2 Project Plan

2.1 Project Management/Tracking Procedures

Which of agile, waterfall or waterfall+agile project management style are you adopting. Justify it with respect to the project goals.

- As there are two distinct portions of the project - design document and actual implementation, our team will adopt the dual-style of waterfall (for the sequential development of the parts of the design document) + agile (for the implementation). Utilizing the agile approach to project management during the implementation phase will enable us to work on multiple parts of the project simultaneously, with work that may be broken down into small tasks which may be completed in a 2 week sprint period. In addition, the framework's flexibility will allow us to make changes later on as necessary, with emphasis on a modular design for the different parts of the project.

  As specific examples:

  - While part of the team is implementing the frontend visualization of the drone simulation, another part of the team will be implementing the backend queue system for simulation with new input combinations.
  - Complementary, when working on the UI, one sprint would be implementing the drop-down menu functionality, followed by the next one implementing the button-like selectors.

What will your group use to track progress throughout the course of this and the next semester. This could include Git, Github, Trello, Slack or any other tools helpful in project management.

- Git (for code source control)
- Gitlab (for code repository)
- Gitlab Issues & Issue Boards (for task decomposition/user stories)
- Discord (communication)

2.2 Task Decomposition

In order to solve the problem at hand, it helps to decompose it into multiple tasks and subtasks and to understand interdependence among tasks. This step might be useful even if you adopt agile methodology. If you are agile, you can also provide a linear progression of completed requirements aligned with your sprints for the entire project.

1. User Authentication System
    a. Frontend create new user screen
    b. Backend create new user endpoint
    c. Backend login endpoint
    d. Frontend login screen

    e. Reset password functionality
2. User can choose an event phenomena file.
    a. Frontend screen to choose event phenomena file.
    b. Backend saves event input. (standardized format?)
3. A system which can save previous simulations
    a. Database which saves run data for specified input combinations.
    b. Utilizes previous input combinations to output old saved data rather than running a new simulation
4. Running simulations
    a. Simulation is ran on the backend
        i. Queue system for new input combinations.
            1. FCFS -> SRTF -> MLFQ
        ii. Push Notification for when simulations are complete.
        iii. 30 min time limit on simulation runtime.
        iv. Finished simulations will return a file with information on the ran simulation
    b. Simulation on the Frontend
        i. Screen with input options for setting up the simulation.
            1. List of options for the algorithms
            2. List of options for event phenomena
        ii. Screen with input options for viewing the simulation
            1. List of options for which simulation results to display

2.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

What are some key milestones in your proposed project? It may be helpful to develop these milestones for each task and subtask from 2.2. How do you measure progress on a given task? These metrics, preferably quantifiable, should be developed for each task. The milestones should be stated in terms of these metrics: Machine learning algorithm XYZ will classify with 80% accuracy; the pattern recognition logic on FPGA will recognize a pattern every 1 ms (at 1K patterns/sec throughput). ML accuracy target might go up to 90% from 80%.

In an agile development process, these milestones can be refined with successive iterations/sprints (perhaps a subset of your requirements applicable to those sprint).

Fall 2022

- User Needs & Problem Statement (+ literature survey)
- Functional & Non-functional requirements
- Project Planning
- Project Design
- Tool & Platform Considerations
- Testing
- Professionalism
- Final Document & Presentation

Progress on Fall 2022 milestones will be measured by the feedback received on the submitted documents.

Spring 2023

- Platform Setup
    - Databases
    - URLs
    - Code base template

Progress will be measured by direct checks for each sprint in terms of functionality of the respective components/subcomponents.

- Authentication
    - Create new user (front & back)
    - Login user (front & back)

Progress will be measured by completing unit testing for each applicable outcome. This will include failed login/registration attempts, along with successful ones as well.

There are multiple milestones under a common "umbrella" of algorithm execution (running simulation):

- Different source code language support
    - Python
    - C++
- System that takes event input, translates it to be passed given algorithm, executes algorithm with event input.
    - Different grid-sizes and phenomena distribution
    - Different (algorithms, input) pairs

Progress will be measured by ensuring the correct parameters are sent by using mocking calls to the server and verifying results with unit testing.

- Takes output of algorithm running, translates output into simulation data file.
- Simulation will be terminated if it exceeds 30 minutes

Progress will be measured by ensuring functionality that algorithm runs and output is as expected using unit testing.

- Simulation Setup/Request
    - Event phenomena selection
    - Algorithm selection

Progress will be measured by observing the execution of a specific algorithm over different datasets.

- Simulation data response
    - w/Repeated input combination
        - Requested input combination will return previously run simulation files.
    - w/New input combination

- Input combinations will be put into queue
- Data of completed simulations will be stored based on the input combination
- Push notification will be sent to the user that initiated the simulation upon completion/failure

Progress will be measured by ensuring the correct data is stored in the database in a readable format. This can be done unit testing the database.

- Visualizing the Simulation (frontend)
  - Display drones
  - Display phenomena (obstacles, wind, etc.)
  - Display diagnostic data (speed, battery life, etc.)

Progress will be measured two ways. One can be the eye test, making sure that each simulation is visualized in a proper manner, the second being using unit tests with a predictable dataset.

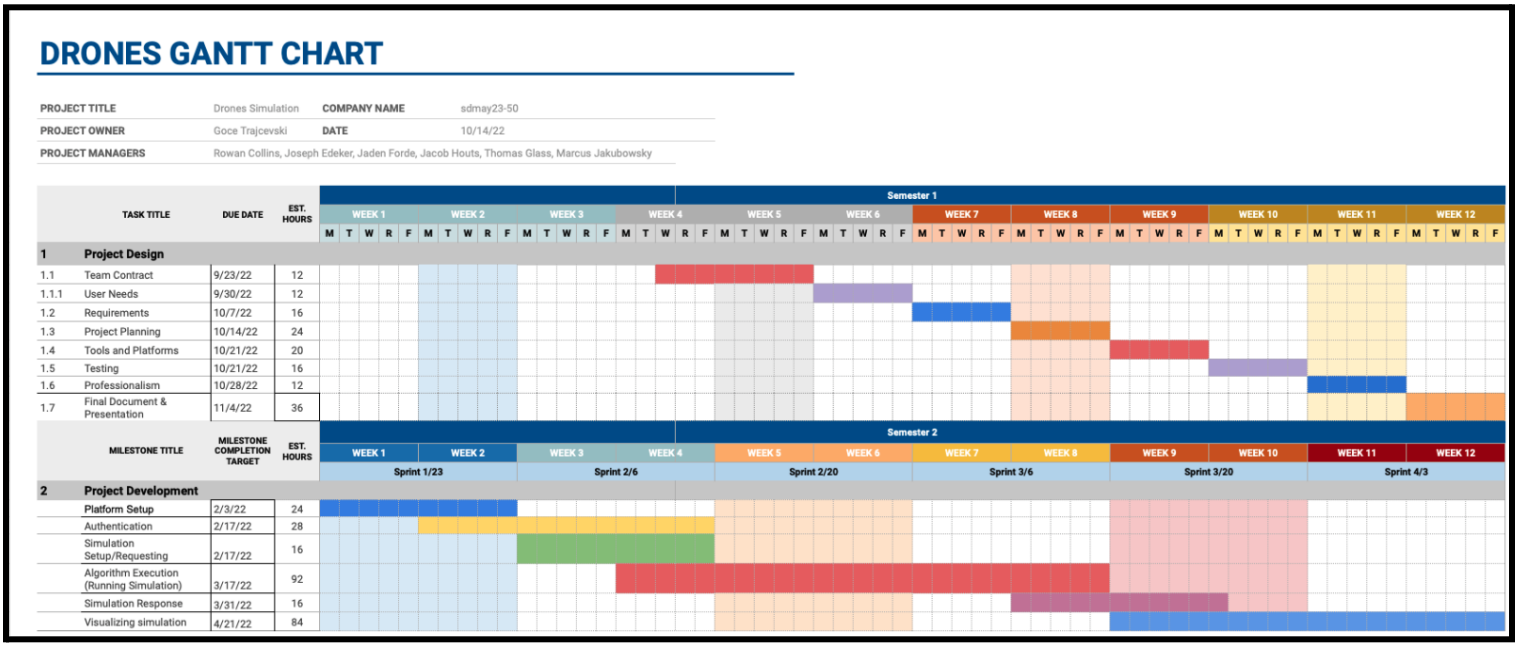
2.4 Project Timeline/Schedule

• A realistic, well-planned schedule is an essential component of every well-planned project

• Most scheduling errors occur as the result of either not properly identifying all of the necessary activities (tasks and/or subtasks) or not properly estimating the amount of effort required to correctly complete the activity

• A detailed schedule is needed as a part of the plan:

– Start with a Gantt chart showing the tasks (that you developed in 2.2) and associated subtasks versus the proposed project calendar (including both 491 and 492 semesters). The Gantt chart shall be referenced and summarized in the text.



2.5 Risks And Risk Management/Mitigation

Consider for each task what risks exist (certain performance target may not be met; certain tool may not work as expected) and assign an educated guess of probability for that risk. For any risk factor with a probability exceeding 0.5, develop a risk mitigation plan. Can you eliminate that task and add another task or set of tasks that might cost more? Can you buy something off-the-shelf from the market to achieve that functionality? Can you try an alternative tool, technology, algorithm, or board?

Workers not completing assigned tasks

Agile projects can associate risks and risk mitigation with each sprint.

Task/Component	Risk	Risk Probability	Mitigation
Design	All requirements were not properly defined	0.2	None
Development	Workers not completing assigned tasks	0.5	Discipline for missing deadlines will be increased.
Planning	Client size unknown	0.5	Conduct research to figure out how many people will use the product.
Maintenance	Users not knowing how to use the application	0.6	Develop tutorials to help the users.
Testing	Tests do not cover all possible combinations of environments and events	0.7	Researching into testing tools that can help.

Planning	Cannot find proper frameworks to create application	0.2	None
Design	Use cases may not properly describe how a user uses the application	0.6	Conduct proper research to understand how different users will interact with application
Development	Going over budget	0.4	None
Design	User Information Privacy	0.6	Design code to be safe from sql injection attacks etc.

2.6 Personnel Effort Requirements

Include a detailed estimate in the form of a table accompanied by a textual reference and explanation. This estimate shall be done on a task-by-task basis and should be the projected effort in the total number of person-hours required to perform the task.

Total Hours: 6 ppl x approx. 3 hrs/week x 12 weeks = 216 hours

Task Name	Detail	Notes	Hours Required
Platform Setup			
Frontend	Establish a baseline template for the frontend	Should be accessible from a web browser	8
Backend	Establish a baseline template for the backend	Should be reachable via HTTP requests	8
Database	Set up a data storage medium	Should be reachable via IP or other communication protocol	8
User Authentication			

User Creation	Create a sign up page where users can create a new user with their chose information	Information Needed for New User: Username, Password, Email	16
Reset Password	Functionality to allow users to change passwords		4
User Login	Create a login page where users can sign in using previously created users	Login with: Username or Email and Password	8
Simulation Setup/Request			
Event Phenomena Selection	Create a series of drop down menus and fields/sliders for users to select phenomena to be applied to the algorithm	Possible Phenomena: Wind, Fire, Explosion	8
Algorithm Selection	Create a dropdown of possible algorithms to use or the ability to add their own algorithm	Algorithms come from Professor Goce and his team	8
Algorithm Execution			
Algorithm Source Code Translation Layer	Ability to decode different algorithms into usable programs for the backend		32
Algorithm Execution	Ability to run selected algorithms on backend		16
Exporting Algorithm Data	1. Define file format for algorithm output 2. Encode algorithm output to file		8
Algorithm Notification System	1a. Define notification events 1b. Backend recognizes events 2. Backend communicates events to Frontend 3. Frontend displays notifications	Sample events: simulation request received, simulation executing, simulation execution complete, simulation runtime error	24
Simulation Queue System	Ability to queue multiple algorithms. Notifications to be sent upon completions of queued algorithms.	Possible Queue Systems: FCFS -> SRTF -> MLFQ	12
Simulation Data Response			

Simulation File Transfer (between Front & Back End)	Ability to send stored data (back end) to the UI (front end)	Transfer data in an efficient format using JSON	4
Storage for Previously Completed Simulations	Upon completion of algorithms, store data into database	Store datasets into a database on our preferred cloud service provider	12
Simulation Visualization			
Display Grid/Coordinate System	Create a 2D grid with coordinates built in		8
Display Drones	Display drones on 2D grid based on position data from algorithm		8
Display Phenomena	Define indicators and display phenomena on the 2D grid	Possible Phenomena: Wind, Fire, Explosion	24
Indicators for Drone Diagnostic Data	Hover, Overlay, or Panel for Drone information	Examples: Battery Life, Position (grid coordinates), Speed, Elevation	16
Media Controls for Drone Simulations	Play/Pause/Fast Forward/Rewind		16
Simulation Menu Selection	Selection for which simulation to play and loading screen when waiting for simulation to start		12
		Total	260

2.7 Other Resource Requirements

Identify the other resources aside from financial (such as parts and materials) required to complete the project.

Algorithms given by Prof Goce & Prabin.

Platform to host the application.