

1.1 REQUIREMENTS & CONSTRAINTS

Functional Requirements:

- The software shall allow the user to login to a personal account profile.
- The software shall give the user the ability to choose which phenomena (e.g. fire, explosion, ...) to apply to the drone flight simulation.
- The software shall give the user the ability to choose which drone algorithm(s) to apply to the drone flight simulation.
- The software shall accept input combinations through selected files.
- The software shall provide a visualization of the drone flight.
 - The software shall have a 2d grid layout of the geographical area.
- The software shall calculate drone statistics and values over time and record them to storage.
- The software shall provide a view of statistics around the drones (battery life, location, speed, etc).
- The software shall save previously run simulations for input combinations that will be accessible for other users.
- If the user selects an input combination that has been simulated before then the software shall return back the already run simulation data.
- If the user selects an input combination that has not been simulated previously then the software shall queue the simulation to be run on the backend.
- If a simulation is run from a queued input combination then the software shall notify the requesting user when the simulation is completed (push-based notification).
- The software shall accept user input (file) which is given in source code which is ready to run (compiled or interpreted, etc).
- The software shall store the simulation output in a format containing: starting location, starting time, destination location, arrival time, trajectory.
- The software shall have 3 UI components: list of algorithms to pick, list of event phenomena input, visualization screen of simulation output.
 - Possibly a 4th component for notifications.
-

Non-Functional Requirements:

- The software should be easy to use and understandable since not all of our users have a technical background
- UI elements of the software shall be intuitive and clearly labeled or documented.
- The software shall handle errors in the input gracefully.
- The software shall combine concurrent access for already executed input combinations and will run push notifications for users with new input simulations.
- The software shall be compatible with Windows, MacOS, and Linux.
- The software shall be developed in a manner that is supportable & maintainable after our team leaves.

Constraints:

- The software development process shall not cost more than \$300.
- If the runtime of the simulation exceeds a time period of 30 min, then the software shall terminate the simulation and notify the user of the termination.

1.2 ENGINEERING STANDARDS

What Engineering standards are likely to apply to your project? Some standards might be built into your requirements (Use 802.11 ac wifi standard) and many others might fall out of design. For each standard listed, also provide a brief justification.

Engineering Standards	Justification
Scrum Methodology	We need to have structure on how we develop the project. We will work in 2 week sprints.
IEEE Standard 1063: Standard for Software User	We will make sure that our project is documented correctly for future users and developers.
IEEE Standard 1012: Standard for Software Verification and Validation	We must make sure that our project meets the clients requirement completely and thoroughly.
IEEE 610.12: Standard Glossary of Software Engineering Terminology	Make sure we use language similar to our peers.
Software Engineering Code of Ethics and Professional Practice (Principle 3)	We will make sure that our product meets the highest possible standards.
Software Engineering Code of Ethics and Professional Practice (Principle 2)	We will meet the expectations of our client as well as the potential users of our product.

LIST OF QUESTIONS (ASKED TO HELP COMPILE OUR REQUIREMENTS)

1. How many users should be able to use the software at the same time?
 - a. Web-based application. User selects algorithm & user select event phenomena set.
 - b. If a user selects an input combination which has already been run and information stored, then previous run data should be returned.
 - c. Multiple users with their own set of saved data.
 - d. Multiple users can use the system concurrently IF the saved data from previously run simulations is available, otherwise the user's new input combination then will be a "pushed-based" notification for when the simulation is done running in the background (could take up to 60 min). When completed, the user can then visualize the run simulation.
 - i. Queue based system.
2. How will the input be given? (File for events & algorithm?)
 - a. Should the user be able to provide their own algorithms?
 - b. Input will be given in files.
 - c. The user's input will be given in source code which is ready to run.
 - i. The source code will be in a format which can be run on the backend (compile or interpreted, etc)
3. Should there be generated output describing the results of the simulation?
 - a. Yes. Input should be paired to the storage of output.
 - b. Output format: starting location, starting time, destination location, arrival time, trajectory
 - i. Possible more format for multiple drones (to be discussed)
4. What will the UI look like?
 - a. UI will have 3 components after the login screen:
 - i. List of algorithms to pick
 - ii. List of event phenomena input
 - iii. Visualization screen
 - iv. Notification page (eventually)
 - b. Layout of grid & visuals? (What visuals will we need?)
5. Do we need a login for each user? Or is it just run locally?
 - a. Login screen for the user. Each user has own profile.
6. Is there a need to save simulations? Or simulation data? Export the data?
 - a. Backend will save simulation data for input combination, which can be used for future runnings.
7. Is there a preferred language/framework for either the frontend or the backend?
 - a. To be determined
8. What "data" will be collected while running the simulation?
 - a. Output format: starting location, starting time, destination location, arrival time, trajectory
9. Are there different types of events? How many?
 - a. For the purposes of this project there will be 1 "event" (ie something happened at this location at this time).
10. What system(s) will this be running on? [Physical resources]
 - a. Skip for the moment. Might be cloud based
11. Is the software supposed to be supportable/maintainable after our team leaves?
 - a. Possibility yes (non functional requirement)