each edge of the PERT chart is labeled with an optimistic time, most likely time, and pessimistic time for success.

The critical path for the project was determined using the PERT chart in Appendix D. The critical path is the route from the start node to the finish node with the longest expected duration. Using the expected time calculations in Appendix B, the length of the critical path ABHTUWXZ is 20.6267 weeks. Based on the PERT chart analysis, the probability that the team will complete the project one week before the GT Capstone Expo can be found using the following Z statistic:

$$Z_s = (T_s - T_e)/\sigma_T$$

T_s is the time associated with finishing the project one week before the GT Capstone Expo. It includes the time spent working on documentation and planning in ECE 4871 (1/14/21 – 04/16/21) plus the time during the fall semester one week before the Design Expo (08/23/21 – 11/30/21). Therefore, $T_s = 13.14 + 14.14 = 27.28$weeks. T_e is the duration of the critical path found using the expected time calculations in Appendix B. σ_T is the standard deviation of the critical path found from Appendix B.

$$Z_s = (T_s - T_e)/\sigma_T = (27.28 - 20.6267)/1.8322177 = 3.613130208$$

Using a Z table, $P(Z < 3.613130208) = 0.99985$. There is a 99.985% probability that the team will complete the project one week before the GT Capstone Expo.

7. Marketing and Cost Analysis

7.1 Marketing Analysis

There are virtually no existing commercial products designed to count the number of people on a moving bus. However, the idea has been proposed in a recent research paper where the number of passengers on a public bus was tracked using infrared sensors, and the location of the bus could be viewed on an Android application [16]. For comparison to the larger marketplace, the proposed product can be categorized as a people counting system. People counting systems include any device that tracks the number of individuals that move through a particular passage or entrance over a fixed time. The global market size of people counting systems was $730.1 million in 2018 and is projected to reach $1,491.3 million by 2026 [17]. The majority of counters are targeted at the retail industry. The Bi-Directional People Counter by Immotion gathers data from more than 1000 infrared sensors to track the preferences and behaviors of customers on the floor of a department store. The proposed product provides a web application where users can access counts and maps of bus routes. In contrast, the Immotion counter leaves the software analysis up to the user [18].

Similar products also exist in corporate settings. The Monnit Wi-Fi Infrared Motion Sensors use passive infrared technology to track the occupancy of desks, conference rooms, and hallways [19]. The device connects to a Wi-Fi network and communicates to the iMonnit Online Sensor Monitoring and Notification System, where sensor data can be reviewed and exported. Although the product in [19] provides customers with a secure online platform to analyze motion sensor readings, it would not be helpful inside of a bus because it must stay within range of a Wi-Fi access point. Thus, the proposed new product is differentiated by its mobility and the application software included in the purchase.

7.2 Cost Analysis

The equipment for a prototype of the IntelliBus system includes the IoT development kit, infrared sensors to track passenger motion, and additional wires and connectors. The Nordic Semiconductor cellular IoT development kit is the most expensive item in the project budget at $139.00 [7]. PIR sensors come in batches of 5, and the per batch price cost is $12 [20]. The costs of wires and connectors are estimated to be around $10. The embedded software development environment nRF Connect SDK is included with the IoT development kit purchase [7]. The prototype will not need more than 10GB of standard storage and will use Amazon Web Services' free-tier [21]. The total equipment cost for the prototype is approximately $161.00.

Table 4. Equipment Costs

Product Description	**Quantity**	**Unit Price ($)**	**Total Price ($)**
nRF9160 DK Cellular IoT Development Kit	1	$139.00	$139.00
Infrared Sensors	1 Batch of 5	$12.00	$12.00
Wires and connectors	1	$10.00	$10.00

nRF Connect SDK	1	$0.00	$0.00
Amazon Web Services	1	$0.00	$0.00
Total Cost			**$161.00**

The Development costs in Table 2. were calculated assuming a typical engineer's salary of $45.00 per hour [22]. Back-end code development is a particularly high-risk task and will take the most significant amount of labor hours.

Table 5. Development Costs

Project Component	**Labor Hour**	**Labor Cost**	**Part Cost ($)**	**Total Component Cost ($)**
Embedded System				
Assembly	8	$360.00	$161.00	$521.00
Code Development	50	$2,250.00		$2,250.00
Quality Assurance	50	$2,250.00		$2,250.00
Front-End Development				
Code Development	172	$7,740.00		$7,740.00
Quality Assurance	55	$2,475.00		$2,475.00
Back-End Development				
Code Development	212	$9,540.00		$9,540.00
Quality Assurance	60	$2,700.00		$2,700.00
Demo	50	$2,250.00		$2,250.00

Preparation				
Group Meetings	30	$1,350.00		$1,350.00
Total Labor Cost	687	$30,915.00		$30,915.00
Total Part Cost			$161.00	
Total Cost (Labor + Part)				**$31,076.00**

Using a fringe benefit of 30% of total labor cost and an overhead of 65% of total materials, labor cost, and fringe, the total development cost of the prototype is $66,578.33 as seen in Table 3. The overhead cost includes factors indirectly tied to the product's production, such as rent and utilities.

Table 6. Total Development Cost

Parts	$161.00
Labor	$30,915.00
Fringe Benefits % of Labor	$9,274.50
Subtotal	$40,350.5
Overhead, % of Material, Labor & Fringe	$26,227.83
Total	$66,578.33

The production will consist of 100 units sold over five years at $778.27 per unit. The cellular IoT kit cannot be purchased at a bulk discount, and the cost for each unit is $139.00 [7]. The wires and sensors can be purchased at a discounted price of $8 and $10, respectively. The cloud services will cost $20 per unit. Technicians employed at $20 per hour will assemble the embedded system on the bus infrastructure in two hours. A technician will spend one hour testing the device, and a software engineer will need an additional hour to assess and verify the product's software. Advertising to local transportation departments will make up 6 % of the total sales. Assuming the product has a residual value of zero dollars and a total lifespan of twenty years, the annual fixed amortization over all input costs is $25.86. For 100 units sold at $778.27, the total revenue and total profit of the production are $77,827 and $8,500. The expected profit per unit is $85 yielding a 10.9% payback.

Table 7. Selling Price and Profit Per Unit (Based on 100 unit production)

Parts Cost	$139.00 (cellular IoT kit) + $10 (sensors) + $8 (wires) + $20 (cloud services) = $177
Assembly Labor	$40.00
Testing Labor	$65.00
Total Labor	$105.00
Fringe Benefits, % of Labor	$31.50
Subtotal	$313.50
Overhead, % of Material, Labor & Fringe	$203.78
Subtotal, Input Costs	$517.28
Sales Expense	$46.69
Amortized Development Costs	$129.30
Subtotal, All Costs	$693.27
Profit (Per Unit)	$85
Percent Profit (Per Unit)	10.9%
Selling Price	$778.27

8. Current Status

From the table of tasks in Appendix B, the team is 91% complete with the planning and documentation phase of the project. Sub-tasks A through J are 100% complete; the only outstanding item for spring 2021 is to submit the remaining two meeting minutes with Dr. Madisetti. In August of 2021, Thomas and David will start the embedded system setup and integration task by ordering the cellular IoT kit and infrared sensors. Simultaneously, Shadman and Noah will begin the back-end and front-end development of the web application. Overall, the project is currently 36% complete with sub-tasks L through β not yet started.

9. Leadership Roles

Shadman Ahmed will serve as Overall Team Coordinator and Backend Software Lead. Noah Chong will serve as Frontend Software Lead, Expo Coordinator, and Webmaster. Thomas Talbot will serve as Embedded Systems Co-Lead and Documentation Lead. Yue Pan will serve as Embedded Systems Co-Lead and Testing Lead. Leadership roles are not permanently assigned, and team members may be called upon to assist in other areas of the project depending on the workload.

10. References

[1] Nordic Semiconductor, “Development kit for LTE-M/NB-IoT/GPS/Bluetooth Low Energy”, nRF9160 DK datasheet, Oct. 2018, [Revised Nov. 2020].

[2] B. E. Benhiba, A. A. Madi, and A. Addaim, "Comparative Study of The Various new Cellular IoT Technologies," in Proc. 2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra, 2018, pp. 1-4, doi: 10.1109/ICECOCS.2018.8610508.

[3] *Information technology – Message Queuing Telemetry Transport (MQTT)*, ISO Standard ISO/IEC 20922:2016, June 2016. [Online]. Available: https://iso.org/standard/69466.html

[4] *Hyper Text Transfer Protocol – HTTP/1.1*, RFC 2616, June 1999. [Online]. Available: https://ietf.org/rfc/rfc2616.txt

[5] “Goal 9: Build RESILIENT infrastructure, promote inclusive and sustainable industrialization and foster innovation,” Sep-2017. [Online]. Available: https://sdg.data.gov/9-c-1/. [Accessed: 12-Apr-2021].

[6] “Amazon Compute Service Level Agreement,” *Amazon Web Services*. [Online]. Available: https://aws.amazon.com/compute/sla/. [Accessed: 12-Apr-2021].

[7] “nRF9160 DK,” *Nordic Semiconductor*, 2020. [Online]. Available: https://www.nordicsemi.com/Software-and-tools/Development-Kits/nRF9160-DK. [Accessed: 14-Feb-2021].

[8] “AWS IoT Core,” *AWS IoT Core Overview*. [Online]. Available: https://aws.amazon.com/iot-core/. [Accessed: 12-Apr-2021].

[9] “Amazon DynamoDB,” *Amazon Dynamo DB | NoSQL Key Value Database*. [Online]. Available: https://aws.amazon.com/dynamodb/. [Accessed: 12-Apr-2021].

[10] *Create an EC2 instance and install a web server*, 2021. [Online]. Available: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_Tutorials.WebServerDB.CreateWebServer.html. [Accessed: 12-Apr-2021].

[11] R. Nadipalli, "Effective business intelligence with QuickSight: from data to actionable business insights using Amazon QuickSight!," *Amazon*, 2017. [Online]. Available: https://aws.amazon.com/quicksight/. [Accessed: 12-Apr-2021].

[12] "USB Charger (USB Power Delivery)," *USB*. [Online]. Available: https://usb.org/usb-charger-pd. [Accessed: 16-Mar-2021].

[13] "I2C Bus Specification," *I2C Info – I2C Bus, Interface and Protocol*. [Online]. Available: https://i2c.info/i2c-bus-specification. [Accessed: 26-Mar-2021].

[14] "Long Term Evolution for Machines: LTE-M," *Internet of Things*, 19-May-2020. [Online]. Available: https://www.gsma.com/iot/long-term-evolution-machine-type-communication-lte-mtc-cat-m1/#:~:text=LTE%2DM%20is%20the%20simplified,CatM1%2C%20suitable%20for%20the%20IoT

[15] K. Flynn, "A Global Partnership," *Standards for the IoT*. [Online]. Available: https://www.3gpp.org/news-events/1805-iot_r14. [Accessed: 25-Mar-2021].

[16] B. K. Harini, A. Parkavi, M. Supriya, B. C. Kruthika, and K. M. Navya, "Increasing Efficient Usage of Real-Time Public Transportation Using IOT, Cloud and Customized Mobile App," *SN Computer Science*, 09-May-2020. [Online]. Available: https://link.springer.com/article/10.1007/s42979-020-00161-8. [Accessed: 01-Apr-2021].

[17] "People Counting System Market Size, Share, Trends And Forecast," *Verified Market Research*, 03-Dec-2020. [Online]. Available: https://www.verifiedmarketresearch.com/product/people-counting-system-market/. [Accessed: 31-Mar-2021].

[18] *Counting Solutions*, 01-Jul-2017. [Online]. Available: https://www.im-motion.com/counting-solutions/. [Accessed: 15-Mar-2021].

[19] Monnit, “Monnit Wi-Fi Infrared Motion Sensor”, MS-IR-WDS-01 datasheet, 2016

[20] *DAOKI 5Pcs IR Human Sensor AM312 Mini Detector Module* . [Online]. Available: https://www.amazon.com/DAOKI-Detector-HC-SR312-Pyroelectric-Automatic/dp/B07XF62F9J. [Accessed: 26-Mar-2021].

[21] “Amazon QuickSight Pricing” *Amazon*, 2021. [Online]. Available: Amazon QuickSight Pricing - Business Intelligence Service - Amazon Web Services. [Accessed: 2-Apr-2021].

[22] “Engineers: Employment, pay, and outlook : Career Outlook,” *U.S. Bureau of Labor Statistics*, 2016. [Online]. Available: https://www.bls.gov/careeroutlook/2018/article/engineers.htm#:~:text=The%20U.S.%20Bureau%20of%20Labor,median%20wage%20for%20all%20workers. [Accessed: 20-Mar-2021].

Appendix A - Project QFD Chart

Customer Requirements	Throughput	Hardware Cost	Response Time	Weight	Device Area	Scalability	Power Supply Voltage	Cloud Computing	Singapore Land Transport Authority	Beijing Transport
	Engineering Requirements								Benchmarks	
Accuracy	X		X			X		X	O	O
Low Cost		X				X			O	O
Charging Capability		X					X			
Compact				X	X				O	
Ease of Setup				X		X		X	O	O
Ease of Use						X		X	O	O
Analysis of Data Collected	X		X					X	O	O
Units	kbps	$	s	g	cm^2	c/n	V	$		
Engineering Targets	5.6	200	5	256	680	15	5	50		

c/n = conection per node

Appendix B - Tasks, Persons Assigned, and Risk Level

The expected time of a task was found using the following formula:

$t_e = (t_o+4t_m+t_p)/6$

The standard deviation of a task was found using the following formula:

$\sigma=(t_p-t_o)/6$

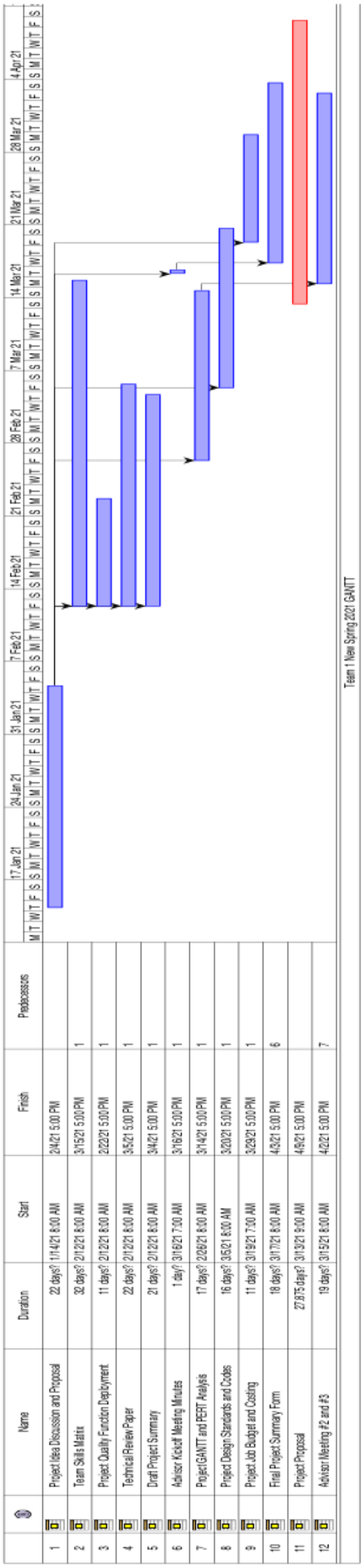
Letter	Task	Predecessor	Task Lead	Risk Level	t_o	t_m	t_p	Expected Time	Standard Deviation
A-K	**Planning and Documentation (ECE 4871)**	Start	All	Low					
A	Project Idea Discussion and Proposal	Start	All	Low	2	3	4	3	0.3333
B	Technical Review Paper	A	All	Low	1	2	3	2	0.3333
C	Project QFD	A	All	Low	1	1.5	2	1.5	0.16667
D	Draft Project Summary	A	All	Low	2	3	4	3	0.3333
E	Kick Off Meeting Minutes	A	Thomas	Low	0.14	0.28	1	0.37666	0.14333
F	Project GANTT and PERT Analysis	A	All	Medium	1	2	3	1.83333	0.3333
G	Project Design Standards and Codes	A	All	Medium	1	2	3	2	0.3333

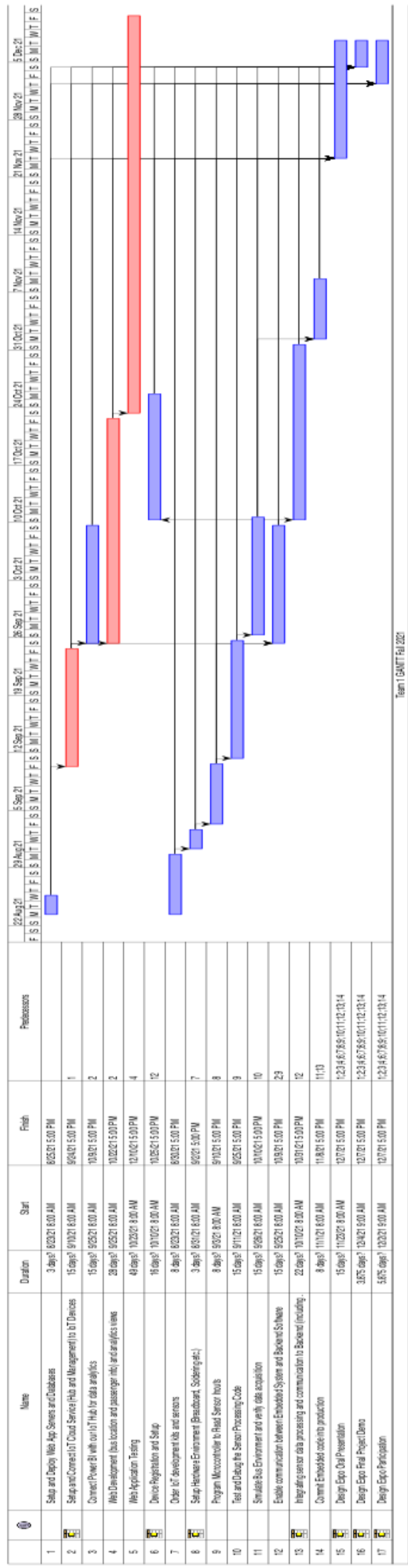
H	Project Job Budgeting and Costing	B	All	Medium	0.5	1	2	1.083333	0.25
I	Final Project Summary Form	D	All	Low	2	3	4	3	0.3333
J	Project Proposal	C,F,G,H	All	High	2	3	4	3	0.3333
K	Remaining Two Meeting Minutes	E	Shadman, David	Low	0.14	0.28	1	0.37666	0.1433
L-P	**Embedded System Setup and Integration**	H	Thomas, David	Medium					
L	Order IoT development kit and infrared sensors	H	Thomas, David	Low	0.28	1	2	1.046667	0.28667
M	Set up hardware environment	L	Thomas, David	Medium	0.14	0.28	1	0.37666	0.1433
N	Program microcontroller to read sensor inputs	M	Thomas, David	Medium	0.42	1	2	1.07	0.2633
O	Test and debug the sensor processing code	N	Thomas, David	Medium	0.42	2	3	1.9033333	0.43
P	Simulate the bus environment and verify accuracy of data acquisition	O	Thomas, David	High	1	2	3	2	0.3333

Q	Enable communication between the embedded system and Backend Software	M, U	Noah, Shadman	Medium	1	2	4	1.5	0.5
R	Integration of sensor data processing and communication to the backend software including testing and debugging	Q	Noah, Shadman	Medium	1	3	4	2.8333	0.5
S	Put the embedded device code into production	R,P	Thomas, David	Low	0.42	1	2	1.07	0.2633
T-V	**Back-End Software Development**	H	Shadman	High					
T	Setup web app and servers and database (Cloud Serices)	H	Shadman	Low					
U	Setup and connect IoT Hub and IoT Management with devices	T	Shadman	Medium	1	2	4	2.166667	0.5
V	Connect Power BI with our IoT Hub for data analytics	U	Noah, Thomas	High	1	2	4	2.1666667	0.5

W-Y	**Front-End Software Development**	H	Noah	High					
W	Web Development (for bus location and passenger info) and analytics views	H, U	Noah, Shadman	Low	3	4	6	4.166667	0.5
X	Web App Testing	W	Noah, Shadman	Low	4	6	7	5.83333	0.5
Y	Device Registration and Setup	Q	Noah, Shadman	Low	1	2	4	2.166667	0.5
Z- β	**Presentations and Design Expo**	Y,S,V,X	All	Medium					
Z	4872 Oral Presentation	Y, S, V, X	All	Low	1	2	3	2	0.3333
α	4872 Final Project Demonstration	Y, S, V, X	All	Medium	0.28	0.42	1	0.493333	0.12
β	4872 Participation in Design Expo	Y, S,V, X	All	Medium	0.28	0.70	1	0.68	0.12

Appendix C - Spring and Fall 2021 Project GANTT Chart





Appendix D - Project PERT Chart