

Agenda:

1. Gantt Chart/Project Management Objectives
2. Risk Assessment
3. Budget Management
4. Full Tech Stack Set-up
5. Capstone Deliverables
 - a. Functional vs. Non-functional Requirements

Notes:**Gantt Chartt:**

- Jira Software can be used to make a timeline with the sprints.
- Can't have a full Gantt chart. Evolving.
- Sprints/Sprint Review = Milestones.
- Sprints will be based on tickets and points will be used to determine time taken for tickets?

Budget Management:

- Domain Purchase? Github Student Package?

Risk Assessment:

- Do it early in the week.
- If we run into any issues, we can ask questions to the CSA/team based on the
- Informal similar to ENG2001 risk assessment/matrix.

Full Tech Stack:

- How do we establish a mock ground station? Based on lat/lon? Based on Schedules for requests made for availability? Up-link & Down-link rates?
- What are the outputs? Image files for different sizes?
- Database will be receiving information and needs to be dynamic so we'll use SQL.
- Architecture:
 - Backend: Django (can support REST APIs and ORM), Flask.
 - Database: SQL (MySQL / Postgres) => AWS RDS, AWS S3
 - Frontend: Typescript, NextJS
 - Deployment: Docker / Kubernetes
 - Testing: Unit Testing, Integration, E2E Testing (Cypress / Selenium)
 - Frontend and Backend should be separate components to mitigate risk.
 - Performance should be able to work on any computer

Backend Inputs:

- Orbit of Satellites, Orbital Optimization/Calculations
- Image Requests
- Schedule
 - Maintenance Activities

Set-up a Github repository.

Develop UML design for Architecture.

What are the expected outputs? Time-bound commands?

Capstone Deliverables:

- Develop folders for reports/presentations
- Mission Requirements

Next Actions:

- Send email about questions for Regina before sending to the CSA.
 - How do we establish a mock ground station? Based on lat/lon? Based on Schedules for requests made for availability? Up-link & Down-link rates?
 - What are the expected outputs? Time-bound commands?
 - Here is the sample architecture for our full tech stack, we wanted feedback on the tech stack to see if any of them would be a no-go as mentioned in our previous meeting.

Backend Inputs:

- Orbit of Satellites, Orbital Optimization/Calculations
- Image Requests
- Schedule
 - Maintenance Activities

Database will be receiving information and needs to be dynamic so we'll use SQL.

Architecture:

Backend: Django (can support REST APIs and ORM), Flask.

Database: SQL (MySQL / Postgres) => AWS RDS, AWS S3

Frontend: Typescript, NextJS

Deployment: Docker / Kubernetes

Testing: Unit Testing, Integration, E2E Testing (Cypress / Selenium)

Frontend and Backend should be separate components to mitigate risk.

- Set-up Github repository.
- Develop UML design.
- Send email to Franz about Friday Project feedback/peer review on problem definition.
- Work on early definition for risk assessment.