Design Proposal and Team Contract

G10 LIONS

Autonomous Drone System

Team Member Motivations and Ideas

Rayyan Jamil

The LIONS Autonomous Drone System project presents a compelling and multidisciplinary challenge that matches backgrounds in Aviation Design, Information Technology, and Software Engineering. I am excited to anticipate and strategically overcome the technical and logistical roadblocks that are inherent in achieving seamless, intelligent cooperation between these autonomous aerial platforms in dynamic environments. The profound challenge of designing an AI system that can adapt, learn, and make robust decisions in unpredictable, real-world scenarios is precisely the kind of problem I find most engaging and intellectually stimulating.

An exciting draw of this project lies in its hardware requirements, notably the integration of the compact computing platforms such as the NVIDIA Jetson Nano for and the Arduino Portenta H7 for native, on-board processing of AI models and other dockerized services. This direct interaction with deployable AI hardware in an airborne context offers a tangible link between AI concepts and their practical physical use case.

Furthermore, I have hands-on experience working with simulation data, a process often involving feeding this data into Agentic AI workflows to test their decision-making processes within controlled environments. However, an appealing distinction in this LIONS project is the explicit decision to avoid any artificial or simulated data for the demonstration phase. Instead, the system will rely on live data streamed directly from on-board sensors and other databases.

Finally, another compelling reason for my strong interest in this project is its undeniably AI-heavy nature. The explicit focus on the use of Large Language Models (LLMs), Agentic workflows, and the challenge of learning how to implement these advanced AI concepts within systems that function autonomously in the real world is interesting to me. It is one thing to develop and test AI models in a controlled

laboratory setting, but quite another to see them deployed in dynamic, unpredictable environments where their decisions have immediate, tangible consequences. With these autonomous drones, the application of AI is not merely academic; it is practical, offering concrete solutions to real-world problems. This project offers a unique and invaluable chance to bridge the critical gap between cutting-edge AI and impactful, real-world deployment.

Abraham Ng

My motivation for picking this project is threefold. First, I am interested in developing stronger skills in Python and its associated AI-related libraries. Python has become ubiquitous in programming with the rise of AI, and it has allowed for greater readability and faster code output. However my primary experience with programming prior to this point has been focused on C and Java, as they have been the only required languages to use in my coursework thus far, aside from no more than a handful of assignments that required Python. Given the relevance and prevalence of Python, I think it is important to gain more experience and familiarity with the language (particularly alongside using PyTorch).

Second, I am eager to learn how to use and train AI models. In the large project in the prerequisite processes courses, I did have some limited experience with an AI model, but primarily from the front-end creating an API to interface with it and call it, rather than actually being involved in training and developing the AI model itself. In this project, I would like to gain a better understanding of how to utilize LLMs and the technical aspects behind them.

Third, I am hopeful to gain experience working directly with a defense contractor. As I have settled down in the Orlando area for the time being with a wife and a four-year-old son, and as the local job market has a high concentration of defense contractors and related companies, I am looking to continue my career locally for now, and I anticipate that this project will be instrumental in helping me land my first post-bachelor job related to computer science.

Given my prior areas of focus in the two projects in the prerequisite processes class, I think my skill set is best suited in the immediate future to focus on building the front-end React framework and its necessary APIs, developing and integrating APIs between the drone sensors and the back-end databases, and managing and creating the back-end database to store open-source GPS/weather data and drone sensor

data. Over the next few weeks, I also hope to rapidly acquire skills in Python and AI models in order to help our team in programming instruction parsing backed by LLMs.

Nathan Dalfonso

I'm very excited to work on the autonomous drone project that Northrop Grumman, now Serco, has pitched and is willing to sponsor. When I first saw the presentation, I knew it would be a great way to combine my interests in AI, multi-agent systems, and hardware. Drones have always fascinated me, and getting the chance to work on one and help design it is an opportunity I couldn't let pass by. This project enables us to develop a system that accepts human instructions and utilizes AI to execute them independently. Having the freedom to bring our ideas to life while still receiving guidance from professionals makes this a great opportunity to learn new things, hone my skills, push my current capabilities and work in a real-world setting.

One of the things I like most about this project is that it's hands-on. I've done software projects before, but I haven't had the chance to build a system that brings software and hardware together like this. Designing a drone, connecting sensors, and making sure everything works as it should is something I hope to do as a full-time job in the future. It's a challenge, but one that I think will be very rewarding.

For the design part, the system should be easy for people to use. The human-in-the-loop interface should help users see what the drone is doing and why. The drone should be able to explain itself in simple language that anyone can understand. A reliable system is also important so that people trust it to work, especially in critical situations.

I'm very appreciative that Serco is giving us the design freedom to come up with our ideas instead of handing us just a strict plan. Although sometimes having a strict plan from a client makes things easier, I feel that this kind of creative freedom is really valuable because it lets us try different ideas and see what works best. It's how innovative and useful solutions come about that sometimes produce world-renowned breakthroughs and new ideas from others with more experience which I find fascinating. Working closely with my teammates to combine all of our skills and ideas will be just as important as the technology itself. Everyone brings something different

to the table, and I think that teamwork will help us build something we can all be proud of.

I'm looking forward to learning a lot from this project. There's so much to explore from the AI portion of the design to the sensors, and getting all the pieces to work together in a real system. It's one thing to write code in a class, but it's a whole different challenge to build something that works. Helping deliver a drone that performs well will show what we can achieve as a team as well as show my limitations at this moment. This project will give me new skills, a better understanding of complex problem-solving, and more confidence in working on interdisciplinary projects in the future.

Connor Hallman

I'm really excited about this project. We have been afforded a lot of creative freedom and autonomy by our sponsors. I was expecting to get more specific requirements when we met with our sponsors, but they surprised me. They are intentionally being very lenient with the requirements. They just want a system of drones (likely just 2) that performs a task and uses ai. I originally was dreading getting this response from them, but I've since come to terms with it and am now more excited about the project. It means we have true freedom. They didn't do it to make our lives worse, but to make the project more fun.

Another great thing is that we have a very diverse team. Rayyan already has experience with agentic ai and aerospace. Nathan is an experienced dev himself with prior job experience. And Abraham brings years of real job experience coming from working full time jobs in a different industry. All together, I don't think we could have asked for a better team.

I come with a fair amount of web development experience. I've built many web apps in many languages: react, svelte, blazor. This will make the react web app portion of the project a lot easier. That's really good since the less time we spend working on the web app, the more time we get to spend working on the agentic ai drones! I also have some familiarity with AI, I'm somewhat of a chatgpt user myself, you could call me a professional prompter. Jokes. On a serious note, I have used AI in some web apps I've created for parsing unstructured data and generating content for users. I'm familiar with having LLM's created structured (JSON) outputs which I'm sure we will need for this assignment. I also took the algorithm for machine learning course which taught me a lot about the other low level concepts of the class.

Additionally, I have some professional workspace experience. I've been working as a full-stack developer for about 2 years. This has familiarized me with scrum/agile development and a lot of the common pitfalls of working in a team. I am very familiar with the importance of gathering requirements, being understanding with team

members, sticking to deadlines, and everything else that comes with developing in a team.

I am yet to come up with specific “solutions” for the project. But I do know what I want a solution to look like. I was the drones to accomplish a shared goal. Each drone should have its own role with a few overlapping responsibilities. Decision making should be done by using the multi-modal sensors of the drones to describe the situation to an LLM and have it generate a structured response. That structured response will be parsed by our program and our code will then behave like a finite-state automata. Whatever the task, I want it to be very ambitious. Something that cannot be accomplished algorithmicly alone.

Autonomous Drone System Technical Outline

1. Introduction

The following outlines our senior design project: an autonomous drone system. Our aim is to develop a platform where an Agentic AI workflow directs a multi-drone team to execute complex, real-world missions. This core intelligence, driven by Large Language Models (LLMs) and a sophisticated Agentic workflow, orchestrates the drones, which function as coordinated actors in the field. The system is designed to translate high-level human instructions into a series of synchronized actions for the drone team, while interpreting real-time data to generate natural language status reports.

1.1. Use Case: Rapid Assessment and Response

When a disaster strikes or changes are made to an environment, the immediate needs are to understand the scope of the damage and locate survivors. Our system is designed specifically for this high-stakes scenario. The central agentic system will deploy and command a team of drones to perform two critical, overlapping functions. Initially, it will direct the drones to conduct a swift aerial survey to generate an accurate, up-to-date 2D/3D map of the disaster zone for immediate situational awareness. Concurrently, it will command the drones to execute a coordinated thermal search, identifying and pinpointing the locations of potential survivors and overlaying this critical data onto the environmental map to create data for rescue teams. The system will algorithmically assess and discover items that have been misplaced and other points of interest (POI's).

2. Centralized Agentic Architecture

Our project's architecture is defined by a single, powerful agentic system that serves as the "brain" of the operation, with the drones acting as its physical extensions. This centralized model ensures coherent, top-down mission planning and execution, treating the drones as specialized "tools" that are called upon to perform specific functions. All high-level reasoning, planning, and adaptation occur within the central system, not on the individual drones.

2.1. The Central Agentic System (Command Hub)

The core of our project is the agentic workflow system, which will reside at the Ground Control Station (GCS). This system is the sole source of intelligence and decision-making. It will be built in Python 3.9+ and powered by the LangChain and LangGraph frameworks, enabling it to perform complex reasoning, create multi-step plans, and adapt to new information. This central agent will host a primary Large Language Model (e.g., GPT-4, Llama 3) to interpret human commands and synthesize incoming data. Its "tools" will be functions that allow it to command the drone actors—for example, by calling “dispatch_drone_to_waypoint(drone_1, ...)” or “activate_thermal_scan(drone_2)”.

The human interface to this central agent will be a React-based web dashboard. This GCS provides the operator with a comprehensive operational view, centered around an interactive map that is populated in real-time. Operators will issue natural language commands to the agent, monitor its status and reasoning through a log window, and view live video or thermal feeds transmitted from the drone actors.

2.2. The Drone Actors (Field Hardware)

The drones themselves are task-specific actors that execute commands issued by the central agentic system. The physical platform will be modular which allows for the necessary payload and compatibility with software-controlled flight controllers. The on-board compute requirements are focused on command execution and data streaming, not independent reasoning.

The drone's payload consists solely of its sensors and a capable flight controller. All sensor data, including high-resolution video, thermal imaging, GPS, and IMU data, is collected by the flight controller and streamed directly to the Ground Control Station. The flight controller’s only role is to execute flight commands received from the GCS and relay sensor information. This offloads all computational weight, maximizing flight time and simplifying the drone's design, while concentrating all processing power at the GCS where it is not constrained by weight or battery life.

3. Real-time Open-Source Data Integration

To enhance the central agent's planning capabilities, the system will ingest data from open-source APIs. A Python backend service within the GCS will fetch and standardize data from the OpenWeather API, FAA data sets, and NOAA data sets. The central

agent will treat this external information as another data source for its decision-making process, allowing it to dynamically adjust mission parameters—for example, by commanding the drone actors to alter their flight paths to avoid a high-wind corridor.

4. LLM-Driven Functional Flow

The centralized LLM running on an NVIDIA Jetson Nano is the cognitive engine that initiates and interprets the mission flow. The process begins when an operator issues a high-level command to the central agent, such as: "A building has collapsed at 123 Main St. Deploy to create a 3D map of the rubble pile and identify all heat signatures." The agent's LLM interprets this and uses its LangGraph framework to create a structured plan of action. This plan involves calling a sequence of tools that command the drone actors.

The agent might first call a tool to send Drone 1 on a high-altitude photogrammetry sweep, and then call another tool to command Drone 2 to begin a low-altitude thermal search. The drone actors execute these commands and stream back data—telemetry, video feeds, or CV alerts. The central agent continuously processes this incoming information, using the LLM to synthesize it into human-readable reports and make adaptive decisions, such as re-tasking a drone to get a closer look at a detected anomaly.

5. Development Environment and Tools

Our development will be managed using Git and Github for version control. VS Code will serve as our primary IDE. We will utilize Docker to containerize the different software components, including the central agentic system, the GCS frontend, and any data processing streams. For rapid prototyping, a synthetic data container will simulate the drone actors, allowing us to test the central agent's logic and LLM responses in a controlled and predictable environment.

6. Conclusion

This project pioneers a deeply integrated autonomous drone system where a single, centralized agentic AI orchestrates a team of drone actors for a critical purpose. By shifting intelligence from the drones to a core command system, we can achieve more sophisticated and coherent multi-drone coordination. The architecture outlined here

provides a roadmap for delivering a system that interprets human intent, commands hardware in real-world environments, and executes missions with unified intelligence.

Team Contract

General Expectations

The format and content for this chart was heavily inspired by the SD bootcamp.

Norm	**Behaviors to Support the Norm**
Be respectful	◎ Be early to all meetings ◎ Let people know if you are running late ◎ Listen first, speak later ◎ Repeat back what you hear to communicate clearly
Be honest	◎ Don't guess; look it up or collect it ◎ Share openly even if you hit a problem or fall behind
Be accountable	◎ Do what you say you will do!!!
Be innovative	◎ Understand the tasks and goals ◎ Plan (not just hope) to exceed expectations ◎ Think outside the box and try new things
Be compassionate	◎ Err on the side of forgiveness instead of criticism ◎ Treat others the way you want to be treated ◎ Trust each other and work together ◎ Frame issues as problems to solve, not as crises

As we begin working on this project, it is essential that we establish clear expectations to ensure that we function effectively and responsibly as a team. We are all expected to respect each other's time by arriving on time for all meetings and notifying the team promptly if we are running late to a meeting or if a deadline needs to be extended. Punctuality shows respect for each other's efforts and helps us stay productive. Respecting each other's opinions is equally important, and we should listen

carefully before instantly responding or dismissing them. Repeating back what someone says can help confirm issues, reduce misunderstandings, and support effective collaboration among teammates.

We should base our decisions on data and facts rather than assumptions. When questions or issues arise, we are encouraged to research or gather the necessary information to make informed decisions and maintain project momentum. We should always strive to exceed expectations rather than just meet the minimum requirements for the task at hand. Understanding what is expected and what we aim to deliver will help us ensure we achieve high-quality results. Each of us is responsible for completing our assigned tasks within a reasonable timeframe. If a task cannot be completed, it is important to communicate this promptly with the project manager and the team so that we can find a solution and keep the project moving forward in a timely manner. Accountability fosters trust, keeps the project moving forward, and creates a clean and open work environment.

Honesty is essential to maintaining a productive team environment. Hiding mistakes or being dishonest harms the entire team, not just the individual member. We should be open about any problems that we might be experiencing that may affect the project so that we can resolve them together as a team. We should assume that others are acting in good faith and treat each other with respect and trust to promote a positive and supportive atmosphere in the work environment. Innovation and creativity are vital for success, and we are encouraged to think creatively, propose new ideas, and be open to exploring different solutions.

A compassionate and patient approach is essential for a successful team. We all have other responsibilities that can occasionally impact our work, and supporting each other through challenges strengthens our team bond and workflow. Collaboration is key; we should work together, share ideas openly, and maintain clear communication to solve problems efficiently and build a strong final product that we would all be proud to represent. Challenges should be viewed as problems to solve, not crises. By

maintaining a calm and solution-focused mindset, we can navigate obstacles and keep the project moving in a positive direction.

By embracing these values: respect, honesty, accountability, innovation, compassion, collaboration, and a focus on problem-solving, we will be able to build a strong, successful team and deliver a project that meets and exceeds all expectations from the team itself and our sponsors.

How To Meet And Handle Meetings

There are two types of meetings that we will have throughout the semester, virtual and in-person. Virtual meetings will be performed on an as-needed basis with 2 or more members to accomplish small, isolated goals. Things like debugging code, brain-storming solutions to a small problem, questions about someone else's code, and the occasional touch-basis with all members to see what everyone is doing and if anyone is experiencing any blockers. I'd like to say that virtual meetings should be planned in advance, but many will likely happen on the fly and that's okay. Virtual meetings shouldn't exceed 1 hour, anything longer than that should be in person.

In person meetings will be done on a recurring basis every week on the same day (TBD). They should last 30 minutes to an hour but may be longer or shorter depending on the fast or famine of that week. Attendance will be mandatory. Here is the non-exhaustive list of things we will do in our meetings:

- Sprint Recap/TB - Go over last week's agenda (jira sprint board) and determine what tasks have been completed or still need work.
- Share any blockers we are encountering
    - Try to see if anyone can solve another members blocker
- Collaborate on issues where possible
- Sprint Planning - Plan out and prioritize tasks for the next sprint
- Code Reviews - Review commits from previous week and make sure are good quality
- Demo new features - Let members show off what they have completed
- Work on documentation, if needed
- Discuss technical debt
- Purchase items needed for projects

In addition to these weekly meetings, there will also be occasional in person meetings at a separate drone-friendly location to fly the drones and test our code on real scenarios. We still need to discuss with Gerber where to have these meetings.

Accountability

Foremost, by meeting regularly as per the previous section, we will be able to hold each other accountable by asking each group member to show their own work. These meetings will also be an opportunity for us to offer suggestions, whether in regards to areas of improvement or in regards to problems we are unable to solve independently. In addition, the inclusion of our mentor biweekly will help with accountability, particularly since our mentor is himself a software engineer with decades of experience.

Furthermore, by using collaborative tools such as Jira and GitHub, we will clearly know who is delegated what work and also be able to monitor coding progress in a pseudo-live fashion respectively. Doing so will also allow us to delineate specific targets and deadlines for each task as well as see commented and committed documentation of work, also respectively. Additionally we will host scrum meetings to discuss progress on Jira tickets. Our sponsor from Serco has expressed a lot of interest in getting access to the Jira board and being part of the scrum meetings. This additional oversight will further boost accountability and motivation to complete assigned tickets.

Finally, by fostering a supportive atmosphere with our norms and expectations, we will be able to build trust and subsequently transparency with one another regarding any missed deadlines or difficulties encountered while coding. Through supporting one another, we will be able to hold to the “trust but verify” mantra in an open, non hostile manner.

How To Handle Unavoidable Conflicts

Conflicts are unavoidable. The way we will handle conflicts will depend on the type of conflict. We will categorize conflicts into two groups: Can Work, Can't Work; depending on whether the member can still complete work. In either situation, the member should notify the other members ASAP once they know a conflict will occur. The member should also consider their responsibilities and delegate tasks that they can't to other members, especially time sensitive ones. No blame or ill will should be received for a member experiencing a conflict. They are to be expected when working on a project.

The Can Work category will contain most vacations and illnesses. Any situation where the member can still complete work in some facet. In this situation, the member will be expected to do what they can while they are "out of office". If possible, they should still attend all meetings, virtually, and do their share of work. If the member can't meaningfully contribute to a meeting virtually, like a meeting to work on hardware, they are not expected to attend.

In the Can't Work situation, the member will be excused from any meeting or work during their conflict. During their absence, their lost productivity should be shared among all the remaining members, not placed on one individual member. However, once they return to office, they will be expected to make up for their lost productivity by doing extra contributions once they return. These extra responsibilities should be reasonable though. They shouldn't lead to burnout or exceed the amount of work that wasn't done during the absence.

“Signatures”

Rayyan Jamil: Rayyan Jamil

Abraham Ng: Abraham Ng

Nathan Dalfonso: Nathaniel D'Alfonso

Connor Hallman: Connor Hallman