**Feasibility Report**

**Clients:**

Prof. Pradipta De pde@georgiasouthern.edu

Prof. Andrew Allen andrewallen@georgiasouthern.edu

**Members:**

Taivon Watkins (Team Lead) tw05404@georgiasouthern.edu

William Collins wc01597@georgiasouthern.edu

Katie Lum kl03256@georgiasouthern.edu

Jared Dean jd06679@georgiasouthern.edu

Marvin Jenkins mj03505@georgiasouthern.edu

Jonathan Roney jr07449@georgiasouthern.edu

**Overview:**

The goal of our project is to create a low-cost parking lot tracking application for Georgia Southern campuses that will provide its user with information on the number of available spots in a given lot. The application will use nodes and a gateway that will update the database whenever a vehicle enters or leaves a designated lot.

**Need for the product:**

Currently the Georgia Southern Parking Office sends out scout cars to physically count the amount of available spots in each parking lot. This approach can be time consuming if there are multiple lots spread across the campus. Utilizing this product will give real-time readings on the amount of available parking spots at each lot instantaneously. The Parking Office will then be able to use this data to determine whether expansion is needed as well as be able to determine the busiest times at each lot. This information will also be beneficial to students, faculty, and guest as it will help them quickly determine which lot is best for them. This will ultimately lead to a decrease in overall congestion as drivers will not waste as much time finding parking spots and it will lower the frustration that comes with the search for parking spots.

**Scope:**

The expectations from this team shall be to design a sensory network that will keep track of the amount of cars entering and exiting the campus parking lots. The provided nodes will serve as a way to connect to a gateway to share data with the server’s database. The server will then scrutinize the shared information to ensure the accuracy of the readings and display it in the end-user application.

**Technical:**

**Amazon Web Service (AWS) -** We will use AWS for our database and storage due to its low cost, and reliability.

**Node -** The nodes will be equipped with special sensors that will detect when an object has passed through them. The nodes will then send information to the gateway using LoRaWan.

**Gateway -** The gateway shall be used to receive and upload information from the nodes to the database.

**Long Range Wide Area Network (LoRaWAN) -** LoRaWAN will be used to allow the nodes to communicate with the gateway from long distances. LoRaWAN is low cost, reliable, and secure making it the best option for our project.

**Task:**

The task of the project is to continue the work of the previous team with the hopes of delivering a fully functional prototype. The prototype should be free of the errors and concerns that the previous team dealt with. Due to the fact that we will be working off of what the previous team has done we have a list of future changes that they wanted to implement but were unable to. The tasks to be done that were listed in their final report include:

1. Gateway:
   1. Change encryption from ABP to OTAA.
      1. Reason: if an attacker gains access to code for the node or gateway security will be compromised
   2. It sounds like the gateway currently can only receive from the node and cannot transmit to the node.
      1. (from the final report) “If changing from single channel to multi channel gateway using a concentrator, the way packets are captured will need to be changed. Packet processing algorithm however should still function.”
   3. Gateway should be able to run a data analysis query after some condition is met
   4. Gateway currently does not send messages back to the node
      1. (from the final report)”The gateway sending messages back to the node has yet to be implemented/tested. This is a necessary step in order to implement time synchronization between the nodes and the gateway.”
2. Node:
   1. Change encryption method
   2. Packet transmission reliability has not been tested
   3. Clock sync has not been implemented
      1. Without a synchronized clocks any meaningful analysis of parking lot data is impossible.
   4. Node values must be tweaked after testing in an actual test environment.
   5. Separate code for the node into multiple files to make code easier to read and write
   6. Test to ensure node does not run out of memory after long periods of time

**Process to be followed:**

We will be using the Agile Scrum methodology for the project. There will be monthly goals set to ensure there is progress towards completing the project. We will have weekly sprints each month to achieve these goals.

**Deliverables:**

1. A Fully functional system consisting of the nodes, gateway, and database that will be able to effectively give data on the number of available parking spots on campus.
2. An application that the faculty and students can use to identify the number of available spaces.
3. An administrative console to monitor the system.

**Plan Of Action:**

Tentative Goals:

1. September

Week 1 - Project introduction, first team meeting

Week 2 - Obtain hardware and code from previous team, analyze and meet with team about code, set up cloud services and database

Week 3 - Get current code and hardware working to previous status

Week 4 - Discuss possible fixes and implementations for the project, prepare for sprints

1. October

Week 1 - Test packet reliability of the node and gateway

Week 2 - Ensure gateway can communicate with the node, adjust queries for determining whether a car has passed or not

Week 3 - Fix control algorithms

Week 4 - Prepare for demo

1. November

Week 1 - Fix encryption issue

Week 2 - Adjust and ensure the GUI is presenting the correct data

Week 3 - Final testing and QA

Week 4 - Prepare final report and demo

**Risk Analysis:**

1. **Technological risks:**
   1. If selected low-cost technology proves ineffective at detecting traffic events accurately or consistently, then the whole project may become completely ineffective. To combat this, the team will study data carefully and ensure all use cases are handled properly.
   2. Some types of weather can damage equipment and lead to hardware failure. The team has added specialized weatherproof hardware and housings to protect against this.
   3. The node may potentially use more power than originally intended, causing frequent maintenance on the batteries. If this becomes an issue the node may be attached to the power supply on the gate. If the gates are unavailable we will acquire a larger battery or potentially use solar panels.
   4. If the campus network is down or unaccessible, our data won’t be able to be sent to our cloud server from the gateway and our parking counts may be off depending on how long the network is down and how many cars come in and out during that period. A solution would be to use AWS Device Shadow. It allows us to retrieve the last set of data the node collected before going offline. This will let us post an accurate count up until it went offline and we can do manual count and update for the lot if the network is down too long.
2. **Requirements risk:**
   1. *Change of requirements*: It is possible that the client will change the requirements after the project begins. Depending on the new requirements the team may have to redesign already completed sections to accommodate this change. To avoid this the team will maintain communication with the client during development so it will be possible to easily adapt.
   2. *Miscommunication:*It is possible that while working, more than one person will be doing the same thing that another member is already working on. This can result in time being wasted or something else not being completed and could put us behind schedule. To avoid this the team will communicate with each other frequently informing the others of the progress of what they are working on, when they have completed something, any unexpected problems, and what they will be working on next. All code will be shared with other members via some collaborative application such as google drive or github.
3. **Resources and time risk:** 
   1. It is very possible that we will not have enough time to complete the full functionality and implementation during the timeframe of one semester. There is also a similar risk of falling behind schedule if there become any large impediments. The team is all taking different classes and is involved in extracurricular activities that causes our schedules to differ.
   2. The technologies and hardware we have chosen to use could be insufficient to complete the tasks. We have chosen some open-source libraries and software to help save on costs and time. So, these libraries and software could cause impediments within our project.
4. **Social and legal risk:** 
   1. When it comes to legal aspects we will have to worry about the protection of the users’ login information. We will be dealing with LoRaWAN security. In this data transfer system there are 2 different levels of keys. First you have the network session key that is used for determining the validity of the messages between the node and network. Then you have the application session key that deals specifically with the encryption of the data between the node and the handler. These two keys are unique for the specific device and session. Using OTAA the keys will regenerate every time the device is activated. In this system the application key is used to find the 2 session keys on activation.
5. **Financial risk:** 
   1. To the best of our knowledge, we currently have all the materials needed to complete the project.

**Visibility Plan:**

**External:** To ensure the clients are up-to-date with the project’s progress, issues that appear while working on the project, and changes to the requirements the client will be contacted as needed. For smaller updates the client will be updated by email on an as needed basis to be informed of issues that arise or the completion of a requirement. For large updates the team will have a meeting with the client so the client will be able to view overall progress and the project in its current state via deliverables.

**Internal:** The team will meet at least once per week to discuss the work that has been done, problems that have occurred, and what will be worked on next. Meetings shall be documented with a list of those who have attended to keep track of participation and so we will be able to inform those who missed the meeting on what has happened. Additionally team members will communicate with each other via email, groupme, or some other application to notify others of their progress and source code will also be shared.